BAB IV
IMPLEMENTASI DAN PENGUJIAN SISTEM

IV.1 Implementasi Sistem

Tahap implemenasi sistem merupakan tahap untuk mengaplikasikan apa yang telah dirancang pada tahap perancangan sistem berdasarkan hasil analisis sistem. Tahapan ini dilakukan setelah tahapan analisis dan perancangan selesai dilakukan, kemudian diimplementasikan pada bahasa pemrograman untuk diterapkan pada keadaan yang sesungguhnya. Hasil dari penerapan tersebut diharapkan dapat menjadi sebuah sistem yang siap diuji dan digunakan.

IV.1.1 Implementasi Perangkat Keras

Perangkat keras yang akan digunakan untuk mengimplementasikan sistemini adalah sebagai berikut :

1. *Processor* : AMD C-60 APU
2. *Harddisk* : 320GB
3. *Memory* : 2048MB DDR3
4. *Monitor* : 14”

IV.1.2 Implementasi Perangkat Lunak

Perangkat lunak yang akan digunakan untuk mengimplementasikan sistem ini adalah sebagai berikut :

Tabel IV-1 Tabel Perangkat Lunak Yang Digunakan

Perangkat Lunak	Keterangan
Microsoft Xp	Sebagai Sistem Operasi
MySQL	Sebagai *Database*
HeidiSQL	Sebagai *Interface Database*
Delphi XE5	Sebagai *Code Editor*

IV.1.3 Implementasi Basis Data

Data yang dibutuhkan pada sistem ini dibuat menjadi tabel–tabel yang dimasukkan kedalam database. DBMS yang digunakan untuk mengelolah data-data tersebut adalah MySQL.

Tabel IV-2 Implementasi Basis Data

No	Nama Tabel	Nama File
1	preprocessing	preprocessing.sql
2	temp_transaksi	temp_transaksi.sql
3	ck	ck.sql
4	lk	lk.sql
5	confidence	confidence.sql
6	korelasi	korelasi.sql
7	periode	periode.sql
8	laporan	laporan.sql
9	transaksi	transaksi.sql
10	detail_transaksi	detail_transaksi.sql
11	support	support.sql

a. Tabel preprocessing

```
CREATE TABLE IF NOT EXISTS `preprocessing` (
`id_transaksi` char(8) DEFAULT NULL,
`id_barang` char(8) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

b. Tabel temp_transaksi

```
CREATE TABLE IF NOT EXISTS `temp_transaksi` (
`id_transaksi` char(8) DEFAULT NULL,
`Tanggal` date DEFAULT NULL,
`id_pelanggan` char(6) DEFAULT NULL,
`pelanggan` varchar(50) DEFAULT NULL,
`id_barang` char(8) DEFAULT NULL,
`nama_barang` varchar(50) DEFAULT NULL,
`Id_Supplier` char(6) DEFAULT NULL,
`Supplier` varchar(50) DEFAULT NULL,
`Qty` int(3) DEFAULT NULL,
```

```
`Harga` int(10) DEFAULT NULL,
`Total_Bayar` int(10) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

c. Tabel ck

```
CREATE TABLE IF NOT EXISTS `ck` (
`id_barang` char(8) DEFAULT NULL,
`supp_count` int(5) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

d. Tabel lk

```
CREATE TABLE IF NOT EXISTS `lk` (
`id_barang` char(8) DEFAULT NULL,
`supp_count` int(5) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

e. Tabel confidence

```
CREATE TABLE IF NOT EXISTS `confidence` (
`itemset` varchar(100) DEFAULT NULL,
`nilaikepastian` decimal(5,2) DEFAULT '0.00'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

f. Tabel korelasi

```
CREATE TABLE IF NOT EXISTS `korelasi` (
`itemset` varchar(100) DEFAULT NULL,
`nilai_keterhubungan` decimal(5,2) DEFAULT '0.00'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

g. Tabel periode

```
CREATE TABLE IF NOT EXISTS `periode` (
`TglAwal` date DEFAULT NULL,
`TglAkhir` date DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

h. Tabel laporan

```
CREATE TABLE IF NOT EXISTS `laporan` (
`id_barang` char(8) DEFAULT NULL,
`id_barang_rekomendasi` char(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

i. Tabel transaksi

```
CREATE TABLE IF NOT EXISTS `transaksi` (
`id_transaksi` char(10) NOT NULL DEFAULT '',
`Tanggal` date DEFAULT NULL,
`id_pelanggan` char(6) DEFAULT NULL,
`id_supplier` char(6) DEFAULT NULL,
PRIMARY KEY (`id_transaksi`),
KEY `id_pelanggan` (`id_pelanggan`),
KEY `id_supplier` (`id_supplier`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

j. Tabel detai_transaksi

```
CREATE TABLE IF NOT EXISTS `detail_transaksi` (
`id_transaksi` char(10) DEFAULT NULL,
`id_barang` char(8) DEFAULT NULL,
`Qty` int(3) DEFAULT NULL,
`Harga` int(10) DEFAULT NULL,
`Total_bayar` int(10) DEFAULT NULL,
KEY `id_transaksi` (`id_transaksi`),
KEY `id_barang` (`id_barang`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

k. Tabel support

```
CREATE TABLE IF NOT EXISTS `support` (
`itemset` varchar(100) DEFAULT NULL,
```

```
`nilaisupport` decimal(5,2) DEFAULT '0.00'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

IV.1.4 Implementasi Antar Muka

Implementasi antar muka dilakukan dengan menampilkan setiap tampilan sistem yang dibangun dan pengkodeannya dalam bentuk *file* program. Adapun untuk implementasi tampilan perangkat lunak *data mining* yang dibangun dapat dilihat pada Tabel IV-3 berikut:

Tabel IV-3 Implementasi Antar Muka

Antarmuka	Proses	File
Ekstraksi data	1	UEkstraksiData.pas
Periode	1	UPeriode,pas
Preprocessing data	2	UPreprocessingData.pas
Association Rule	3	UAssociationRule.pas

IV.2 Pengujian Sistem

Pengujian system merupakan tahapan untuk melakukan serangkaian tes untuk mencoba sistem yang telah dibangun dengan tujuan mengetahui bagaimana kualitas dari suatu sistem tersebut.

IV.2.1 Rancangan Pengujian

Pengujian yang akan dilakukan adalah dengan menggunakan metode *whitebox* dan *blackbox* serta pengujian beta dengan mewawancarai user yang akan menggunakan perangkat lunak ini.

Tabel IV-4 Rancangan Pengujian

No	Item Uji	Detail Pengujian	Jenis Uji
1	Algoritma Apriori	Uji Algoritma	White Box
2	Ektraksi data transaksi	Pengujian validasi ekstraksi data	Black Box
3	Input Periode	Pengujian validasi input periode	Black Box
4	Rule	Hasil dari association rule	Black Box

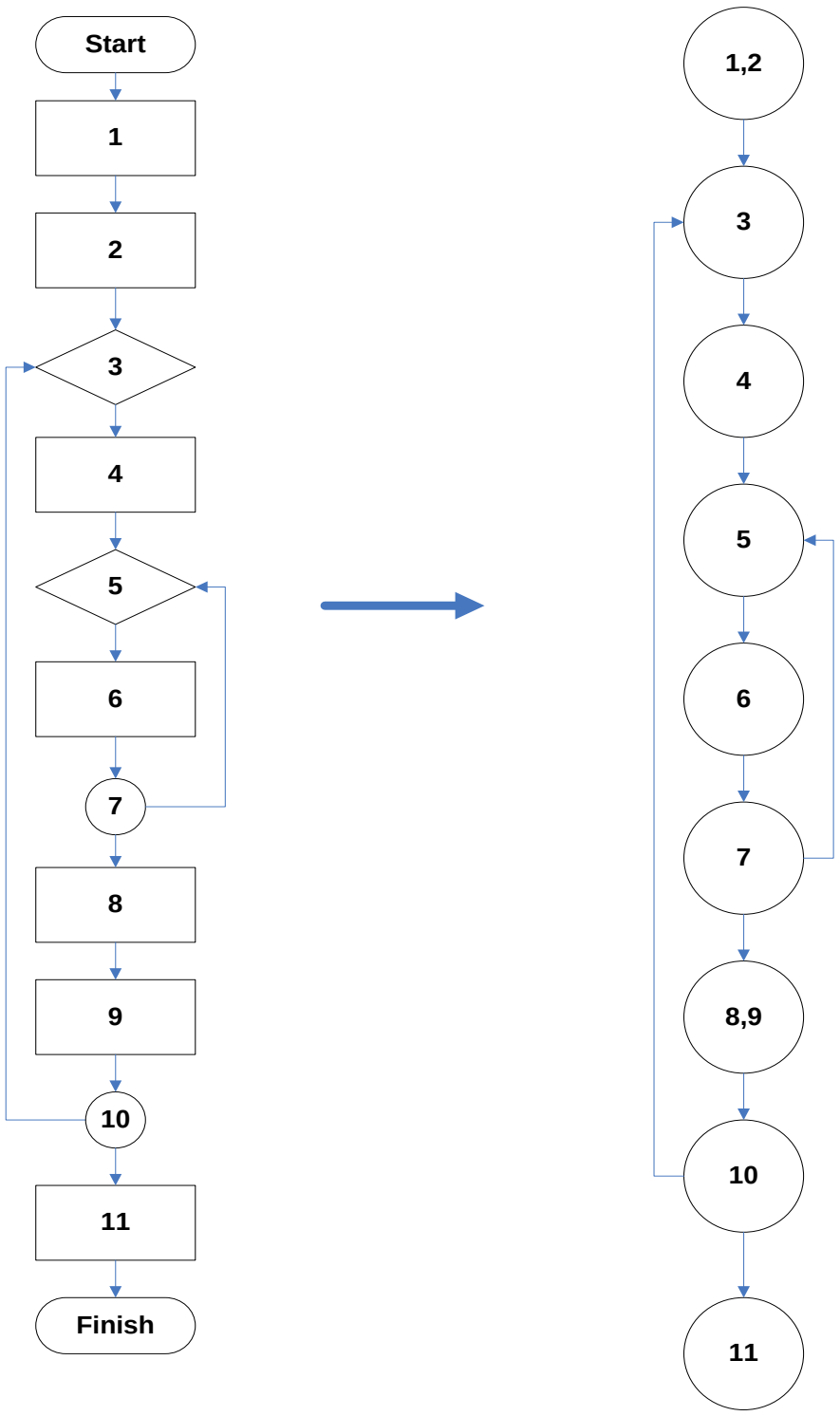
a. Pengujian White Box

Pengujian white box bertujuan untuk mengetahui kinerja logika yang dibuat pada sebuah perangkat lunak, apakah berjalan dengan baik atau tidak. Pengujian white box digunakan pada algoritma apriori untuk mengukur

kinerja logika berdasarkan *pseudocode* yang telah dibuat pada tahap analisis. Tahapan yang dilakukan yaitu mengubah *pseudocode* menjadi *flowchart* kemudian menjadi *flowgraf*.

pseudocode

k ← 2
L_1 ← { Data 1-itemset yang melebihi nilai minsup }
while L_{k-1} not empty **do**
Ck ← { Data kandidat k-itemset hasil dari penggabungan L_{k-1}
}
foreach C_k**do**
Cari banyaknya transaksi yang mengandung semua item Ck
di tabel transaksi
endfor
L_k ← { Data Ck yang memenuhi nilai minimum support }
k ← k + 1
endwhile
Hasilnya ← { Seluruh frequent itemset }



Gambar IV-1Flowchar dan Flowgraf Algoritma Apriori

1. Cyclometic Complexity

   V(G) = E - N + 2

   V(G) = 10 – 9 + 2

   V(G) = 3

2. Hitung jumlah region

   Jumlah region = Jumlah kurva tertutup + 1 (Region terluar)

   Jumlah region = 2 +1

   Jumlah region = 3

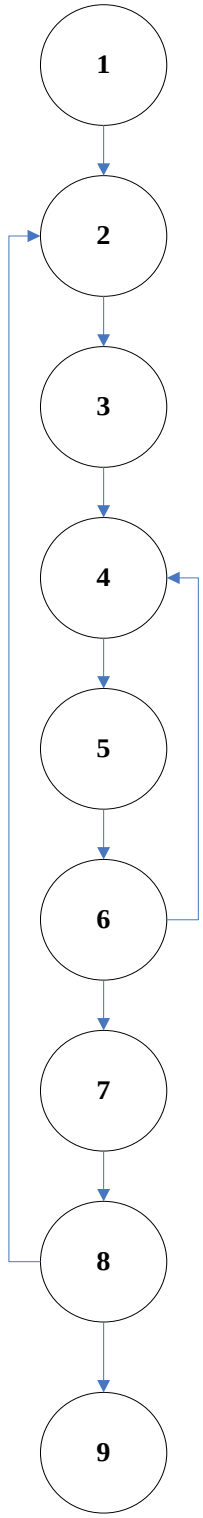
3. Hitung jumlah predicate note

   V(G) = Jumlah Predicate Note + 1

   V(G) = 2+1

   V(G) = 3

4. Penomoran ulang flowgraf



Gambar IV-2 Penggambaran Ulang Flowchart

5. Graph Matric

Tabel IV-5 Tabel Graph Matriks

Node	1	2	3	4	5	6	7	8	9	Jumlah
1		1								0
2			1							0
3				1						0
4					1					0
5						1				0
6				1			1			1
7								1		0
8		1							1	1
9										0
Jumlah										2

6. Kesimpulan pengujian *whitebox*

   Berdasarkan pengujian pada setiap metode, dihasilkan nilai *cyclomaticcomplexity* yang sama yaitu tiga. Maka dapat disimpulkan bahwa pengujian *whitebox* pada algoritma Apriori berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

b. Pengujian *BlackBox*

Pengujian *blackbox* dilakukan dengan tujuan untuk mengetahui apakah data yang dimasukkan sudah benar atau belum serta bagaimana hasil keluarannya apakah sesuai dengan harapan atau tidak. Teknik pengujian *black box* yang digunakan adalah *Equivalence Partitioning* dan *Cause-Effect Relationship*.

1. Ekstraksi data

   Tabel di bawah ini merupakan scenario pengujian menggunakan teknik pengujian Equivalence Partitioning Testing yang dilakukan pada bagian ekstraksi data.

Tabel IV-6 Pengujian Ekstraksi Data

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Alamat file excel data transaksi yang akan diekstrak	Data berhasil di*import*	Dapat menampilkan tabel yang telah diekstrak	[√] Diterima [] Ditolak
Kasus dan Uji coba (Data Kosong)			

Alamat file excel data transaksi yang akan diekstrak dikosongkan.	Muncul pesan kesalahan file excel belum dipilih	Menampilkan pesan kesalahan alamat file harus diisi.	[] Diterima [√] Ditolak

2. Pemilihan atribut data

   Tabel di bawah ini merupakan scenario pengujian menggunakan teknik pengujian *Equivalence Partitioning Testing* yang dilakukan pada bagian pemilihan atribut data.

Tabel IV-7 Pengujian Pemilihan Atribut

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memilih atribut data	Atribut data berhasil dipilih	Dapat menampilkan atribut data yang dipilih	[√] Diterima [ ] Ditolak
Kasus dan Uji coba (Data Kosong)			
Tidak memilih atribut	Mencul pesan kesalahan atribut data belum dipilih	Menampilkan pesan kesalahan atribut data belum dipilih	[ ] Diterima [√] Ditolak

3. Association rule

   Tabel di bawah ini merupakan scenario pengujian menggunakan teknik pengujian *Equivalence Partitioning Testing* yang dilakukan pada bagian *association rule*.

Tabel IV-8 Pengujian Association Rule

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nila minimum penunjang dan nilai minimum kepastian	Hasil rule dari proses data mining association rule	Dapat menampilkan rule yang dihasilkan	[√] Diterima [ ] Ditolak
Kasus dan Uji coba (Data Kosong)			
Nilai minimum penunjang dan nilai minimum kepastian tidak diisi	Muncul pesan kesalahan	Menampilkan pesan kesalahan	[ ] Diterima [√] Ditolak

c. Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung ke lapangan, dengan menggunakan wawancara mengenai kepuasan user dalam hal ini supervisor terhadap perangkat lunak yang telah dibangun. Berikut merupakan daftar pertayaan yang diajukan :

a. Apakah tampilan antar muka perankat lunak ini mudah dipelajari?
b. Apakah perangkat lunak ini mudah digunakan ?
c. Apakah perangkat lunak ini dapat membantu dalam menentukan barang apa saja yang dapat direkomendasikan ke pelanggan?
d. Bagaimana menurut anda mengenai hasil rekomendasi dari perangkat lunak ini?
e. Bagaimana menurut anda mengenai keseluruhan perangkat lunak ini?

Jawaban :

a. Ya, karena untuk tampilannya dibuat sederhana sehingga mudah dipelajari.
b. Ya, karena untuk menggunakan perangkat lunak ini tekan-tekan tombol saja.
c. Ya, karena dari perangkat lunak ini bisa diketahui barang yang dibeli bersama-sama dan berapa sering barang tersebut terjual.
d. Bagus, karena dari hasilnya bisa diketahui berapa sering barang terjual dan dibeli bersama-sama dengan barang apa.
e. Bagus, karena perangkat lunak ini dapat membantu dalam menentukan barang yang akan dijadikan paket dan perangkat lunak ini mudah dipelajari dan digunakan.

d. Pengujian Sample

Pengujian sample ini menguji perankat lunak yang telah dibangun apakah menghasilkan data yang diinginkan dan sesuai dengan hasil contoh kasus dibab III. Pengujian ini menggunakan sample data transaksi satu minggu dari

tanggal 1 juli 2012 sampai 7 juli 2012 dengan nilai minimum penunjang 3 dan nilai minimum kepastian 80%. Data hasil proses akhir dari perangkat lunak dapat dilihat pada tabel dibawah ini.

Tabel IV-9 Laporan dari perangkat lunak

Barang	Barang yang direkomendasikan	Korelasi (%)
079	040	16.67
108	040	16.67
222	040	16.67
AHT	TW	20
TW	AHT	20

IV.2.2 Kesimpulan Pengujian

Berdasarkan hasil pengujian *whitebox, blackbox* dan pengujian beta yang telah dilakukan, dapat diambil keputusan sebagai berikut :

1. Perangkat lunak yang dibangun dapat membantu pihak perusahaan merekomendasikan barang kepada pelanggannnya.
2. Perangkat lunak yangdibangun dapat menghasilkan rule sesuai dengan metode dan algoritma yang dipakai.
3. Berdasarkan pengujian pada setiap metode, dihasilkan *nilai Cyclomatic Complexity* yang sama yaitu 3. Maka dapat disimpulkan bahwa pengujian *whitebox* pada algoritma Apriori berjalan dengan baik karena setiap pengujian menghasilkan nilai yang sama.