BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1. Implementasi

Implementasi merupakan tahap menerjemahkan perancangan berdasarkan hasil analisis. Tujuan implementasi adalah untuk mengkonfirmasikan modul program perancangan pada para pelaku sistem sehingga *user* dapat memberikan masukkan kepada pengembang sistem.

4.1.1. Perangkat Keras

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 :

Tabel 4.1 perangkat keras

No.	Perangkat Keras	Spesifikasi Minimum	Eksistensi	Keterangan
1.	Processor	Intel Pentium IV 1,3 GHz	Intel Pentium Dual Core	Memenuhi
2.	RAM	512 MB	1 Gb	Sudah melebihi batas minimum
4.	Harddisk	40 GB	80 GB	Memenuhi

4.1.2. Perangkat Lunak

Perangkat lunak yang digunakan dalam mengimplementasikan sistem ini adalah sebagai berikut:

Tabel 4.2 perangkat lunak yang digunakan

No.	Perangkat Lunak	Keterangan
1.	Windows XP SP 2	Sistem operasi
2.	XAMPP 1.7	penyimpanan *database*.
3.	*Borland Delphi 7*	*Tools* pembangun aplikasi

4.1.3. Implementasi Basis Data

Pembuatan *database* dilakukan dengan menggunakan aplikasi DBMS MySQL.

Implementasi *database* dalam bahasa SQL adalah sebagai berikut :

1. Pembuatan *database*:

```
DROP DATABASE IF EXISTS `dbwarehouse`;
CREATE DATABASE IF NOT EXISTS `dbwarehouse`;
USE `dbwarehouse`;
```

2. TABEL dim_jamaah

```
DROP TABLE IF EXISTS `dim_jamaah`;
CREATE TABLE IF NOT EXISTS `dim_jamaah` (
  `no_jamaah` int(10) NOT NULL AUTO_INCREMENT,
  `nama` varchar(25) NOT NULL,
  `jenis_kelamin` varchar(10)  NOT NULL,
  PRIMARY KEY (`no_jamaah`)
) ENGINE=InnoDB AUTO_INCREMENT=11728 DEFAULT
CHARSET=latin1;
```

3. TABEL dim_paket

```
DROP TABLE IF EXISTS `dim_paket`;
CREATE TABLE IF NOT EXISTS `dim_paket` (
        `kode_paket` varchar(30) NOT NULL,
        `jenis1` varchar(5) NOT NULL,
        `jenis2` varchar(20) NOT NULL,
        `tgl_berangkat` varchar(10) NOT NULL,
        `bulan` varchar(15) NOT NULL,
        `tahun` year(4) NOT NULL,
        PRIMARY KEY (`kode_paket`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4. TABEL dim_rekap_bayar

```
DROP TABLE IF EXISTS `dim_rekap_bayar`;
CREATE TABLE IF NOT EXISTS `dim_rekap_bayar` (
  `id_rekap_bayar` char(5)  NOT NULL,
  `kewajiban_rp` double NOT NULL DEFAULT '0',
  `kewajiban_usd` double NOT NULL DEFAULT '0',
  PRIMARY KEY (`id_rekap_bayar`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

5. TABEL dim_waktu

```
DROP TABLE IF EXISTS `dim_waktu`;
CREATE TABLE IF NOT EXISTS `dim_waktu` (
      `id_waktu` int(5) NOT NULL AUTO_INCREMENT,
      `tanggal` varchar(10) NOT NULL,
      `bulan` varchar(15) NOT NULL,
      `tahun` year(4) NOT NULL,
      PRIMARY KEY (`id_waktu`)
    ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

6. TABEL fact_pembayaran

```
DROP TABLE IF EXISTS `fact_pembayaran`;
CREATE TABLE IF NOT EXISTS `fact_pembayaran` (
  `no_invoice` varchar(15) NOT NULL,
  `kode_paket` varchar(30) NOT NULL,
  `id_rekap_bayar` char(5) NOT NULL,
  `id_waktu` int(10) NOT NULL,
  `tgl_bayar` date NOT NULL,
```

```
  `total_usd` double NOT NULL DEFAULT '0',
  `total_rp` double NOT NULL DEFAULT '0',
  PRIMARY KEY (`no_invoice`),
  KEY `fk_kode_paket` (`kode_paket`),
  KEY `fk_rekap_bayar` (`id_rekap_bayar`),
  KEY `id_waktu` (`id_waktu`),
  CONSTRAINT `fact_pembayaran_ibfk_1` FOREIGN KEY (`kode_paket`)
REFERENCES `dim_paket` (`kode_paket`),
  CONSTRAINT `fact_pembayaran_ibfk_2` FOREIGN KEY (`id_rekap_bayar`)
REFERENCES `dim_rekap_bayar` (`id_rekap_bayar`),
  CONSTRAINT `fact_pembayaran_ibfk_3` FOREIGN KEY (`id_waktu`)
REFERENCES `dim_waktu` (`id_waktu`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

7. TABEL fact_pendaftaran

```
DROP TABLE IF EXISTS `fact_pendaftaran`;
CREATE TABLE IF NOT EXISTS `fact_pendaftaran` (
  `no_pendaftaran` int(10) NOT NULL,
  `id_waktu` int(10) NOT NULL,
  `kode_paket` varchar(30) NOT NULL,
  `no_jamaah` int(10) NOT NULL,
  `tgl_daftar` date NOT NULL,
  `jumlah` int(10) NOT NULL,
  PRIMARY KEY (`no_pendaftaran`),
  KEY `id_waktudx` (`id_waktu`),
  KEY `kode_paketdx` (`kode_paket`),
  KEY `no_jamaahdx` (`no_jamaah`),
  CONSTRAINT `fact_pendaftaran_ibfk_1` FOREIGN KEY (`id_waktu`)
REFERENCES `dim_waktu` (`id_waktu`) ON DELETE CASCADE ON
UPDATE CASCADE,
  CONSTRAINT `fact_pendaftaran_ibfk_2` FOREIGN KEY (`kode_paket`)
REFERENCES `dim_paket` (`kode_paket`) ON DELETE CASCADE ON
UPDATE CASCADE,
  CONSTRAINT `fact_pendaftaran_ibfk_3` FOREIGN KEY (`no_jamaah`)
REFERENCES `dim_jamaah` (`no_jamaah`) ON DELETE CASCADE ON
UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin7;
```

4.1.4. Implementasi Antarmuka

Implementasi antarmuka dilakukan dengan menampilkan setiap tampilan sistem yang dibangun dan pengkodeannya dalam bentuk *file* program. Adapun untuk tampilan implementasi sistem adalah sebagai berikut :

Tabel 4.3 Tabel Implementasi Perangkat Lunak

No. Proses	Nama Proses	Nama File
1	*login*	*login*.pas
2	Upload data	Unit7.pas
3	ETL	Unit2.pas
4	Analisis	Unit6.pas
4.1	Input Filter Data	Unit6.pas
4.2	Tampil analisis data	Unit6.pas
4.3	Cetak Laporan	Unit6.pas

4.1.5. Implementasi Aplikasi *Data warehouse*

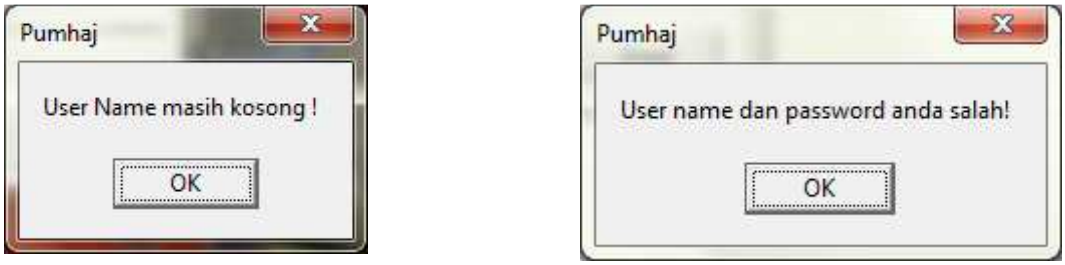
Pada subbab ini akan dibahas mengenai menu-menu pada aplikasi *data warehouse* yang telah dibangun mulai dari *login* aplikasi dan *form* menu utama.

4.1.5.1. *Form Login*

Form login berfungsi untuk masuk ke aplikasi *data warehouse*. Pada *form login* ini pengguna diharuskan memasukkan *username* dan *password* agar bisa masuk dan mengakses aplikasi. *Username* dan *password* pada proses *login* akan divalidasi oleh sistem apakah data *username* dan *password* yang dimasukkan sesuai dengan data yang ada pada tabel pegawai di *database* umhaj.



Gambar 4.1 Proses input *username* dan *password*



(a) (b)

Gambar 4.2 *Login* gagal

Jika *username* atau *password* yang dimasukkan salah maka akan keluar pesan *error* seperti ditunjukkan pada gambar 4.2(b), sebaliknya jika kombinasi *username* dan *password* yang dimasukkan benar maka akan muncul tampilan utama aplikasi seperti ditunjukkan pada gambar 4.3



Gambar 4.3 Tampilan utama aplikasi *data warehouse*

4.1.5.2. *Form* Menu Utama

Form ini merupakan *form* utama yang dapat diakses setelah proses *login* berhasil. Pada *form* ini terdapat dua menu utama, yaitu menu proses ETL dan menu analisis data.

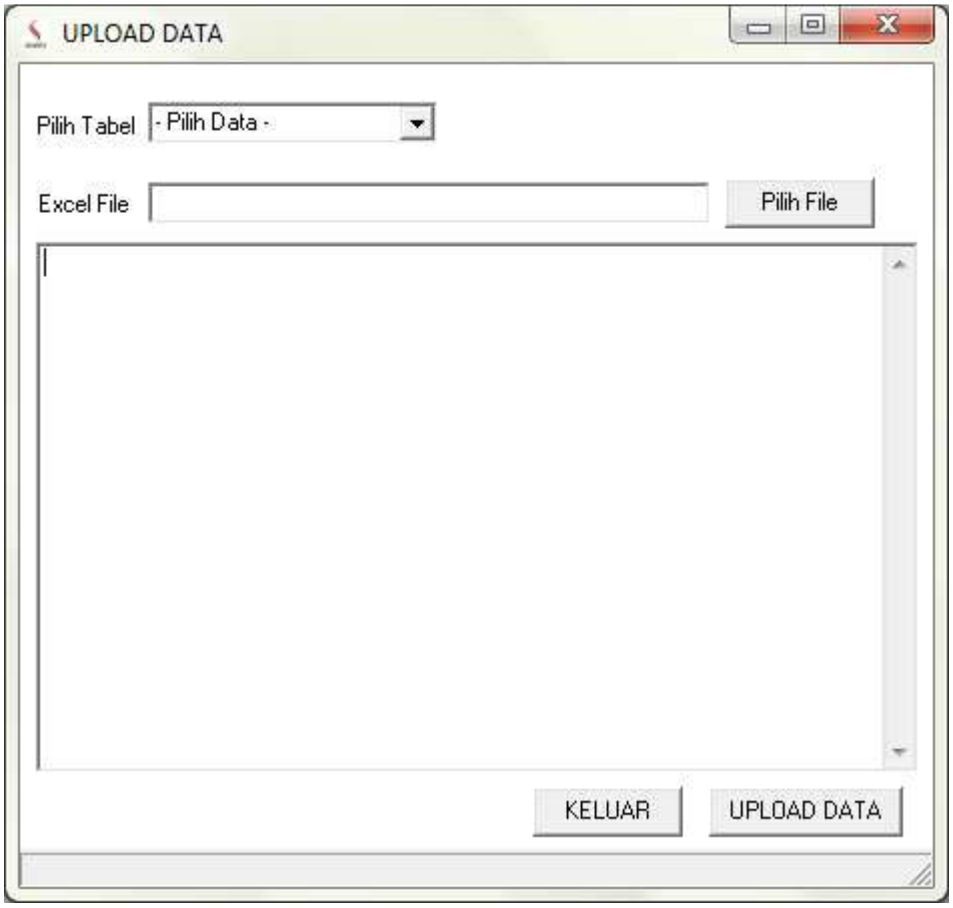


Gambar 4.4 Sub-menu menu ETL

Pada gambar 4.4 Menunjukkan sub menu yang dapat diakses.

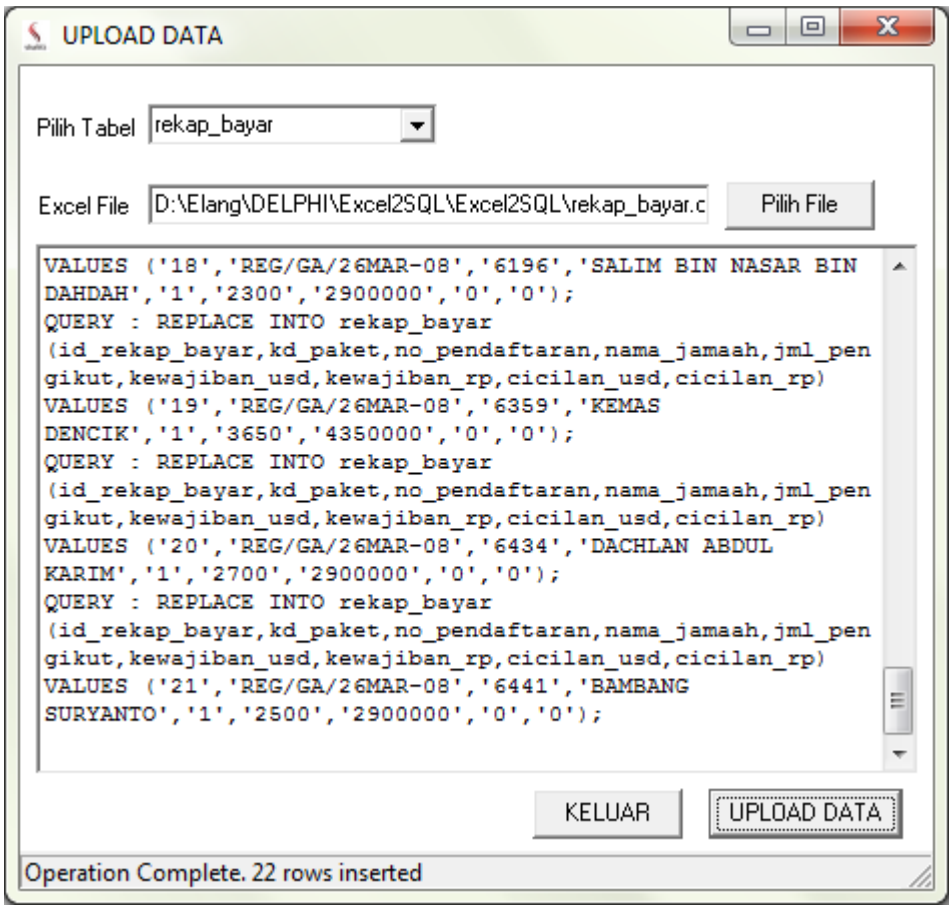
4.1.5.3. *Form Upload File Excel* Ke *Database* UMHAJ

Proses *upload file* dilakukan untuk memasukkan data dari sumber data. Sumber data pada aplikasi *data warehouse* ini adalah *file excel* yang dimasukkan ke dalam tempat penampungan sementara, yaitu *database* umhaj, terlebih dahulu, sebelum data diekstrak ke *database warehouse*. Untuk melakukan proses *upload data*, *user* menekan tombol *upload*, selanjutnya memilih data apa yang akan dimasukkan, kemudian *browse file data*, setelah itu tekan tombol *upload data*.



Gambar 4.5 Tampilan *form* sebelum proses *upload*

Pada gambar 4.5 menunjukkan kondisi *form menu upload data* sebelum tombol *upload data* ditekan. Pada awal *from* dibuka, *form* tidak menampilkan data apapun. Setelah memilih *file data* kemudian menekan tombol *upload data* dan proses *upload* dilakukan, maka akan tampil pesan informasi bahwa *upload* berhasil dilakukan. Pada *form* juga akan tampil keterangan data apa saja yang berhasil di-upload dan masuk ke dalam *database.*



Gambar 4.6 Tampilan *form* setelah proses *upload* data

4.2. 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.

4.2.1. Pengujian Alpha

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

1. Pengujian *Login*

Pengujian *login* adalah pengujian untuk verifikasi *username* dan *password user* yang akan menggunakan aplikasi dijelaskan pada tabel berikut:

Tabel 4.4 Pengujian *Login*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Username: admin *Password* : admin	*username, password* tercantum dalam *text box.*	Dapat mengisi *username*, dan *password*	[√]diterima [ ]ditolak
Klik tombol *Masuk*	Data *user* dicari di tabel *user*.	Tombol *login* dapat berfungsi dan *user* dapat masuk menu utama sesuai yang diharapkan.	[√]diterima [ ]ditolak

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Username : admin *Password* : 123	*username, password* tercantum dalam *text box.*	Dapat mengisi *username*, dan *password*	[√]diterima [ ]ditolak
Klik tombol *Masuk*	Tidak dapat *login* dan menampilkan pesan "user name dan *password* anda salah!"	*user* tidak dapat *login* dan menampilkan pesan "user name dan *password* anda salah!" sesuai yang diharapkan.	[√]diterima [ ]ditolak

2. Pengujian ETL

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

Tabel 4.5 Pengujian ETL

Kasus dan hasil uji (data normal)			
Data masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Periode ETL Awal : 01/01/2007 Akhir : 30/12/2007	Dapat melakukan proses ETL sesuai dengan batasan periode yang dimasukkan.	Hasil ETL sesuai dengan periode yang dimasukkan user	[√] diterima [ ] ditolak

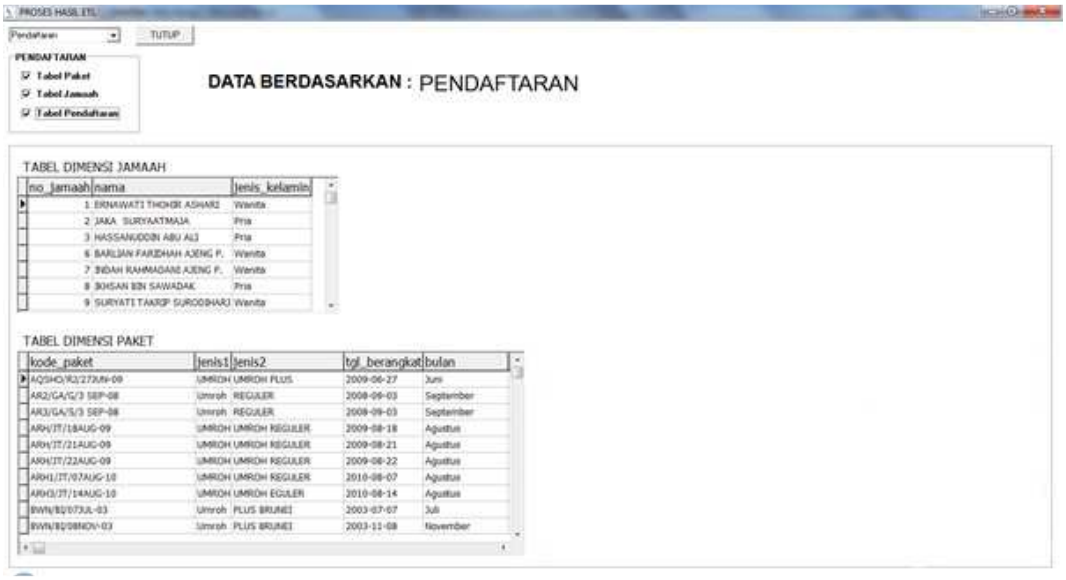
Pada aplikasi ini proses *extract, transform, load* data dilakukan secara bersamaan ketika menekan tombol proses ETL. Proses pengujian ini dilakukan untuk mengekstrak, mentrans*form*asi, dan me-load data dari data sumber ke *data warehouse*. Sebelum melakukan ekstrak dan *transform* data, *user* harus memilih

periode waktu terlebih dahulu. Pemilihan periode waktu ini dilakukan untuk menentukan data mana yang akan di-load ke *data warehouse*.



Gambar 4.7 *Setting* periode ETL

Setelah periode ditentukan maka selanjutnya tekan tombol proses dan proses ETL berlangsung. *User* dapat melihat data yang telah berhasil di-load ke *data warehouse* sesudah proses ETL pada *form* hasil ETL. Pada *form* ini *user* dapat menentukan data apa yang ingin ditampilkan.



Gambar 4.8 *Form* hasil ETL

Gambar 4.8 menunjukkan tampilan setelah proses ETL dilakukan.

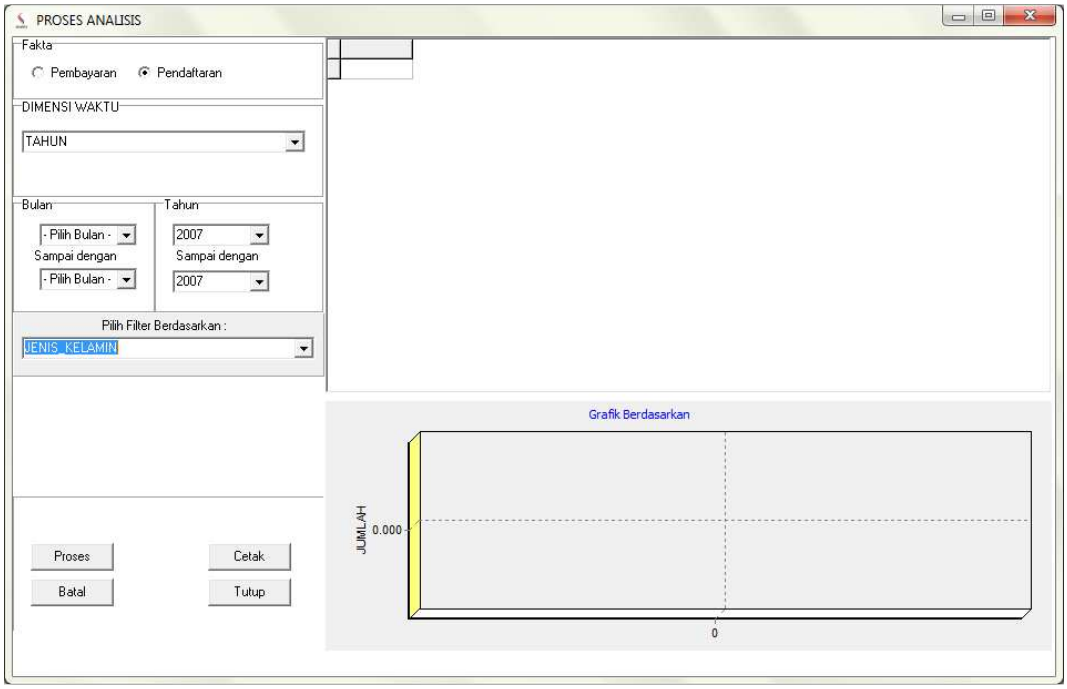
3. Pengujian Analisis

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

Tabel 4.6 Tabel Pengujian Analisis

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Fakta : pendaftaran	Menampilkan data sesuai fakta yang dipilih	Tampil data sesuai fakta yang dipilih	[√] diterima [ ] ditolak
Fakta : pendaftaran Dimensi waktu : Tahun : 2007 Filter data : jenis_kelamin	menampilkan data sesuai fakta yang dipilih dengan di filter berdasarkan dimensi yang dipilih	Tampil data sesuai fakta yang dipilih dan di filter berdasarkan dimensi yang dipilih.	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Tidak memilih fakta Fakta : -	Proses analisis dibatalkan dan muncul pesan “Pilih fakta terlebih dahulu”.	Proses analisis dibatalkan kemudian muncul pesan “Pilih fakta terlebih dahulu”	[√] diterima [ ] ditolak

Proses pengujian pada menu analisis data adalah lanjutan dari pengujian proses ETL. Sebelum analisis data dilakukan *user* harus terlebih dahulu memilih data apa yang akan dianalisis.

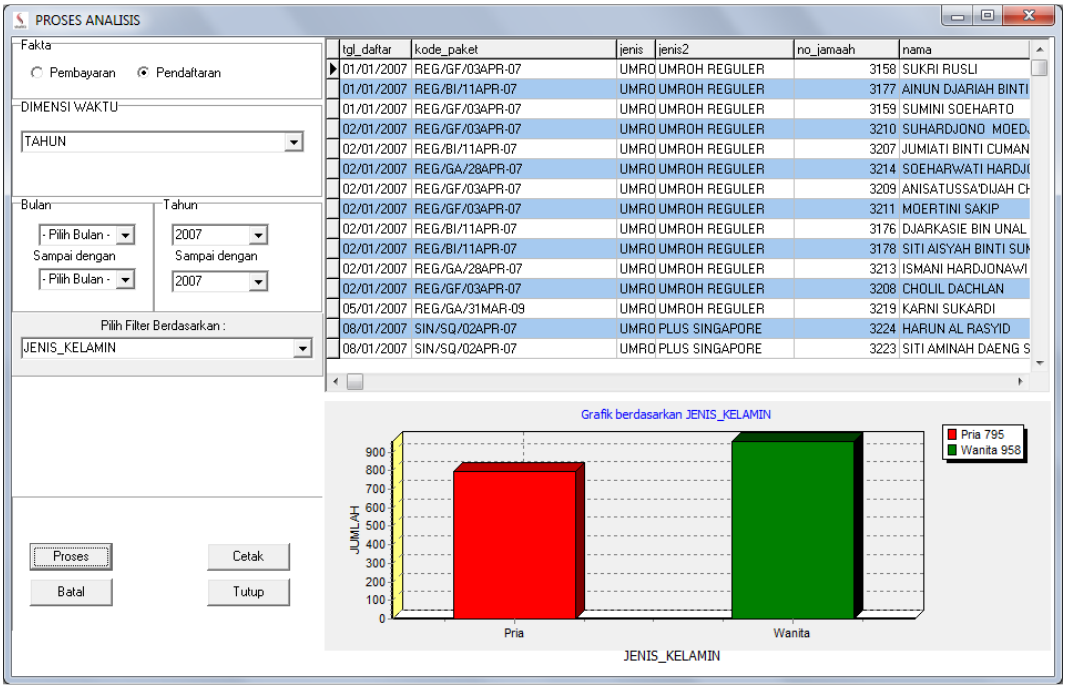


Gambar 4.9 *Form* analisis data

Gambar 4.9 menjelaskan pemilihan periode dan *filter* data untuk analisis data. Data yang akan dianalisis dan ditampilkan oleh perangkat lunak OLAP adalah data pendaftaran pada tahun 2007 berdasarkan jenis kelamin jamaah.

Dari perangkat lunak OLAP, Shafira dapat melakukan analisis data. Dari analisis data yang dilakukan oleh Shafira terhadap informasi dari perangkat lunak OLAP dapat membantu Shafira dalam pembuatan laporan yang dibutuhkan serta mengambil keputusan dari bisnis proses pada Shafira, misalnya dapat melakukan beberapa analisis seperti :

1. Shafira menganalisis jumlah jamaah yang telah mendaftar berdasarkan jenis kelamin untuk membantu dalam proses penyuluhan dan meningkatkan mutu pelayanan.
2. Shafira menganalisis jenis paket yang paling banyak diminati jamaah per bulan dan per tahun dapat membantu untuk proses promosi.



Gambar 4.10 Analisis data pendaftaran jamaah pada tahun 2007

Gambar 4.10 menjelaskan tentang analisis data pendaftaran dari jumlah jamaah pada tahun 2007. Selain itu juga ditampilkan grafik dari jumlah jumlah jamaah berdasarkan jenis kelamin.

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.

4.2.2. Pengujian *Beta*

Pengujian *beta* merupakan pengujian yang dilakukan secara langsung pada responden yang dapat menjadi *user* aplikasi ini. Yaitu untuk mengetahui kualitas dari *software* yang telah dibuat, apakah sudah sesuai dengan yang diharapkan atau belum, pengujian dilakukan dengan melakukan wawancara.

Wawancara

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

1. Bagaimana tampilan/desain aplikasi *data warehouse* ini ?
2. Bagaimana kemudahan penggunaan aplikasi *data warehouse* ini ?
3. Apakah aplikasi *data warehouse* ini membantu user dalam menganalisis data jadi lebih mudah?
4. Bagaimana keakuratan hasil output?
5. Apakah lebih mudah dalam pembuatan laporan?
6. Apakah ada saran untuk pengembangan aplikasi ini kedepannya ?

Hasil Wawancara

Berikut adalah jawaban dari user terhadap pertanyaan utama diatas :

1	Pertanyaan	Bagaimana tampilan/desain aplikasi *data warehouse* ini ?
	Jawaban	Aplikasi *data warehouse* sudah cukup bagus
2	Pertanyaan	Bagaimana kemudahan penggunaan aplikasi *data warehouse* ini ?
	Jawaban	Mudah digunakan
3	Pertanyaan	Apakah aplikasi *data warehouse* ini membantu user dalam menganalisis data jadi lebih mudah?
	Jawaban	*Data warehouse* ini cukup membantu dalam proses analisis data

4	Pertanyaan	Bagaimana keakuratan hasil output ?
	Jawaban	Cukup akurat
5	Pertanyaan	Apakah lebih mudah dalam pembuatan laporan ?
	Jawaban	Cukup membantu dalam pembuatan laporan
6	Pertanyaan	Apakah ada saran untuk pengembangan aplikasi ini kedepannya ?
	Jawaban	Dikembangkan lagi dengan menambahkan fungsi di bagian menu análisis dan grafik serta data yang akan dianalisis.

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*.