BAB 4
IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini akan dilakukan implementasi dan pengujian terhadap perangkat lunak *data preparation* yang telah dirancang. Tahapan ini dilakukan setelah analisis dan perancangan selesai dilakukan dan selanjutnya akan diimplementasikan pada bahasa pemrograman. Setelah implementasi dilakukan, maka dilakukan pengujian terhadap perangakat lunak *data preparation* yang telah dibangun dan akan dilihat kekurangan-kekurangan pada perangkat lunak untuk dievaluasi dan pengembangan selanjutnya.

4.1 Implementasi Sistem

Setelah sistem dianalisis dan desain secara rinci, maka akan menuju tahap implementasi sistem. Tahap implementasi sistem merupakan tahap menerjemahkan sistem ke dalam bahasa yang dimengerti oleh mesin berdasarkan hasil analisis dan perancangan sistem yang telah dilakukan. Implementasi bertujuan untuk mengkonfirmasi modul-modul perancangan kepada para pelaku sistem, sehingga pengguna dapat memberi masukan untuk pengembangan sistem.

4.1.1 Implementasi Perangkat Keras yang Digunakan

Spesifikasi perangakat keras (*hardware*) yang digunakan untuk membangun perangkat lunak *data preparation* ini dapat dilihat pada tabel 4.1 berikut :

Tabel 4.1 Spesifikasi Perangkat Keras

Jenis Perangkat Keras	Spesifikasi
Processor	Intel core i5 4.3 GHz
Memory (RAM)	4GB
Harddisk	320GB
VGA	128MB

Jenis Perangkat Keras	Spesifikasi
Optical Drive	DVD RW
Monitor	13”
Keyboard	*Standard*
Mouse	*Standard*

4.1.2 Implementasi Perangkat Lunak yang Digunakan

Spesifikasi perangkat lunak (*Software*) yang digunakan untuk membangun aplikasi *data preparation* ini dapat dilihat pada tabel 4.2 berikut :

Tabel 4.2 Spesifikasi Perangkat Lunak

Jenis Perangkat Lunak	Keterangan
Ubuntu 10.10	Sistem Operasi
Netbeans 7.0.1	Developer tool
JDK 1.7	Java Development Kit
MySQL 5.0	DBMS
MySQL Administrator dan MySQL Query Browser	*MySQL manager and admin tools*
Visual Paradigm for UML 8.3 dan Astah Profesional	UML modeling tool

4.1.3 Implementasi Basis Data

Pembuatan *database* dilakukan dengan menggunakan bantuan aplikasi MySQL Query Browser sebagai GUI tool dalam merancang dan membangun *database* perangkat lunak *data preparation* di *MySQL*. Implementasi pembuatan *database* dengan bahasa *SQL* beserta tabel-tabel yang mendukung perangkat lunak *data preparation* adalah sebagai berikut :

1. **Pembuatan *Database* Operasional**

Database operasional pada pembangunan aplikasi *data preparation* adalah sebagai berikut :

```
CREATE DATABASE  `db_cuaca`;
```

a. Tabel Temperatur

Struktur tabel temperatur pada pembangunan aplikasi *data preparation* adalah sebagai berikut :

```
CREATE TABLE `temperatur` (
  `KodeTemperatur` varchar(7) NOT NULL,
  `T1` float NOT NULL,
  `T2` float NOT NULL,
  `T3` float NOT NULL,
  `RataRata` float NOT NULL,
  PRIMARY KEY (`KodeTemperatur`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Berikut dapat dilihat implementasi dari tabel temperatur :

Tabel 4.3 Tabel Temperatur

Column Name	Data Type	NOT NULL	AUTO INC	Flags	Default Value	Comments
KodeTemperatur	VARCHAR(7)	☑	☐		NULL	
T1	FLOAT	☑	☐		NULL	
T2	FLOAT	☑	☐		NULL	
T3	FLOAT	☑	☐		NULL	
RataRata	FLOAT	☑	☐		NULL	
		☑	☐			

b. Tabel Kelembaban

Struktur tabel kelembaban pada pembangunan aplikasi *data preparation* adalah sebagai berikut :

```
CREATE TABLE `kelembaban` (
  `KodeKelembaban` varchar(7) NOT NULL,
  `K1` int(11) NOT NULL,
  `K2` int(11) NOT NULL,
  `K3` int(11) NOT NULL,
  `RataRata` int(11) NOT NULL,
  PRIMARY KEY (`KodeKelembaban`)
      ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Berikut dapat dilihat implementasi dari tabel kelembaban :

Tabel 4.4 Tabel Kelembaban

Column Name	Data Type	NOT NULL	AUTO INC	Flags	Default Value	Comments
KodeKelembaban	VARCHAR(7)	☑	☐		NULL	
K1	INTEGER	☑	☐		NULL	
K2	INTEGER	☑	☐		NULL	
K3	INTEGER	☑	☐		NULL	
RataRata	INTEGER	☑	☐		NULL	
		☑	☐			

c. Tabel Angin

Struktur tabel angin pada pembangunan aplikasi *data preparation* adalah sebagai berikut :

```
 CREATE TABLE `angin` (
`KodeAngin` varchar(7) NOT NULL,
`KecRataRata` float NOT NULL,
`ArahTerbanyak` char(10) NOT NULL,
`KecAnginTerbesar` float NOT NULL,
`Arah` char(10) NOT NULL,
PRIMARY KEY (`KodeAngin`)
 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Berikut dapat dilihat implementasi dari tabel angina :

Tabel 4.5 Tabel Angin

Column Name	Data Type	NOT NULL	AUTO INC	Flags	Default Value	Comments
KodeAngin	VARCHAR(7)	☑	☐		NULL	
KecRataRata	INTEGER	☑	☐		NULL	
ArahTerbanyak	CHAR(10)	☑	☐		NULL	
KecAnginTerbesar	INTEGER	☑	☐		NULL	
Arah	CHAR(10)	☑	☐		NULL	
		☑	☐			

d. Tabel Kondisi

Struktur tabel kondisi pada pembangunan aplikasi *data preparation* adalah sebagai berikut :

```
 CREATE TABLE `kondisi` (
`KodeKondisi` varchar(7) NOT NULL,
`KodeTemperatur` varchar(7) NOT NULL,
`KodeKelembaban` varchar(7) NOT NULL,
`KodeAngin` varchar(7) NOT NULL,
`CurahHujan` float NOT NULL,
`Tanggal` date NOT NULL,
PRIMARY KEY (`KodeKondisi`)
   ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Berikut dapat dilihat implementasi dari tabel kondisi :

Tabel 4.6 Tabel Kondisi

Column Name	Data Type	NOT NULL	AUTO INC	Flags	Default Value	Comments
KodeKondisi	VARCHAR(7)	☑	☐		NULL	
KodeTemperatur	VARCHAR(7)	☑	☐		NULL	
KodeKelembaban	VARCHAR(7)	☑	☐		NULL	
KodeAngin	VARCHAR(7)	☑	☐		NULL	
CurahHujan	FLOAT	☑	☐		NULL	
Tanggal	DATE	☑	☐		NULL	
		☑	☐			

2. Pembuatan *Database Data Mart*

Database data mart pada pembangunan aplikasi *data preparation* adalah sebagai berikut :

```
    CREATE DATABASE  `db_martcuaca`;
```

a. Tabel Fakta Kondisi

Struktur tabel fakta kondisi pada pembangunan aplikasi *data preparation* adalah sebagai berikut :

```
CREATE TABLE `fakta_kondisi` (
  `kode_kondisi` int(11) NOT NULL AUTO_INCREMENT,
  `kode_cuaca` int(11) NOT NULL,
  `curah_hujan` varchar(50) NOT NULL,
  `jumlah` int(11) NOT NULL,
  `kode_waktu` int(11) NOT NULL,
  PRIMARY KEY (`kode_kondisi`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Berikut dapat dilihat implementasi dari tabel kondisi :

Tabel 4.7 Tabel Fakta Kondisi

Column Name	Data Type	NOT NULL	AUTO INC	Flags	Default Value	Comments
kode_kondisi	INTEGER	☑	☑		NULL	
kode_cuaca	INTEGER	☑	☐		NULL	
curah_hujan	VARCHAR(50)	☑	☐		NULL	
jumlah	INTEGER	☑	☐		NULL	
kode_waktu	INTEGER	☑	☐		NULL	
		☑	☐			

b. Tabel Dimensi Cuaca

Struktur tabel dimensi cuaca pada pembangunan aplikasi *data preparation* adalah sebagai berikut :

```
CREATE TABLE `dim_cuaca` (
  `kode_cuaca` int(11) NOT NULL AUTO_INCREMENT,
  `unsur_cuaca` varchar(50) NOT NULL,
  PRIMARY KEY (`kode_cuaca`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Berikut dapat dilihat implementasi dari tabel kondisi :

Tabel 4.8 Tabel Dimensi Cuaca

Column Name	Data Type	NOT NULL	AUTO INC	Flags	Default Value	Comments
kode_cuaca	INTEGER	☑	☑		NULL	
unsur_cuaca	VARCHAR(50)	☑	☐		NULL	
		☑	☐			

c. Tabel Dimensi Waktu

Struktur tabel dimensi waktu pada pembangunan aplikasi *data preparation* adalah sebagai berikut :

```
CREATE TABLE `dim_waktu` (
  `kode_waktu` int(11) NOT NULL,
```

```
  `bulan` int(11) NOT NULL,
  `tahun` int(11) NOT NULL,
PRIMARY KEY (`kode_waktu`)
    ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Berikut dapat dilihat implementasi dari tabel kondisi :

Tabel 4.9 Tabel Dimensi Waktu

Column Name	Data Type	NOT NULL	AUTO INC	Flags	Default Value	Comments
kode_waktu	INTEGER	☑	☐		NULL	
bulan	INTEGER	☑	☐		NULL	
tahun	INTEGER	☑	☐		NULL	
		☑	☐			

4.1.4 Implementasi *Class*

Implementasi class pada aplikasi *data preparation* analisa pola data pemantauan cuaca dapat dilihat pada tabel berikut :

1. Package dim.cuaca.view

Tabel 4.10 Implementasi Class View

Class Name	***Syntax***
MainFrame	public class MainFrame extends javax.swing.Jframe
MenuData	public class MenuData extends javax.swing.Jpanel
MenuETL	public class MenuETL extends javax.swing.Jpanel
Tentang	public class Tentang extends javax.swing.Jpanel

Class Name	*Syntax*
BrowseDataDialog	public class BrowseDataDialog extends java.awt.Dialog
LihatTabelDialog	public class LihatTabelDialog extends java.awt.Dialog
LihatTabelTemperaturPanel	public class LihatTabelTemperaturPanel extends javax.swing.Jpanel
LihatTabelKondisiPanel	public class LihatKondisiPanel extends javax.swing.Jpanel
LihatTabelAnginPanel	public class LihatTabelAnginPanel extends javax.swing.Jpanel
LihatTabelKelembabanPanel	public class LihatTabelKelembabanPanel extends javax.swing.Jpanel
GrafikDialog	public class GrafikDialog extends java.awt.Dialog
ProsesETLDialog	public class ProsesETLDialog extends java.swt.Dialog
GrafikETLDialog	public class GrafikETLDialog extends java.swt.Dialog

2. Package dm.cuaca.model.table

Tabel 4.11 Implementasi Class Model Tabel

Class Name	*Syntax*
TabelAnginModel	public class TabelAnginModel extends AbstractTableModel
TabelKondisiModel	public class TabelKondisiModel extends AbstractTableModel

Class Name	*Syntax*
TabelKelembabanModel	public class TabelKelembabanModel extends AbstractTableModel
TabelKondisiModel	public class TabelKondisiModel extends AbstractTableModel
TabelTemperaturModel	public class TabelTemperaturModel extends AbstractTableModel

3. Package dm.cuaca.model.table.dw

Tabel 4.12 Implementasi Class Model Tabel DW

Class Name	*Syntax*
TabelFactKondisiModelDw	public class TabelFactKondisiModelDw extends AbstractTableModel

4. Package dm.cuaca.entitas

Tabel 4.13 Implementasi Class Entitas

Class Name	*Syntax*
Angin	public class Angin
Kelembaban	public class Kelembaban
Kondisi	public class Kondisi
Temperatur	public class Temperatur

5. Package dm.cuaca.entitas.dw

Tabel 4.14 Implementasi Class Entitas Dw

Class Name	*Syntax*
FaktaKondisiDw	public class FaktaKondisiDw
DimCuacaDw	public classDimCuacaDw
DimWaktuDw	Public classDimWaktuDw

6. Package dm.cuaca.dao

Tabel 4.15 Implementasi Class Dao

Class Name	*Syntax*
AnginDao	public class AnginDao
KelembabanDao	public class KelembabanDao
KondisiDao	public class KondisiDao
TemperaturDao	public class TemperaturDao

7. dm.cuaca.dao.dw

Tabel 4.16 Implementasi Class Dao Dw

Class Name	*Syntax*
FaktaKondisiDaoDw	public class FaktaKondisiDaoDw
DimCuacaDaoDw	public classDimCuacaDaoDw
DimWaktuDaoDw	Public classDimWaktuDaoDw

4.1.5 Implementasi Antarmuka

Implementasi antarmuka dilakukan dengan menampilkan setiap tampilan program yang dibangun dan pengkodeannya dalam bentuk *file*

program. Berikut ini adalah implementasi antarmuka sistem dapat dilihat dilihat pada tabel 4.17 berikut :

Tabel 4.17 Implementasi antarmuka

Form	***Description***	***File Name***
Halaman Utama (*Home*)	Tampilan awal aplikasi pada saat dijalankan.	MainFrame.java
Menu Data	Tampilan untuk melihat submenu data.	MenuAnalisis.java
Menu ETL	Tampilan untuk melihat submenu ETL.	MenuETL.java
Halaman Tentang	Tampilan untuk melihat informasi aplikasi dan pembuatnya.	Tentang.java
Browse Data	Tampilan untuk proses pemgambilan data dari file eksternal untuk dimasukkan ke dalam aplikasi.	BrowseDataDialog.java
Lihat Tabel	Tampilan untuk melihat hasil data yang telah diinput dari luar.	LihatTabelDialog.java
Tabel Temperatur	Tampilan untuk melihat data temperatur.	LihatTabelTemperaturPanel .java
Tabel Kondisi	Tampilan untuk melihat data kondisi	LihatTabelKondisi.java
Tabel Angin	Tampilan untuk melihat data angin.	LihatTabelAngin.java

Form	*Description*	*File Name*
Tabel Kelembaban	Tampilan untuk melihat data kelembaban.	LihatTabelKelembaban.java
Lihat Grafik	Tampilan untuk melihat informasi cuaca yang terjadi pada wktu tertentu.	GrafikDialog.java
Proses ETL	Tampilan untuk proses ekstrak, transform data operasional dan load data tersebut ke tempat penyimpanan baru yaitu data mart.	ProsesETLDialog.java
Grafik Hasil ETL	Tampilan untuk grafikhasil ETL.	GrafikETLPanel.java

4.2 Pengujian Perangkat Lunak

Pengujian sistem merupakan hal terpenting yang bertujuan untuk menemukan kesalahan–kesalahan atau kekurangan–kekurangan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahuiperangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut.Untuk menguji kebenaran program tersebut maka program harus diuji dengan data tes yang telah dipersiapkan sebelumnya, sehingga bila tidak sesuai dengan kebutuhan, kesalahan sintaks, kesalahan logika dan kesalahan pada saat *run time* dapat diketahui secara dini.Rencana pengujian yang akan dilakukan adalah dengan menguji sistem secara *alpha* dan *beta* serta pengujian terhadap kasus.

4.2.1 Pengujian *Alpha*

Pengujian *alpha* dilakukan dengan metode *black box*. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak.Pengujian ini menggunakan data pemantauan cuaca dari tahun 2005-2009.Langkah- langkah pengujiannya dilakukan pada setiap proses yang ada pada *form* aplikasi *data preparation* ini. Rencana pengujian selengkapnya dapat dilihat pada tabel berikut :

Tabel 4.18 Rencana Pengujian *Alpha*

Kelas Uji	Butir Uji	Jenis Pengujian
Browse Data	*Browse* Data	*Black Box*
	Load Data	*Black Box*
Lihat Tabel Data	Menampilkan Data Temperatur	*Black Box*
	Menampilkan Data Kelembaban	*Black Box*
	Menampilkan Data Curah Hujan	*Black Box*
	Menampilkan Data Angin	*Black Box*
Lihat Grafik	Menampilkan Grafik Temperatur	*Black Box*
	Menampilkan Grafik Kelembaban	*Black Box*
	Menampilkan Grafik Curah Hujan	*Black Box*
	Menampilkan Grafik Angin	*Black Box*
Proses *Extract*	*Input* Periode	*Black Box*
	Melakukan Proses *Extract*	*Black Box*
Proses *Transform*	Melakukan *transformasi* terhadap data hasil *extract*.	*Black Box*
Proses *Load*	Melakukan *load* data ke *database* yang telah ditransformasi.	*Black Box*
Lihat Grafik ETL	Menampilkan Hasil ETL	*Black Box*

4.2.1.1 Kasus dan Hasil Pengujian *Alpha*

Berdasarkan rencana pengujian, maka akan dijelaskan secara rinci mengenai pengujian terhadap perangkat lunak yang telah dibangun, adapun untuk setiap pengujiannya adalah sebagai berikut :

1. **Pengujian *Browse Data***

Tabel 4.19 Pengujian Browse Data

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Klik tombol *Browse*	Menampilkan directory sumber pengambilan data, dapat memilih data masukan yaitu data exel dan menampilkan path alamat sumber data di *field browse*.	Directory tempat pengambilan data berhasil ditampilkan, dapat memilih data masukan yang diinginkan dan dapat menampilkan path alamat sumber data.	[√] diterima [ ] ditolak
Klik tombol *Read Data*	Menampilkan *data sheet* exel dalam tabel.	*Data sheet*exel berhasil ditampilkan dalam tabel yang telah tersedia.	[√] diterima [ ] ditolak
Klik combobox Bulan. Misal, Maret	Menampilkan *data sheet*pada bulan Maret	*Data sheet* pada bulan Maret berhasil ditampilkan.	[√] diterima [ ] ditolak

Klik Input Data	*Data sheet*exel tersimpan dalam*database* transaksional.	*Data sheet*exel tersebut dapat tersimpan ke dalam *database* transaksional.	[√] diterima [ ] ditolak
Klik tombol input semua data	Seluruh data sheet exel tersimpan dalam *database* transaksional.	Seluruh data sheet exel berhasil tersimpan kedalam *database* transaksional.	[√] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Klik tombol *Browse*	Sistem tidak akan menampilkan jenis file lainnya, selain file exel berformat .xls.	Pada saat pemilihan file, aplikasi tidak dapat mengambil *file* yang bukan exel.	[√] diterima [] ditolak

2. Pengujian Lihat Data

Berikut dapat dilihat pengujian pada Lihat Data yang telah dilakukan yaitu sebagai berikut :

Tabel 4.20 Pengujian Lihat Data

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Klik tombol temperatur	Menampilkan data temperatur	Tampilan data sesuai.	[√] diterima [ ] ditolak

Klik tombol kelembaban	Menampilkan data kelembaban	Tampilan data sesuai.	[√] diterima [ ] ditolak
Klik tombol angina	Menampilkan data angina	Tampilan data sesuai.	[√] diterima [ ] ditolak
Klik tombol curah hujan	Menampilkan data curah hujan	Tampilan data sesuai.	[√] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
-	Tidak dapat menampilkan data temperatur	Tidak menampilkan data temperatur	[√] diterima [] ditolak
-	Tidak dapat menampilkan data kelembaban	Tidak menampilkan data kelembaban	[√] diterima [] ditolak
-	Tidak dapat menampilkan dataangin	Tidak menampilkan dataangin	[√] diterima [] ditolak
-	Tidak dapat menampilkan data curah hujan	Tidak menampilkan data curah hujan	[√] diterima [] ditolak

3. Pengujian Lihat Grafik

Berikut dapat dilihat pengujian pada Lihat Grafik yang telah dilakukan yaitu sebagai berikut :

Tabel 4.21 Pengujian Lihat Grafik

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Input tahun, bulan dan unsur cuaca pada combobox. Misal, Tahun 2005 Bulan 1 Tabel Temperatur	Tahun, bulan dan unsur cuaca tercantum dalam combobox.	Dapat memilih data yang ingin ditampilkan grafiknya.	[√] diterima [ ] ditolak
Klik tombol *show*	Menampilkan grafik temperatur tahun 2005 bulan 1.		[√] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
-	Tidak dapat menampilkan grafik.	Tidak menampilkan grafik.	[√] diterima [] ditolak

4. Pengujian Proses *Extract*

Berikut dapat dilihat pengujian pada Proses *Extract* yang telah dilakukan yaitu sebagai berikut :

Tabel 4.22 Pengujian Proses Ekstrak

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Input periode. Awal : 1 januari 2005 Akhir : 31 desember 2005	Periode tercantum dalam jcalender.	Dapat mengisi periode.	[√] diterima [ ] ditolak
Klik tombol ekstrak	Data hasil ekstrak ditampilkan.	Dapat mengekstrak data dari *database* transaksional sesuai dengan periode yang diinginkan.	[√] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
-	Menampilkan pesan pemilihan periode belum dilakukan.	Muncul pesan periode belum dipilih.	[√] diterima [] ditolak

5. Pengujian Proses *Transform*

Berikut dapat dilihat pengujian pada *Proses Transform* yang telah dilakukan untuk data hasil *extract* yang sebelumnya dilakukan yaitu sebagai berikut :

Tabel 4.23 Pengujian Proses *Transform*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Klik tombol *transform*	Melakukan transform data dan menampilkannya.	Data hasil ekstrak berhasil diransform dan ditampilkan.	[√] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
-	Menampilkan pesanekstrak data belum dilakukan.	Muncul pesan lakukan ekstrak data terlebih dahulu.	[√] diterima [] ditolak

6. Pengujian Proses *Load*

Berikut dapat dilihat pengujian pada proses *load* yang telah dilakukan untuk data hasil *transform* yang sebelumnya dilakukan yaitu sebagai berikut :

Tabel 4.24 Pengujian Proses *Load*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Datal transform	Data tersimpan dalam *data mart.*	Data berhasil di-*load* ke *data mart*.	[√] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
-	Menampilkan pesan ekstrak data dan transform belum	Muncul pesan lakukan ekstrak data serta	[√] diterima [] ditolak

	dilakukan.	transform terlebih dahulu.	

7. Pengujian Lihat Grafik ETL

Berikut dapat dilihat pengujian pada proses Lihat Grafik ETL yang telah dilakukan yaitu sebagai berikut :

Tabel 4.25 Pengujian Lihat Grafik Hasil ETL

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Input tahun, bulan dan unsur cuaca pada combobox. Misal, Tahun 2005 Bulan 1 Tabel Temperatur	Tahun, bulan dan unsur cuaca tercantum dalam combobox.	Dapat memilih data yang ingin ditampilkan grafiknya.	[√] diterima [ ] ditolak
Klik tombol *show*	Menampilkan grafik temperatur tahun 2005 bulan 1.		[√] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
-	Tidak dapat menampilkan grafik.	Tidak menampilkan grafik.	[√] diterima [] ditolak

4.2.1.2 Kesimpulan Hasil Pengujian *Alpha*

Berdasarkan hasil pengujian *alpha*dengan kasus uji *sample* yang telah dilakukan dapat ditarik kesimpulan bahwa perangkat lunak yang telah dibangun sudah memenuhi persyaratan secara fungsional, tetapi tidak menutup kemungkinan dapat terjadi kesalahan pada saat aplikasi digunakan, sehingga membutuhkan proses evaluasi untuk lebih mengetahui kekurangan dari aplikasi. Secara fungsional sistem dapat menghasilkan keluaran yang sesuai dengan yang diharapkan, begitu juga secara perhitungan sudah sesuai dengan analisis yang telah dilakukan.

4.2.2 Pengujian *Data Mart* Menggunakan OLAP

Data yang sudah melaluai proses ETL merupakan data yang akan disimpan dalam *data mart*, dan data ini yang digunakan sebagai bahan analisis dan laporan, dalam periode tertentu. Salah satu cara untuk mengetahui bahwa *data mart* yang telah dirancang sudah siap digunakan dalam proses analisis maka dilakukan pengujian terhadap *data mart* cuaca dengan menggunakan *OLAP*. Perancangan *OLAP* (On-Line Analytical Processing) pada *Data Mart* Cuaca yaitu dengan mengolah dan menganalisis data dari berbagai dimensi. Proses analisis akan dilakukan untuk menganalisa curah hujan yaitu dengan menggunakan *OLAP*, Nilai *atribute* seperti tanggal memiliki nilai yang menyatakan tahun dan bulan. Seringkali pengkategorian ini dapat diorganisasikan sebagai pohon hirarki. Struktur hirarki ini memunculkan operasi *Roll-up* dan *Drill-down.* Contohnya pada data curah hujan yang merupakan data *multidimensional* (fakta kondisi), dapat ditentukan agregasi curah hujan untuk jumlah per 1 bulan dan memisahkan total curah hujan per 3 bulan ke dalam total curah hujan.

1. *Roll up*

   Operator *roll up* menyebabkan peningkatan agregasi data dan menghapus level data yang lebih detail dari sebuah hirarki. Untuk data curah hujan, kita dapat mengumpulkan data (*Roll up*) curah hujan dari seluruh

tanggal (harian) dalam satu bulan. Sebagai contoh, untuk menampilkan jumlah curah hujan per bulan. *Roll up* dapat menampilkan informasi tentang nama cuaca dan menampilkan jumlah curah hujan berdasarkan periode total 3 bulanan. Taebl jumlah curah hujan rendah dalam periode 3 bulan dapat dilihat pada tabel berikut :

Tabel 4.26 Kondisi Curah Hujan Per Bulan

Unsur Cuaca	Curah Hujan	Jumlah		
		Jan	Feb	Mar
Barat	Curah Hujan Rendah	5	5	5
Kecepatan angin agak tinggi	Curah Hujan Rendah	1	1	1
Kelembaban agak tinggi	Curah Hujan Rendah	3	2	2
Temperatur agak tinggi	Curah Hujan Rendah	4	1	1

Jumlah curah hujan untuk 3 bulan dapat dilihat pada tabel berikut :

Tabel 4.27 Operasi *Roll-up*

Unsur Cuaca	Curah Hujan	Jumlah
Barat	Curah Hujan Rendah	15
Kecepatan angin agak tinggi	Curah Hujan Rendah	3
Kelembaban agak tinggi	Curah Hujan Rendah	7
Temperatur agak tinggi	Curah Hujan Rendah	6

Berikut *query* untuk menampulkan fungsi *roll up* tersebut :

```
select Tahun,
case when Bulan = '1' then 'Jan'
when Bulan = '2' then 'Feb'
when Bulan = '3' then 'Mar'
else 'Total' end as Bulan,
ifnull(unsur_cuaca, 'Total') as Unsur_Cuaca,
Curah_Hujan, coalesce(SUM(Jumlah), 0) as Jumlah
from fakta_kondisi as f
join dim_cuaca as dc on f.kode_cuaca = dc.kode_cuaca
join dim_waktu as dw on f.kode_waktu = dw.kode_waktu
where Curah_hujan = 'Curah hujan rendah' AND bulan
between '1' AND '3'
group by dc.unsur_cuaca,dw.bulan with rollup
```

2. *Drill down*

Operator *drill-down* merupakan kebalikan dari *roll-up*. Penggunaan *query* untuk data jumlah curah hujan yang dimensi waktunya 3 bulanan, dapat dibagi ke dalam data jumlah curah hujan per bulan. Tabel jumlah curah hujan rendah dalam 3 bulan dapat dilihat pada tabel berikut :

Tabel 4.28 Kondisi Curah Hujan 3 Bulan

Unsur Cuaca	Curah Hujan	Jumlah
Barat	Curah Hujan Rendah	15
Kecepatan angin agak tinggi	Curah Hujan Rendah	3
Kelembaban agak tinggi	Curah Hujan Rendah	7
Temperatur agak tinggi	Curah Hujan Rendah	6

Jumlah curah hujan per bulan dapat dilihat pada tabel berikut :

Tabel 4.29 Operasi *Drill-down*

Unsur Cuaca	Curah Hujan	Jumlah		
		Jan	Feb	Mar
Barat	Curah Hujan Rendah	5	5	5
Kecepatan angin agak tinggi	Curah Hujan Rendah	1	1	1
Kelembaban agak tinggi	Curah Hujan Rendah	3	2	2
Temperatur agak tinggi	Curah Hujan Rendah	4	1	1

Berikut *query* untuk menampulkan fungsi *drill down* tersebut :

```
select Tahun,
case when Bulan = '1' then 'Jan'
when Bulan = '2' then 'Feb'
when Bulan = '3' then 'Mar'
else 'Total' end as Bulan,
ifnull(unsur_cuaca, 'Total') as Unsur_Cuaca,
Curah_Hujan, coalesce(SUM(Jumlah), 0) as Jumlah
from fakta_kondisi as f
join dim_cuaca as dc on f.kode_cuaca = dc.kode_cuaca
join dim_waktu as dw on f.kode_waktu = dw.kode_waktu
where Curah_hujan = 'Curah hujan rendah' AND bulan
between '1' AND '3'
group by dc.unsur_cuaca,dw.bulan
```

3. Slicing

Operator *slice* adalah proses mengambil potongan kubus berdasarkan nilai tertentu pada satu atau beberapa dimensinya. Contoh data tidak hujan

dan curah hujan sedang pada bulan Januari hingga Februari tahun 2005 dapat dilihat pada tabel berikut :

Tabel 4.30 Curah Hujan Bulan Januari dan Februari

		Jumlah	
Curah Hujan	**Bulan**	**Temperatur Agak Tinggi**	**Temperatur Sedang**
Tidak Hujan	Januari	6	1
	Februari	1	1
Curah Hujan sedang	Januari	1	2
	Februari	4	4

Misal hanya ingin mengambil data Bulan Januari. Hasilnya dapat dilihat pada tabel berkut :

Tabel 4.31 Curah Hujan Bulan Januari

		Jumlah	
Nama Cuaca		**Temperatur Agak Tinggi**	**Temperatur Sedang**
Tidak Hujan		6	1
Curah Hujan Sedang		1	2

Berikut *query* untuk menampulkan fungsi *slicing* tersebut :

```
select Tahun,Bulan,Unsur_cuaca,Curah_hujan, Jumlah
from fakta_kondisi as f join dim_cuaca as dc join
dim_waktu as dw
where Curah_hujan = 'Tidak hujan' and Nama_cuaca =
```

```
    'Temperatur agak tinggi' and Bulan between '1' and
    '2'
    and f.kode_cuaca = dc.kode_cuaca and f.kode_waktu =
    dw.kode_waktu
    order by curah_hujan desc
```

4.2.2.1 Kesimpulan dan Hasil Pengujian *Data Mart*

Berdasarkan hasil pengujian dengan kasus uji *data mart* dengan *OLAP* diatas dapat ditarik kesimpulan bahwa pengujian *data mart* menghasilkan jumlah curah hujan berdasarkan unsur cuaca dan bulan.