

BAB IV

HASIL DAN PEMBAHASAN

4.1 Gambaran Umum Sistem

Pada subbab ini dibahas mengenai gambaran umum sistem penentuan pola penjualan obat dan rekomendasi pembelian obat terhadap *supplier*. Pada sistem ini dibutuhkan inputan data obat dan data transaksi penjualan obat. Data transaksi penjualan obat digunakan untuk menentukan pola penjualan obat. Pada penentuan pola penjualan obat digunakan data tiga bulan terakhir dari proses analisis dilakukan. Data tiga bulan terakhir digunakan untuk menentukan pola penjualan obat dan rekomendasi pembelian obat terhadap *supplier* dalam satu bulan. Setelah sistem menentukan pola penjualan obat maka selanjutnya sistem menentukan rekomendasi pembelian obat terhadap *supplier*. Pada penentuan rekomendasi pembelian obat, terlebih dahulu ditentukan rekomendasi stok dalam satu bulan, setelah itu ditentukan berapa jumlah setiap kali melakukan pemesanan obat dan tahap terakhir adalah menentukan kapan titik pemesanan kembali berdasarkan jumlah minimum obat yang tersisa di apotek. Rekomendasi pembelian obat yang disajikan pada sistem adalah obat – obat yang terdapat pada pola penjualan obat sehingga sebelum melakukan penentuan rekomendasi pembelian obat, terlebih dahulu harus dilakukan penentuan pola penjualan obat.

4.2 Lingkungan Perancangan dan Implementasi Sistem

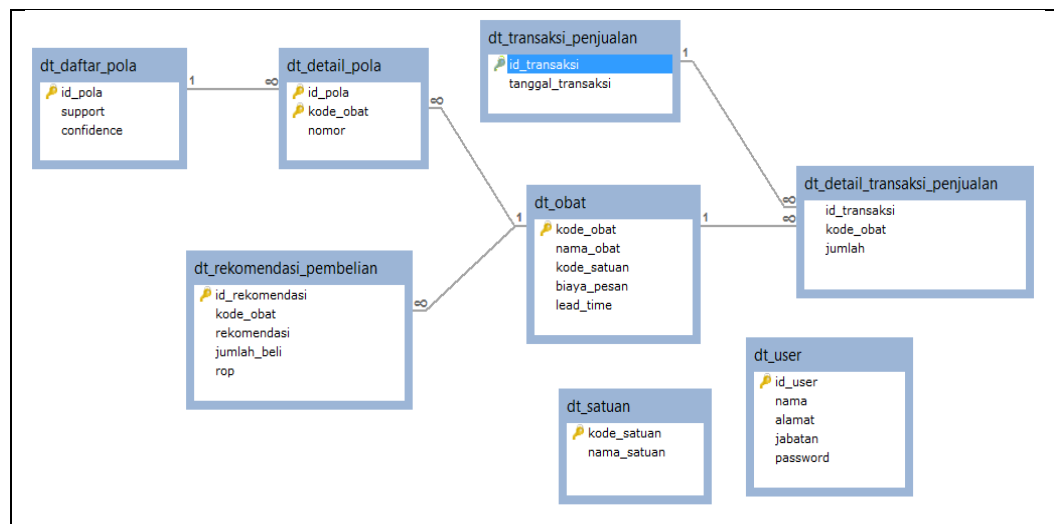
Sistem dirancang di sebuah lingkungan berbasis Sistem operasi Windows 8.1 Pro versi 64-bit. Sistem dirancang di sebuah laptop yang memiliki *Processor* Intel® Core™ i5-3630QM CPU @ 2,40GHz, 2,40GHz, RAM 4 GB dan Kapasitas Penyimpanan sebesar 1 TB.

Sistem diimplementasikan dengan menggunakan bahasa pemrograman Java. *Database Management System* (DBMS) yang digunakan pada sistem ini adalah MySQL.

Dalam perancangan dan implementasi sistem, digunakan beberapa Perangkat Lunak untuk dapat memenuhi semua kebutuhan sistem. Berikut adalah perangkat lunak yang digunakan.

1. Netbeans IDE 8.0.2
2. SQLyog Ultimate - MySQL GUI v9.01
3. Pencil Version 1.2.0
4. Microsoft Visio 2016

4.2 Implementasi Database



Gambar 4.1 Database Diagram

Gambar 4.1 merupakan *database* diagram yang digunakan dalam sistem penentuan pola penjualan dan rekomendasi pembelian obat terhadap *supplier*. Pada Implementasi basis data terdapat delapan buah tabel pada satu database sistem, adapun delapan tabel tersebut adalah sebagai berikut.

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zero fill?	Comment
<input type="checkbox"/>	kode_obat	varchar	15		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	nama_obat	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	kode_satuan	varchar	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	biaya_pesanan	int			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	lead_time	int			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Gambar 4. 2 Tabel dt_obat

Tabel data obat pada gambar 4.2 terdiri dari 5 *field* yang digunakan untuk menyimpan data – data obat berupa kode_obat sebagai *primary key*, nama_obat

untuk menyimpan nama obat, kode_satuan untuk menyimpan kode satuan obat dan *foreign key* ri tb_satuan, biaya_pesan untuk menyimpan biaya pemesanan obat dan *lead_time* untuk menyimpan waktu tunggu kedatangan obat sampai pada Apotek Anggita.

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
<input type="checkbox"/>	kode_satuan	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	nama_satuan	varchar	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Gambar 4.3 Tabel dt_satuan

Gambar 4.3 merupakan tabel dt_satuan yang terdiri dari dua *field* yaitu kode_satuan yang merupakan *primary_key* dan nama_satuan untuk menyimpan data nama satuan obat

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
<input type="checkbox"/>	id_transaksi	varchar	30		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	tanggal_transaksi	date			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Gambar 4.4 Tabel dt_transaksi_penjualan

Pada rancangan ERD terdapat entitas transaksi penjualan, berdasarkan rancangan ERD maka data transaksi penjualan dibagi menjadi dua tabel yaitu tabel dt_transaksi_penjualan dan tabel dt_detail_transaksi_penjualan. Gambar 4.4 merupakan tabel dt_transaksi_penjualan yang terdiri dari dua *field* yaitu id_transaksi yang merupakan *primary key* dan tanggal transaksi untuk menyimpan data tanggal terjadinya transaksi tersebut.

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
<input type="checkbox"/>	id_transaksi	varchar	30		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	kode_obat	varchar	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	jumlah	int	11		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Gambar 4.5 Tabel dt_detail_transaksi_penjualan

Gambar 4.4 merupakan tabel dt_transaksi_penjualan yang terdiri dari tiga *field* yaitu id_transaksi yang merupakan *foreign keys* dari tabel dt_transaksi_penjualan, kode_obat untuk menyimpan kode obat yang merupakan *foreign keys* dari dt_obat dan jumlah untuk menyimpan banyaknya obat yang dibeli untuk setiap obat dari masing – masing id transaksi.

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
<input type="checkbox"/>	id_pola	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	support	double			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	confidence	double			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Gambar 4. 6 Tabel dt_daftar_pola

Pada rancangan ERD terdapat entitas dafta_pola, pada implementasinya daftar_pola dibagi menjadi dua tabel yaitu dt_daftar_pola dan dt_daftar_detail_pola. Gambar 4.6 merupakan tabel dt_daftar_pola yang terdiri dari tiga *field* yaitu id_pola yang merupakan *primary key*, support yang menyimpan data support dari pola yang dihasilkan dan confidence yang menyimpan data confidence dari pola yang dihasilkan.

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
<input type="checkbox"/>	id_pola	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	kode_obat	varchar	100		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	nomor	int	11		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Gambar 4. 7 Tabel dt_detail_daftar_pola

Gambar 4.7 merupakan tabel dt_detail_daftar_pola yang terdiri dari tiga *field* yaitu id_pola yang merupakan *primary key*, kode_obat untuk menyimpan data obat yang memenuhi kriteria untuk pola penjualan dan *foreign key* dari tabel dt_obat dan nomor untuk menyimpan data nomor yang nantinya digunakan untuk penyusunan kalimat di poal penjualan ketika akan ditampilkan pada sistem.

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
<input type="checkbox"/>	id_rekomendasi	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	kode_obat	varchar	50		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	rekomendasi	int	11		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	jumlah_beli	int	11		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	rop	int	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	tac	int	11		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Gambar 4. 8 Tabel dt_rekomendasi_pembelian

Gambar 4.8 merupakan tabel dt_rekomendasi_pembelian yang terdiri dari enam *field* yaitu id_rekomendasi yang merupakan *primary key*, kode_obat yang menyimpan data kode obat yang terdapat pada pola penjualan, rekomendasi untuk menyimpan data rekomendasi stok yang harus disediakan oleh Apotek Anggita dalam satu bulan, jumlah_beli menyimpan jumlah obat yang dibeli setiap kali pemesanan, rop menyimpan data jumlah minimum yang harus tersedia sehingga

ketika telah mencapai jumlah minimum maka pihak apotek harus melakukan pemesanan obat kembali dan tac menyimpan data biaya yang dikeluarkan setiap kali melakukan pemesanan.

<input type="checkbox"/>	Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Zerofill?	Comment
<input type="checkbox"/>	id_user	varchar	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	nama	varchar	30		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	alamat	varchar	30		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	jabatan	varchar	10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	password	varchar	20		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

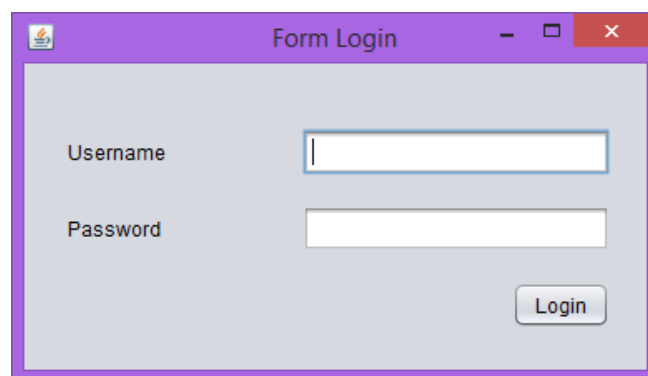
Gambar 4. 9 Tabel dt_user

Gambar 4.9 merupak tabel dt_user yang terdiri dari lima *field* yaitu id_user yang merupakan *primary key*, nama untuk menyimpan nama *user*, alamat untuk menyimpan alamat *user*, jabatan untuk menyimpan jabatan *user* dan *password* untuk menyimpan *password* yang digunakan untuk login ke sistem.

4.3 Implementasi Antarmuka

Implementasi perancangan antarmuka sistem penentuan pola penjualan obat dan rekomendasi pembelian obat terhadap *supplier* diimplementasikan berbasis *desktop* dengan menggunakan bahasa pemrograman java. Sistem ini nantinya digunakan oleh pegawai dan apoteker dari Apotek Anggita.

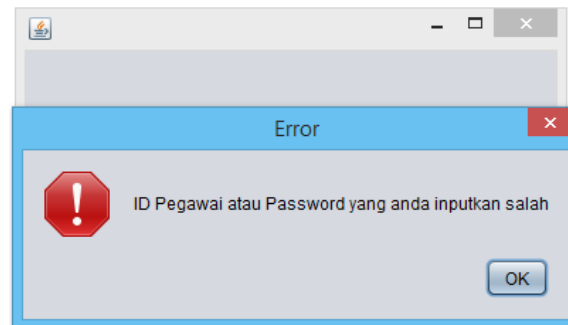
4.3.1 Antarmuka Tampilan Login



Gambar 4. 10 Antarmuka Login User

Gambar 4.10 merupakan antarmuka untuk *login user* ke sistem. Pada halaman ini *user* akan mengisi *username* dan *password* terlebih dahulu untuk dapat

mengakses halaman utama admin. Apabila *username* dan *password* yang diinputkan oleh *user* sesuai dengan yang tersimpan pada *database* maka *user* akan dapat mengakses halaman utama admin namun jika *username* dan *password* yang diinputkan tidak sesuai maka sistem akan mengeluarkan notifikasi bahwa *username* dan *password* yang diinputkan tidak sesuai.



Gambar 4. 11 Antarmuka Notifikasi Kesalahan *Username* atau *Password*

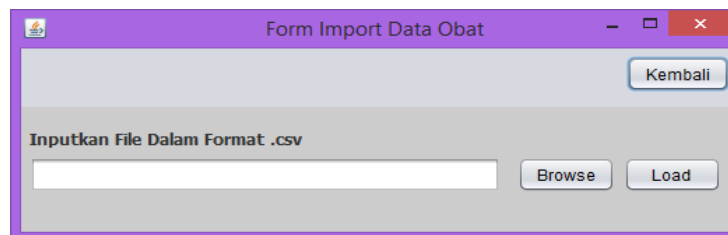
Gambar 4.11 merupakan antarmuka notifikasi kesalahan *username* atau *password*. Apabila notifikasi ini muncul maka *user* disarankan untuk menginputkan *username* dan *password* kembali sehingga *user* dapat mengakses halaman utama admin.

4.3.2 Antarmuka *Form* Utama Admin Menu Data Obat

Kode Obat	Nama Obat	Satuan	Biaya Pemesanan	Lead Time	Edit	Hapus
AG0001	GINCORE CAPSUL	Amp	70000	3	EDIT	HAPUS
AG0002	ADALAT ORDS 30	Tablet	70000	3	EDIT	HAPUS
AG0003	BACTRAZ II	Vial	70000	3	EDIT	HAPUS
AG0004	PRIMPERAN II	Ampul	70000	3	EDIT	HAPUS
AG0005	FAKTU CREAM 20 GIM	Tube	70000	3	EDIT	HAPUS
AG0006	IV CATHETER INFLO NO. 20 ONE MED	Pcs	70000	3	EDIT	HAPUS
AG0007	IV CATHETER INFLO NO. 22 ONE MED	Pcs	70000	3	EDIT	HAPUS
AG0008	ALKOHOL 70 % 100 ML	Btl	70000	3	EDIT	HAPUS
AG0009	ALKOHOL 70 % 300 ML	Btl	70000	3	EDIT	HAPUS
AG0010	KASA STERIL BUNGA BIRU	Kotak	70000	3	EDIT	HAPUS
AG0011	ALKOHOL 95 % 100 ML	Btl	70000	3	EDIT	HAPUS
AG0012	ELASTOMUL 10 CM X 4 M	Cm	70000	3	EDIT	HAPUS
AG0013	ELASTOMUL 8 CM X 4 M	Cm	70000	3	EDIT	HAPUS
AG0014	ELASTOMUL 12 CM X 4 M	Cm	70000	3	EDIT	HAPUS
AG0015	FOLLEY CATETHER 14 RUSCH	Pcs	70000	3	EDIT	HAPUS
AG0016	FOLLEY CATETHER 16 RUSCH	Pcs	70000	3	EDIT	HAPUS
AG0017	FOLLEY CATETHER CH 18 RUSCH	Pcs	70000	3	EDIT	HAPUS
AG0018	SOFRAH 10 CM	Roll	70000	3	EDIT	HAPUS
AG0019	SPINOCAN 270 X 1/2 B BRAUN	Pcs	70000	3	EDIT	HAPUS
AG0020	TENSORPLAST S.T.K ADULT	Pcs	70000	3	EDIT	HAPUS
AG0021	TENSOCOREPE FL 10 CM X 4,5 M	Roll	70000	3	EDIT	HAPUS
AG0022	TENSOCOREPE 15 CM	Roll	70000	3	EDIT	HAPUS
AG0023	ULTRAPX 5 CM X 1 M	Cm	70000	3	EDIT	HAPUS
AG0024	URINE BAG TERUMO	Pcs	70000	3	EDIT	HAPUS
AG0025	KASA 5 CM CM	Pcs	70000	3	EDIT	HAPUS

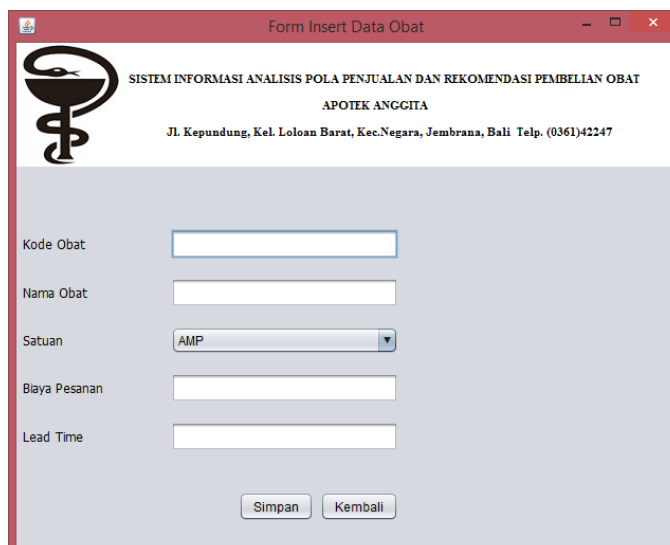
Gambar 4.12 Antarmuka *Form* Utama Admin Menu Data Obat

Gambar 4.12 merupakan antarmuka *form* utama admin menu data obat. Pada halaman ini akan ditampilkan data obat dari Apotek Anggita. Pada bagian atas halaman ini terdapat tiga *button* yaitu *button import data*, *button insert data* dan *button log out*. *Button import* berfungsi untuk mengarahkan *user* ke halaman *import data*. *Button insert data* berfungsi untuk mengarahkan *user* ke halaman penginputan data obat. *Button log out* berfungsi untuk keluar dari halaman utama admin.



Gambar 4.13 Antarmuka *Import Data Obat*

Pada halaman utama admin, apabila *user* memilih *button import data* maka akan ditampilkan halaman seperti pada gambar 4.13. Pada halaman ini *user* dapat melakukan penginputan data obat dengan format csv ke dalam *database*. *Button browse* berfungsi untuk memilih data yang akan *diimport* ke *database*, sedangkan *button load* berfungsi untuk meload data ke dalam *database* dan *button log out* berfungsi untuk kembali ke halaman utama admin.



Gambar 4.14 Antarmuka *Insert Data Obat*

Gambar 4.14 merupakan antarmuka *insert* data obat. Halaman ini berfungsi untuk menginputkan data obat ke dalam *database*. Pada halaman ini terdapat beberapa *field* yang harus diisi oleh *user* yang meliputi nama obat, satuan, biaya pemesanan dan *lead time* (waktu tunggu kedatangan obat). *Button* simpan berfungsi untuk menyimpan data obat yang sebelumnya sudah diisi oleh *user* kedalam *database*, jika data yang diinputkan berhasil disimpan maka akan keluar notifikasi seperti pada gambar 4.15. *Button* kembali berfungsi untuk kembali ke halaman utama admin.

Gambar 4.15 Antarmuka Data Obat Berhasil Diinputkan

4.3.3 Antarmuka Form Utama Admin Menu Transaksi Penjualan

Id Transaksi	Tanggal Transaksi	Kode Transaksi	Nama	Satuan	Jumlah	Edit	Hapus
TR0000022028	2015-01-01	AG0041	INFUS SET DEWASA TERUMO	Pcs	1	EDIT	HAPUS
TR0000022027	2015-01-01	AG0045	IV CATHETER NO. 20 G TERUMO	Pcs	1	EDIT	HAPUS
TR0000022027	2015-01-01	AG0098	THREE WAY STOP COK JMS BEREKOR	Pcs	1	EDIT	HAPUS
TR0000022027	2015-01-01	AG0041	INFUS SET DEWASA TERUMO	Pcs	1	EDIT	HAPUS
TR0000022027	2015-01-01	AG0917	INFUS RL 500 ML SABRE	Flash	1	EDIT	HAPUS
TR0000022027	2015-01-01	AG0113	ID BAND ADULT BLUE WICARD	Pcs	1	EDIT	HAPUS
TR0000022027	2015-01-01	AG0819	ULCERANIN LU	Ampul	1	EDIT	HAPUS
TR0000022027	2015-01-01	AG0951	SARIMOL SUSPENSIS SP	Botol	1	EDIT	HAPUS
TR0000022027	2015-01-01	AG0085	SPUT 3 CC TERUMO	Pcs	10	EDIT	HAPUS
TR0000022027	2015-01-01	AG0815	OTTOZOL LU	Vial	1	EDIT	HAPUS
TR0000022027	2015-01-01	AG0086	SPUT 5 CC TERUMO	Pcs	1	EDIT	HAPUS
TR0000022028	2015-01-01	AG0195	INFUS RL WIGA	Flash	3	EDIT	HAPUS
TR0000022028	2015-01-01	AG0815	OTTOZOL LU	Vial	2	EDIT	HAPUS
TR0000022028	2015-01-01	AG0093	SPUT 5 CC ONE MED	Pcs	2	EDIT	HAPUS
TR0000022028	2015-01-01	AG0925	AQUA PI 25 ML	Botol	1	EDIT	HAPUS
TR0000022029	2015-01-01	AG0138	OXYGEN HASIL CANRULA DEWASA (OXYFLOW)	Pcs	1	EDIT	HAPUS
TR0000022029	2015-01-01	AG0526	NEUROSANBE 5000 MG TAB	Tab	10	EDIT	HAPUS
TR0000022029	2015-01-01	AG0929	ONOSGIN	ML	120	EDIT	HAPUS
TR0000022030	2015-01-01	AG0511	KETOROLAC LU 30 MG	Ampul	1	EDIT	HAPUS
TR0000022031	2015-01-01	AG0045	VOLOX INFUS	Flash	1	EDIT	HAPUS
TR0000022032	2015-01-01	AG081	ROZLER LU	Ampul	2	EDIT	HAPUS
TR0000022032	2015-01-01	AG0093	SPUT 5 CC ONE MED	Pcs	2	EDIT	HAPUS
TR0000022033	2015-01-01	AG0952	SARIMOL SPUT	Botol	1	EDIT	HAPUS
TR0000022033	2015-01-01	AG0492	CESIVAM 100 MG TAB	Tab	10	EDIT	HAPUS
TR0000022033	2015-01-01	AG0811	SARIMOL TAB	Tab	10	EDIT	HAPUS

Gambar 4. 16 Antarmuka Form Utama Admin Menu Transaksi Penjualan Obat

Gambar 4.16 merupakan antarmuka *form* utama admin menu transaksi penjualan. Pada halaman ini akan ditampilkan data transaksi penjualan dari Apotek Anggita. Pada bagian atas halaman ini terdapat tiga *button* yaitu *button import data*, *button insert data* dan *button log out*. *Button import* berfungsi untuk megarahkan *user* ke halaman *import data*. *Button insert data* berfungsi untuk mengarahkan *user* ke halaman penginputan data transaksi penjualan obat. *Button* pola penjualan obat akan di *disable* sebelum data transaksi penjalan diinputkan. *Button log out* berfungsi untuk keluar dari halaman utama admin.

Gambar 4. 17 Antarmuka *Import Data Transaksi Penjualan Obat*

Pada halaman utama admin, apabila *user* memilih *button import data* maka akan ditampilkan halaman seperti pada gambar 4.17. Pada halaman ini *user* dapat melakukan penginputan data transaksi penjualan obat dengan format csv ke dalam *database*. *Button browse* berfungsi untuk memilih data yang akan *diimport* ke *database*, sedangkan *button load* berfungsi untuk meload data ke dalam *database* dan *button log out* berfungsi untuk kembali kehalaman utama admin.

Gambar 4. 18 Antarmuka *Insert Data Transaksi Penjualan Obat*

Gambar 4.18 merupakan antarmuka *insert* data transaksi penjualan obat. Halaman ini berfungsi untuk menginputkan data transaksi penjualan obat ke dalam *database*. Pada halaman ini terdapat beberapa *field* yang harus diisi oleh *user* yang meliputi tanggal transaksi, kode obat dan jumlah. *Button* simpan berfungsi untuk menyimpan data transaksi penjualan obat yang sebelumnya sudah diisi oleh *user* kedalam *database*. *Button* Pola Penjualan berfungsi untuk mengarahkan *user* ke halaman pola penjualan obat. *Button* kembali berfungsi untuk kembali ke halaman utama admin.

4.3.4 Antarmuka Form Pola Penjualan Obat

No	Pola Penjualan Obat	Support	Confidence	Sup x Conf
1	Jika membeli INFUS SET DEWASA TERUMO maka membeli THREE WAY STOP COK JMS BEREKOR	4.64%	93.14%	4.32%
2	Jika membeli WING NEEDLE 25G TERUMO maka membeli SPUT 3 CC ONE MED	4.25%	87.88%	3.74%
3	Jika membeli THREE WAY STOP COK JMS BEREKOR maka membeli INFUS SET DEWASA TERUMO	4.64%	74.22%	3.45%
4	Jika membeli ID BAND ADULT BLUE WICARD maka membeli THREE WAY STOP COK JMS BEREKOR	2.59%	94.64%	2.45%
5	Jika membeli IV CATHETER NO. 20 G TERUMO maka membeli THREE WAY STOP COK JMS BEREKOR	2.59%	85.43%	2.21%
6	Jika membeli IV CATHETER NO. 22 TERUMO, INFUS SET DEWASA TERUMO maka membeli THREE WAY STOP COK JMS BEREKOR	2.05%	97.67%	2.01%
7	Jika membeli INFUS SET DEWASA TERUMO, IV CATHETER NO. 20 G TERUMO maka membeli THREE WAY STOP COK JMS BEREKOR	2.15%	91.67%	1.97%
8	Jika membeli IV CATHETER NO. 22 TERUMO maka membeli THREE WAY STOP COK JMS BEREKOR	2.2%	84.91%	1.87%
9	Jika membeli IV CATHETER NO. 20 G TERUMO maka membeli INFUS SET DEWASA TERUMO	2.35%	77.42%	1.82%
10	Jika membeli ID BAND ADULT PINK WICARD maka membeli THREE WAY STOP COK JMS BEREKOR	1.86%	97.44%	1.81%
11	Jika membeli RANTIDON U 25 MG maka membeli SPUT 3 CC ONE MED	1.96%	90.91%	1.78%
12	Jika membeli ID BAND ADULT BLUE WICARD maka membeli INFUS SET DEWASA TERUMO	2.2%	80.36%	1.77%
13	Jika membeli IV CATHETER NO. 22 TERUMO maka membeli INFUS SET DEWASA TERUMO	2.1%	81.13%	1.71%
14	Jika membeli ID BAND ADULT BLUE WICARD maka membeli INFUS SET DEWASA TERUMO, THREE WAY STOP COK JMS BEREKOR	2.1%	76.79%	1.61%
15	Jika membeli INFUS RL 500 ML SANBE maka membeli THREE WAY STOP COK JMS BEREKOR	1.96%	81.63%	1.6%
16	Jika membeli ID BAND ADULT PINK WICARD maka membeli INFUS SET DEWASA TERUMO	1.61%	84.62%	1.36%
17	Jika membeli ID BAND ADULT PINK WICARD maka membeli INFUS SET DEWASA TERUMO, THREE WAY STOP COK JMS BEREKOR	1.61%	84.62%	1.36%
18	Jika membeli ONDANSETRON 4MG/2ML U maka membeli SPUT 3 CC ONE MED	1.22%	100.0%	1.22%
19	Jika membeli INFUS RL WIDA maka membeli THREE WAY STOP COK JMS BEREKOR	1.37%	87.5%	1.2%
20	Jika membeli INFUS RL 500 ML SANBE maka membeli INFUS SET DEWASA TERUMO	1.66%	69.39%	1.15%
21	Jika membeli ACRA N U maka membeli SPUT 3 CC ONE MED	1.27%	69.66%	1.14%
22	Jika membeli INFUS SET DEWASA TERUMO, SPUT 3 CC ONE MED maka membeli THREE WAY STOP COK JMS BEREKOR	1.17%	96.0%	1.13%
23	Jika membeli INFUS RL 500 ML SANBE maka membeli SPUT 3 CC ONE MED, THREE WAY STOP COK JMS BEREKOR	1.61%	67.35%	1.09%
24	Jika membeli AQUA PI 25ML OTSUKA maka membeli SPUT 1CC OM	1.22%	86.21%	1.05%
25	Jika membeli AQUA PI 25ML OTSUKA, RL WIDA 500ML maka membeli SPUT 1CC OM	.98%	100.0%	.98%
26	Jika membeli ACRA N U maka membeli WING NEEDLE 25G TERUMO, SPUT 3 CC ONE MED	1.17%	82.76%	.97%

Gambar 4.19 Antarmuka *Form* Utama Admin Menu Pola Penjualan Obat

Gambar 4.19 merupakan halaman yang menampilkan pola penjualan obat. Pada bagian atas halaman ini terdapat *button* analisis yang berfungsi untuk melakukan analisis pola penjualan obat. Sebelum melakukan analisis maka *user* harus menginputkan data obat dan data transaksi penjualan obat terlebih dahulu. Selain *button* analisis juga terdapat *button* rekomendasi, *button* ini akan di *enable* jika *user* telah melakukan analisis pola penjualan. *Button* rekomendasi berfungsi untuk mengarahkan *user* ke halaman rekomendasi pembelian obat terhadap *supplier*. Setelah *user* berhasil melakukan analisis data maka data pola penjualan akan ditampilkan seperti gambar 4.19.

4.3.5 Antarmuka Form Utama Admin Menu Rekomendasi Pembelian Obat

Gambar 4.20 merupakan halaman rekomendasi pembelian obat. Pada bagian atas halaman ini terdapat *button* analisis yang berfungsi untuk melakukan analisis rekomendasi pembelian obat. Sebelum melakukan analisis maka *user* harus menginputkan data obat, data transaksi penjualan obat dan menganalisis pola penjualan obat terlebih dahulu, karena data rekomendasi pembelian yang disajikan oleh sistem adalah data obat yang terdapat pada pola penjualan.

No	Nama Obat	Rekomendasi Stok	Nama Satuan	Jumlah Pembelian	Reorder Point	Keterangan
1	IBION MULTIVITAMIN	305	Capsul	49	30	Belum Dibeli
2	RELACTA DHA FOR MOTHER	252	Softcap	45	25	Belum Dibeli
3	ASAM MEFENAMAT 500MG TAB	191	Tablet	39	19	Belum Dibeli
4	CIPROFLOXACIN TABLET	120	Tablet	31	12	Belum Dibeli
5	SPUIT 3 CC ONE MED	119	Pcs	31	11	Belum Dibeli
6	THREE WAY STOP COK JMS BEREKOR	84	Pcs	26	8	Sudah Dibeli
7	SPUIT 5 CC ONE MED	76	Pcs	25	7	Sudah Dibeli
8	INFUS SET DEWARSA TERJUMO	74	Pcs	24	7	Belum Dibeli
9	INFUS RL 500 ML SANBE	53	Flash	21	5	Belum Dibeli
10	ID BAND ADULT BLUE WICARD	46	Pcs	19	4	Belum Dibeli
11	INFUS NS SANBE	45	Flash	19	4	Belum Dibeli
12	ALKOHOL ONE SWAB	40	Pcs	18	4	Belum Dibeli
13	WING NEEDLE 25G TERJUMO	40	Pcs	18	4	Belum Dibeli
14	ID BAND ADULT PINK WICARD	35	Pcs	17	3	Belum Dibeli
15	IV CATHETER NO. 20 G TERJUMO	35	Pcs	17	3	Belum Dibeli
16	IV CATHETER NO. 22 TERJUMO	32	Pcs	16	3	Belum Dibeli
17	ACRAN LU	31	Ampul	16	3	Belum Dibeli
18	LIDOCAN 2 % PHAPROS	25	Ampul	14	2	Belum Dibeli
19	SPUIT 3 CC OM	25	Pcs	14	2	Belum Dibeli
20	SPUIT 100 OM	24	Pcs	14	2	Belum Dibeli
21	IV CATHETER 20 G ONE MED	24	Pcs	14	2	Belum Dibeli
22	RANTIDON LU 25 MG	22	Ampul	13	2	Belum Dibeli
23	SPUIT 5 CC OM	21	Pcs	13	2	Belum Dibeli
24	IV CATH 22 STERA	20	Pcs	13	2	Belum Dibeli
25	RL WIDA 500ML	20	Flash	13	2	Belum Dibeli
26	CEPOTANIME U1 GR	19	Vial	12	1	Belum Dibeli

Gambar 4.20 Form Utama Admin Menu Rekomendasi Pembelian Obat

Apabila *user* memilih *button* cetak maka akan tampil halaman seperti pada gambar 4.21. Data rekomendasi pembelian akan ditampilkan dan siap untuk dicetak.

nama_obat	rekomendasi	nama_satuan	jumlah_beli	rop	keterangan
IBION MULTIVITAMIN	305	Capsul	49	30	Belum Dibeli
RELACTA DHA FOR MOTHER	252	Softcap	45	25	Belum Dibeli
ASAM MEFENAMAT 500MG TAB	191	Tablet	39	19	Belum Dibeli
CIPROFLOXACIN TABLET	120	Tablet	31	12	Belum Dibeli
SPUIT 3 CC ONE MED	119	Pcs	31	11	Belum Dibeli
THREE WAY STOP COK JMS BEREKOR	84	Pcs	26	8	Sudah Dibeli
SPUIT 5 CC ONE MED	76	Pcs	25	7	Sudah Dibeli
INFUS SET DEWARSA TERJUMO	74	Pcs	24	7	Belum Dibeli
INFUS RL 500 ML SANBE	53	Flash	21	5	Belum Dibeli
ID BAND ADULT BLUE WICARD	46	Pcs	19	4	Sudah Dibeli
INFUS NS SANBE	45	Flash	19	4	Sudah Dibeli
ALKOHOL ONE SWAB	40	Pcs	18	4	Belum Dibeli
WING NEEDLE 25G TERJUMO	40	Pcs	18	4	Belum Dibeli
ID BAND ADULT PINK WICARD	35	Pcs	17	3	Belum Dibeli
IV CATHETER NO. 20 G TERJUMO	35	Pcs	17	3	Belum Dibeli
IV CATHETER NO. 22 TERJUMO	32	Pcs	16	3	Belum Dibeli
ACRAN LU	31	Ampul	16	3	Belum Dibeli
LIDOCAN 2 % PHAPROS	25	Ampul	14	2	Belum Dibeli
SPUIT 3 CC OM	25	Pcs	14	2	Belum Dibeli
SPUIT 100 OM	24	Pcs	14	2	Belum Dibeli
IV CATHETER 20 G ONE MED	24	Pcs	14	2	Belum Dibeli
RANTIDON LU 25 MG	22	Ampul	13	2	Belum Dibeli

Gambar 4. 21 Form Cetak Data Rekomendasi

4.3.6 Antarmuka Form Utama User

SISTEM INFORMASI ANALISIS POLA PENJUALAN DAN REKOMENDASI PEMBELIAN OBAT
APOTEK ANGGITA II
 Jl. Kepundung, Kel. Loloan Barat, Kec.Negara, Jembrana, Bali Telp. (0361)42247

Data Obat Pola Penjualan Rekomendasi Pembelian

Cari Log Out

Pola Penjualan

Jika membeli WING NEEDLE 25G TERUMO maka membeli SPUIT 3 CC TERUMO
 Jika membeli IV CATHETER NO. 22 TERUMO maka membeli INFUS SET DEWASA TERUMO
 Jika membeli IV CATHETER NO. 22 TERUMO maka membeli THREE WAY STOP COK JMS BEREKOR
 Jika membeli SPUIT 5 CC ONE MED maka membeli ALKOHOL SWAB
 Jika membeli ALKOHOL SWAB maka membeli SARUNG TANGAN NON STERIL SENSI GLOVES S
 Jika membeli SARUNG TANGAN NON STERIL SENSI GLOVES S maka membeli ALKOHOL SWAB
 Jika membeli OPIGRAN U 1 MG maka membeli ALKOHOL SWAB
 Jika membeli TAXEGRAM U maka membeli ALKOHOL SWAB
 Jika membeli AQUA PI 25 ML maka membeli ALKOHOL SWAB
 Jika membeli INFUS SET DEWASA TERUMO maka membeli IV CATHETER NO. 20 G TERUMO
 Jika membeli IV CATHETER NO. 20 G TERUMO maka membeli INFUS SET DEWASA TERUMO
 Jika membeli INFUS SET DEWASA TERUMO maka membeli THREE WAY STOP COK JMS BEREKOR
 Jika membeli THREE WAY STOP COK JMS BEREKOR maka membeli INFUS SET DEWASA TERUMO
 Jika membeli ID BAND ADULT PINK W/CARD maka membeli INFUS SET DEWASA TERUMO
 Jika membeli ID BAND ADULT BLUE W/CARD maka membeli INFUS SET DEWASA TERUMO
 Jika membeli ID BAND ADULT BLUE W/CARD maka membeli IV CATHETER NO. 20 G TERUMO
 Jika membeli IV CATHETER NO. 20 G TERUMO maka membeli INFUS RL 500 ML SANBE
 Jika membeli IV CATHETER NO. 20 G TERUMO maka membeli THREE WAY STOP COK JMS BEREKOR
 Jika membeli ID BAND ADULT BLUE W/CARD maka membeli IV CATHETER NO. 20 G TERUMO
 Jika membeli IV CATHETER NO. 20 G TERUMO maka membeli INFUS RL 500 ML SANBE
 Jika membeli SPUIT 1 CC TERUMO maka membeli SPUIT 5 CC TERUMO
 Jika membeli SPUIT 5 CC TERUMO maka membeli SPUIT 3 CC TERUMO
 Jika membeli ID BAND ADULT BLUE W/CARD maka membeli SPUIT 3 CC TERUMO
 Jika membeli OPIGRAN U 1 MG maka membeli SPUIT 3 CC TERUMO

Gambar 4. 22 Antarmuka Form Utama User Pola Penjualan Obat

Gambar 4.22 merupakan halaman utama user yang menampilkan pola penjualan. Data yang ditampilkan adalah data yang sebelumnya telah dianalisis oleh Apoteker dari Apotek Anggita.

SISTEM INFORMASI ANALISIS POLA PENJUALAN DAN REKOMENDASI PEMBELIAN OBAT
APOTEK ANGGITA II
 Jl. Kepundung, Kel. Loloan Barat, Kec.Negara, Jembrana, Bali Telp. (0361)42247

Data Obat Pola Penjualan Rekomendasi Pembelian

Cari Log Out

Nama Obat	Rekomendasi	Jumlah Pembelian	ROP
WING NEEDLE 25G TERUMO	12	10	1
IV CATHETER NO. 22 TERUMO	25	14	2
ALKOHOL SWAB	450	60	45
INFUS SET DEWASA TERUMO	27	15	2
IV CATHETER NO. 20 G TERUMO	8	8	0
SPUIT 1 CC TERUMO	17	12	1
SPUIT 3 CC TERUMO	195	39	19
SPUIT 5 CC TERUMO	189	39	18
SPUIT 3 CC ONE MED	88	27	8
SPUIT 5 CC ONE MED	77	25	7
THREE WAY STOP COK JMS BEREKOR	31	16	3
SARUNG TANGAN NON STERIL SENSI GLOVES S	655	72	65
INFUS RL WIDA	41	18	4
ID BAND ADULT PINK W/CARD	19	12	1
ID BAND ADULT BLUE W/CARD	8	8	0
NACL 500 ML WIDATRA	31	16	3
BIOCEF U	29	15	2
OPIGRAN U 1 MG	16	11	1
OPIGRAN U 3 MG	7	7	0
OTTOZOL U	15	11	1
ULCERANIN U	22	13	2
ROCEB U	12	10	1

Gambar 4. 23 Antarmuka Form Utama User Menu Rekomendasi Pembelian

Gambar 4.23 merupakan halaman utama *user* yang hanya menampilkan rekomendasi pembelian obat berdasarkan pola penjualan obat yang dihasilkan. Data yang ditampilkan adalah data yang sebelumnya telah dianalisis oleh Apoteker dari Apotek Anggita.

4.4 Implementasi Program

Pada sub bab ini akan dibahas mengenai implementasi perancangan yang sebelumnya telah dibuat kedalam bahasa pemrograman. Pembahasan kode program dibagi menjadi tiga bagian yaitu penentuan pola penjualan dengan algoritma mApriori, penentuan rekomendasi stok dengan metode *double exponential smothing* dan jumlah pembelian obat terhadap *supplier* dengan metode *Economic Order Quantity (EOQ)*.

4.4.1 Penentuan Pola Penjualan dengan Algoritma mApriori

Tahap pertama yang dilakukan dalam penentuan pola penjualan adalah menentukan data yang dianalisis. Data yang dianalisis hanya data tiga bulan terakhir dari waktu proses analisis dilakukan.

Tabel 4. 1 Kode Penentuan Data Tiga Bulan Terakhir

Baris	Kode
1	<code>Calendar cal = Calendar.getInstance();</code>
2	<code>cal.setTime(new java.util.Date());</code>
3	<code>String[] bulan = new String[jumlah_bulan];</code>
4	<code>int bulan_max = cal.get(Calendar.MONTH) + 1;</code>
5	<code>int tahun_max = cal.get(Calendar.YEAR);</code>
6	<code>for (int i = 0;i<jumlah_bulan;i++)</code>
7	<code>{</code>
8	<code> if (bulan_max - (i + 1)<1)</code>
9	<code> {</code>
10	<code> int bantu = 12+(bulan_max-(i + 1));</code>
11	<code> if (bantu < 10)</code>
12	<code> {</code>
13	<code> bulan[i]= tahun_max-1 + "-0" + bantu;</code>
14	<code> }</code>
15	<code> else</code>
16	<code> {</code>
17	<code> bulan[i] = tahun_max - 1 + "-" + bantu;</code>
18	<code> }</code>
19	<code> }</code>

20	else
21	{
22	int bantu = bulan_max - (i + 1);
23	if (bantu < 10)
24	{
25	bulan[i]= tahun_max + "-0" + bantu;
26	}
27	else
28	{
29	bulan[i] = tahun_max + "-" + bantu;
30	}
31	}
32	}

Setelah ditentukan data yang terpilih, tahap selanjutnya adalah melakukan perhitungan jumlah transaksi pembelian obat untuk masing – masing item obat dengan memberi nilai 0 atau 1. Nilai 1 diberikan jika item obat ditemukan pada transaksi tersebut dan nilai 0 jika item obat tidak ditemukan pada data transaksi tersebut. Setelah itu dihitung jumlah nilai 1 untuk masing – masing item obat dan nilainya disimpan dalam suatu variabel array `jumlah_obat[i]`.

Tabel 4. 2 Kode Penentuan Pemberian Nilai 0 atau 1

Baris	Kode
1	for (int i =0;i<count_obat;i++)
2	{
3	for (int j=0;j<count_transaksi;j++)
4	{
5	rsDataTransaksi = stm.executeQuery("SELECT COUNT(*) AS total FROM dt_detail_transaksi_penjualan WHERE id_transaksi = '"+id_transaksi[j]+' AND kode_obat ='"+id_obat[i]+'");
6	rsDataTransaksi.next();
7	if (rsDataTransaksi.getInt("total")==0)
8	{
9	data[i][j] = 0;
10	}
11	else

12	{
13	
14	data[i][j] =1;
15	jumlah_obat[i]++;
16	}
17	}
18	
19	}

Tahap selanjutnya adalah menentukan nilai satu *itemset*. Penentuan nilai satu *itemset* ditentukan dengan membandingkan nilai `jumlah_obat[i]` dalam tahap sebelumnya dengan nilai minimum *support*. Jika nilai `jumlah_obat[i]` lebih besar atau sama dengan `min_support` maka nilai tersebut disimpan dalam variabel *array* `index_obat[i]` dan frekuensi dari masing – masing obat tersebut disimpan dalam suatu variabel *array* `jumlah_obat[i]`. Hal selanjutnya yang dilakukan adalah menentukan transaksi yang terpilih berdasarkan data obat yang memiliki nilai lebih besar dari minimum *support*. Pada algoritma mApriori dilakukan pengecekan terhadap data transaksi yang berisi data obat yang bernilai 1 dan data transaksi yang terpilih disimpan dalam suatu variabel `index_transaksi_terpilih`, sedangkan pada algoritma apriori tidak perlu memilih data transaksi. Nilai `index_obat[i]`, `jumlah_obat[i]` dan `index_transaksi_terpilih` dimasukkan dalam variabel `Node_Kombinasi` dan `Node_Kombinasi` dimasukkan dalam `LinkedList_Kombinasi`

Tabel 4. 3 Kode Penentuan Satu *Itemset*

Baris	Kode
1	for (int i = 0;i<count_obat; i++)
2	{
3	if (jumlah_obat[i] >= min_support)
4	{
5	Node_Kombinasi = new NodeKombinasi();
6	Node_Kombinasi.index_obat = index_obat[i];
7	Node_Kombinasi.jumlah_obat = jumlah_obat[i];
8	

9	Node_Kombinasi.index_kombinasi_obat =
	LinkedList_Kombinasi.jumlah_kombinasi;
10	Node_Kombinasi.index_kombinasi_obat_sebelumnya =
11	Node_Kombinasi.index_kombinasi_obat;
	Node_Kombinasi.index_obat_minimum =
	Node_Kombinasi.index_obat;
12	
13	if (apriori == false)
14	{
15	Node_Kombinasi.jumlah_transaksi_terpilih =
16	Node_Kombinasi.jumlah_obat;
	Node_Kombinasi.index_transaksi_terpilih = new
17	int[Node_Kombinasi.jumlah_transaksi_terpilih;
18	
19	int bantu_index_transaksi_terpilih = 0;
	for (int j = 0; j < count_transaksi; j++)
20	{
	if (data[Node_Kombinasi.index_obat][j]== 1)
21	{
22	Node_Kombinasi.index_transaksi_terpilih
	[bantu_index_transaksi_terpilih++] = j;
23	}
24	}
25	}
26	else
27	{
28	Node_Kombinasi.jumlah_transaksi_terpilih= 0;
29	Node_Kombinasi.index_transaksi_terpilih =
30	null;
31	}
	LinkedList_Kombinasi.add(Node_Kombinasi);
32	}
33	}

Setelah itu tahap selanjutnya adalah menentukan daftar obat yang digunakan untuk pembentukan *itemset* selanjutnya. Daftar obat tersebut diimpan dalam suatu

variabel *array* Node_Itemset. Node_Itemset selanjutnya dimasukan kedalam LinkedList_Itemset.

Tabel 4. 4 Kode Penentuan Daftar Obat untuk Pembentukan Kombinasi

Baris	Kode
1	if (LinkedList_Kombinasi.jumlah_kombinasi > 1)
2	{
3	Node_Itemset = new NodeItemset();
4	Node_Itemset.itemset =
5	LinkedList_Itemset.jumlah_itemset+1;
6	Node_Itemset.jumlah_obat =
	LinkedList_Kombinasi.jumlah_kombinasi;
7	Node_Itemset.index_obat = new
	int[Node_Itemset.jumlah_obat];
8	NodeKombinasi Node_Kombinasi_Bantu =
	LinkedList_Kombinasi.head;
9	for (int i = 0; i < Node_Itemset.jumlah_obat; i++)
10	{
11	Node_Itemset.index_obat[i] =
12	Node_Kombinasi_Bantu.index_obat;
	Node_Kombinasi_Bantu =
13	Node_Kombinasi_Bantu.next;
	}
14	Node_Itemset.kombinasi = LinkedList_Kombinasi;
	LinkedList_Itemset.add(Node_Itemset);

Tahap selanjutnya adalah penentuan kombinasi dua *itemset* atau lebih. Hal pertama yang dilakukan dalam penentuan kombinasi adalah menyimpan jumlah kombinasi di *itemset* sebelumnya dan disimpan dalam variabel bantu_jumlah_kombinasi, setelah itu dilakukan pembentukan kombinasi untuk itemset selanjutnya. Berdasarkan kombinasi yang telah dibentuk, dilakukan pemilihan data transaksi terpilih dengan cara memilih *itemset* yang memiliki jumlah yang lebih minimum yang digunakan acuan dalam pemilihan data transaksi

sehingga tidak perlu mencocokkan setiap item pada semua data transaksi. Tahapan berikutnya adalah menghitung jumlah kombinasi berdasarkan data transaksi yang terpilih. Data transaksi terpilih nantinya akan disimpan dalam variabel `index_transaksi_terpilih`, namun akan dilakukan pengecekan terlebih dahulu apakah `bantu_jumlah_obat_iterasi` lebih kecil dari `bantu_jumlah_obat` maka data transaksi yang terpilih hanya data yang bernilai satu sedangkan jika `bantu_jumlah_obat_iterasi` = `bantu_jumlah_obat` maka data transaksi disimpan seluruhnya. Jumlah dari kombinasi yang telah terbentuk selanjutnya dibandingkan dengan minimum *support*, jika `bantu_jumlah_obat` lebih besar dari `min_support` maka data obat disimpan ke `Node_Kombinasi`. Tahap terakhir yang dilakukan adalah pengecekan apakah dapat dibentuk kombinasi itemset selanjutnya atau tidak.

Tabel 4. 5 Kode Pembentukan Kombinasi *Itemset*

Baris	Kode
1	Do
2	{
3	int bantu_jumlah_kombinasi =
	LinkedList_Kombinasi.jumlah_kombinasi;
4	LinkedList_Kombinasi = new LinkListKombinasi();
5	NodeItemset Node_Itemset_Iterasi =
	LinkedList_Itemset.tail;
6	NodeKombinasi Node_Kombinasi_Iterasi =
	Node_Itemset_Iterasi.kombinasi.head;
7	for (int i = 0; i < bantu_jumlah_kombinasi; i++)
8	{
9	int index_obat_minimum =
	Node_Kombinasi_Iterasi.index_obat_minimum;
10	int index_obat_sebelumnya =
	Node_Kombinasi_Iterasi.index_obat;
11	for (int j = 0; j <
12	Node_Itemset_Iterasi.jumlah_obat; j++)
13	{
14	int index_obat_sekarang =
	Node_Itemset_Iterasi.index_obat[j];
15	if (index_obat_sekarang > index_obat_sebelumnya
	&& index_obat_minimum != index_obat_sekarang)
16	{
17	int bantu_jumlah_obat = 0;
18	if (apriori == true)

19	{
20	for (int k = 0; k < count_transaksi; k++)
21	{
22	if (data[index_obat_sekarang] [k] == 1)
23	{
24	NodeItemset Node_Itemset_Bantu_Apriori
	= Node_Itemset_Iterasi;
25	int
	index_kombinasi_obat_sebelumnya_bantu_a
	priori =
	Node_Kombinasi_Iterasi.index_kombinasi_
	obat;
27	
	int bantu_jumlah_obat_apriori = 0;
28	
	while (Node_Itemset_Bantu_Apriori !=
29	null)
30	{
31	NodeKombinasi
	Node_Kombinasi_Bantu_Apriori =
	Node_Itemset_Bantu_Apriori.kombinasi
	.head;
32	
	for (int l = 0; l <
	index_kombinasi_obat_sebelumnya_ban
33	tu_apriori; l++)
34	{
	Node_Kombinasi_Bantu_Apriori =
35	Node_Kombinasi_Bantu_Apriori.next;
	}
36	
	if
	(data[Node_Kombinasi_Bantu_Apriori.
37	index_obat][k] == 1)
	{
38	
39	bantu_jumlah_obat_apriori++;
40	}
41	index_kombinasi_obat_sebelumnya_bantu_apriori =
42	Node_Kombinasi_Bantu_Apriori.index_kombinasi_obat_s
	ebelumnya;
43	Node_Itemset_Bantu_Apriori =
	Node_Itemset_Bantu_Apriori.prev;
44	}
45	if (bantu_jumlah_obat_apriori ==
	LinkedList_Itemset.jumlah_itemset)
46	{
47	bantu_jumlah_obat++;

```

48         }
49     }
50 }
51 }
52 else
53 {
54     boolean baru = false;
55     if (jumlah_obat[index_obat_sekarang] <
56         jumlah_obat[index_obat_minimum])
57     {
58         index_obat_minimum = index_obat_sekarang;
59         baru = true;
60     }
61     else
62     {
63         baru = false;
64     }
65     if (baru == true)
66     {
67         int jumlah_transaksi_terpilih =
68             jumlah_obat[index_obat_minimum];
69         int[] index_transaksi_terpilih = new
70             int[jumlah_transaksi_terpilih];
71         int bantu_index_transaksi_terpilih = 0;
72
73         for (int k = 0; k < count_transaksi; k++)
74         {
75             if (data[index_obat_minimum][k] == 1)
76             {
77                 index_transaksi_terpilih[bantu_index_transaksi_
78                     terpilih++] = k;
79             }
80         }
81         NodeItemset Bantu_Node_Itemset =
82             Node_Itemset_Iterasi;
83         int bantu_index_kombinasi_sebelumnya =
84             Node_Kombinasi_Iterasi.index_kombinasi_obat;
85         bantu_jumlah_obat = jumlah_transaksi_terpilih;
86
87         do
88         {
89             NodeKombinasi Bantu_Node_Kombinasi =
90                 Bantu_Node_Itemset.kombinasi.head;
91             for (int k = 0; k <
92                 bantu_index_kombinasi_sebelumnya; k++)
93             {
94                 Bantu_Node_Kombinasi =
95                     Bantu_Node_Kombinasi.next;

```

```

86         }
            int bantu_jumlah_obat_iterasi = 0;
87
88         for (int k = 0; k < jumlah_transaksi_terpilih; k++)
            {
89
90             if(data[Bantu_Node_Kombinasi.index_obat][index_transaksi_terpilih[k]] == 1)
91             {
92                 bantu_jumlah_obat_iterasi++;
93             }
94         }
95         if (bantu_jumlah_obat_iterasi <= bantu_jumlah_obat)
96         {
97             int jumlah_transaksi_terpilih_bantu =
98                 bantu_jumlah_obat;
99             int[] index_transaksi_terpilih_bantu =
100                 index_transaksi_terpilih;
101             bantu_jumlah_obat = bantu_jumlah_obat_iterasi;
102             jumlah_transaksi_terpilih = bantu_jumlah_obat;
103             index_transaksi_terpilih = new
104                 int[jumlah_transaksi_terpilih];
105             index_obat_minimum =
106                 Bantu_Node_Kombinasi.index_obat;
107             bantu_index_transaksi_terpilih = 0;
108             for (int k = 0; k <
109                 jumlah_transaksi_terpilih_bantu; k++)
110             {
111
112                 if(data[Bantu_Node_Kombinasi.index_obat][index_transaksi_terpilih_bantu[k]] == 1)
113                 {
114                     index_transaksi_terpilih[bantu_index_transaksi_terpilih++] =
115                         index_transaksi_terpilih_bantu[k];
116                 }
117             }
118         }
119
120         bantu_index_kombinasi_sebelumnya =
121             Bantu_Node_Kombinasi.index_kombinasi_obat_sebelumnya;
122         Bantu_Node_Itemset = Bantu_Node_Itemset.prev;
123     }
124     while (Bantu_Node_Itemset != null);
125 }
126 else
127 {
128     int jumlah_transaksi_terpilih =
129         Node_Kombinasi_Iterasi.jumlah_transaksi_terpilih;

```

121	int[] index_transaksi_terpilih = new int[jumlah_transaksi_terpilih]; int bantu_index_transaksi_terpilih = 0;
122	
123	for (int k = 0; k < jumlah_transaksi_terpilih; k++)
124	{
125	if(data[Node_Kombinasi_Iterasi.index_obat][Node_Kombinasi_Iterasi.index_transaksi_terpilih[k]] == 1)
126	{
127	index_transaksi_terpilih[bantu_index_transaksi_terpilih++] = Node_Kombinasi_Iterasi.index_transaksi_terpilih[k]
128	};
129	}
130	for (int k = 0; k < jumlah_transaksi_terpilih; k++)
131	{
132	if(data[index_obat_sekarang][index_transaksi_terpilih[k]] == 1)
133	{
134	bantu_jumlah_obat++;
135	}
136	}
137	}
138	}
139	if (bantu_jumlah_obat >= min_support)
140	{
141	Node_Kombinasi = new NodeKombinasi();
142	Node_Kombinasi.index_obat = index_obat_sekarang;
143	if (apriori == false)
144	{
145	Node_Kombinasi.index_obat_minimum = index_obat_minimum;
146	}
147	else
148	{
149	Node_Kombinasi.index_obat_minimum = -1;
150	}
151	Node_Kombinasi.jumlah_obat = bantu_jumlah_obat;
152	Node_Kombinasi.index_kombinasi_obat = LinkedList_Kombinasi.jumlah_kombinasi;
153	Node_Kombinasi.index_kombinasi_obat_sebelumnya = Node_Kombinasi_Iterasi.index_kombinasi_obat;
154	if (apriori == false)
155	{
	Node_Kombinasi.jumlah_transaksi_terpilih = Node_Kombinasi.jumlah_obat;

```

156     Node_Kombinasi.index_transaksi_terpilih =
157     newint[Node_Kombinasi.jumlah_transaksi_terpilih];
158     int bantu_index_transaksi_terpilih = 0;
159
160     for (int k = 0; k <
161         Node_Kombinasi_Iterasi.jumlah_transaksi_terpilih;
162         k++)
163     {
164         if (data[Node_Kombinasi.index_obat][Node_Kombinasi
165             _Iterasi.index_transaksi_terpilih[k]] == 1)
166         {
167             Node_Kombinasi.index_transaksi_terpilih[bantu_i
168                 ndex_transaksi_terpilih++] =
169             Node_Kombinasi_Iterasi.index_transaksi_terpilih
170             [k];
171         }
172     }
173 }
174 else
175 {
176     Node_Kombinasi.jumlah_transaksi_terpilih = 0;
177     Node_Kombinasi.index_transaksi_terpilih = null;
178 }
179 LinkedList_Kombinasi.add(Node_Kombinasi);
180 }
181 }
182 Node_Kombinasi_Iterasi = Node_Kombinasi_Iterasi.next;
183 }
184 if (LinkedList_Kombinasi.jumlah_kombinasi > 0)
185 {
186     Node_Itemset = new NodeItemset();
187     Node_Itemset.itemset =
188     LinkedList_Itemset.jumlah_itemset+1;
189     int bantu_jumlah_obat = 0;
190     NodeKombinasi Bantu_Node_Kombinasi_1 =
191     LinkedList_Kombinasi.head;
192
193     for (int i = 0; i <
194         LinkedList_Kombinasi.jumlah_kombinasi; i++)
195     {
196         NodeKombinasi Bantu_Node_Kombinasi_2 =
197         LinkedList_Kombinasi.head;
198         int j = 0;
199         for (j = 0; j < i; j++)
200         {
201             if (Bantu_Node_Kombinasi_1.index_obat ==
202                 Bantu_Node_Kombinasi_2.index_obat)
203             {

```

```

190         break;
191     }
192     Bantu_Node_Kombinasi_2 =
193     Bantu_Node_Kombinasi_2.next;
194     }
195     if (j == i)
196     {
197         bantu_jumlah_obat++;
198     }
199     Bantu_Node_Kombinasi_1 = Bantu_Node_Kombinasi_1.next;
200     }
201     Node_Itemset.jumlah_obat = bantu_jumlah_obat;
202     Node_Itemset.index_obat = new
203     int[Node_Itemset.jumlah_obat];
204     bantu_jumlah_obat = 0;
205     Bantu_Node_Kombinasi_1 = LinkedList_Kombinasi.head;
206     for (int i = 0; i <
207     LinkedList_Kombinasi.jumlah_kombinasi; i++)
208     {
209         NodeKombinasi Bantu_Node_Kombinasi_2 =
210         LinkedList_Kombinasi.head;
211         int j = 0;
212         for (j = 0; j < i; j++)
213         {
214             if (Bantu_Node_Kombinasi_1.index_obat ==
215             Bantu_Node_Kombinasi_2.index_obat)
216             {
217                 break;
218             }
219             Bantu_Node_Kombinasi_2 = Bantu_Node_Kombinasi_2.next;
220         }
221         if (j == i)
222         {
223             Node_Itemset.index_obat[bantu_jumlah_obat++] =
224             Bantu_Node_Kombinasi_1.index_obat;
225         }
226         Bantu_Node_Kombinasi_1 = Bantu_Node_Kombinasi_1.next;
227     }
228     Node_Itemset.kombinasi = LinkedList_Kombinasi;
229     LinkedList_Itemset.add(Node_Itemset);
230     }
231     if (LinkedList_Kombinasi.jumlah_kombinasi > 1 &&
232     LinkedList_Itemset.tail.jumlah_obat > 1)
233     {
234         iterasi = true;
235     }
236     else
237     {
238         iterasi = false;

```


222	}
223	}
224	while (iterasi == true);
225	

Setelah pola terbentuk, tahap selanjutnya adalah penentuan nilai *support* dan nilai *confidence*. Nilai *confidence* yang dihitung dalam algoritma mApriori adalah untuk pola yang tidak terbalik dan untuk pola yang terbalik. Perbedaan dari perhitungan nilai *confidence* terletak pada baris ke-1, baris ke-15, baris ke-33 dan baris ke-47. Nilai *confidence* dan *support* yang telah diperoleh disimpan dalam Linked List *Itemset*.

Tabel 4. 6 Kode Penentuan Nilai *Support* dan Nilai *Confidence*

Baris	Kode
1	double confidence = (double) (Node_Kombinasi_Akhir.jumlah_obat) / (double) (Node_Kombinasi_Akhir_Sebelumnya.jumlah_obat);
2	if (confidence >= min_confidence)
3	{
4	double support = (double) (Node_Kombinasi_Akhir.jumlah_obat) / (double) (count_transaksi);
5	Node_Pola = new Node_Pola();
6	Node_Pola.support = support;
7	Node_Pola.confidence = confidence;
8	int bantu_nomor = 2;
9	int bantu_index_kombinasi_obat_sebelumnya = Node_Kombinasi_Akhir_Sebelumnya.index_kombinasi_obat_se belumnya;
10	NodeItemset Node_Itemset_Akhir_Bantu = Node_Itemset_Akhir.prev.prev;
11	Node_Pola.jumlah_obat = 2+(Node_Itemset_Akhir_Bantu !=null ?Node_Itemset_Akhir_Bantu.itemset:0);
12	Node_Pola.index_obat = new int [Node_Pola.jumlah_obat];
13	Node_Pola.nomor = new int [Node_Pola.jumlah_obat];
14	Node_Pola.index_obat[0] =Node_Kombinasi_Akhir.index_obat;
15	Node_Pola.nomor[0] =0;
16	Node_Pola.index_obat[1] = Node_Kombinasi_Akhir_Sebelumnya.index_obat;
17	Node_Pola.nomor[1] = 1;
18	while (Node_Itemset_Akhir_Bantu != null)

```

19     {
20         NodeKombinasi Node_Kombinasi_Akhir_Bantu =
            Node_Itemset_Akhir_Bantu.kombinasi.head;

21         for (int i = 0; i <
            bantu_index_kombinasi_obat_sebelumnya; i++)
22         {
23             Node_Kombinasi_Akhir_Bantu =
                Node_Kombinasi_Akhir_Bantu.next;
24         }
25         bantu_index_kombinasi_obat_sebelumnya =
            Node_Kombinasi_Akhir_Bantu.index_kombinasi_obat_sebe
            lumnya;
26         Node_Pola.index_obat[bantu_nomor] =
            Node_Kombinasi_Akhir_Bantu.index_obat;
27         Node_Pola.nomor[bantu_nomor] = 1;
28         bantu_nomor++;
29         Node_Itemset_Akhir_Bantu =
            Node_Itemset_Akhir_Bantu.prev;
30     }
31     LinkedList_Pola.add(Node_Pola);
32 }

33     confidence = (double) (Node_Kombinasi_Akhir.jumlah_obat)
        / (double) (jumlah_obat[Node_Kombinasi_Akhir.index_obat])
        ;
34     if (confidence >= min_confidence)
35     {
36         double support =
            (double) (Node_Kombinasi_Akhir.jumlah_obat) /
            (double) (count_transaksi);
37         Node_Pola = new Node_Pola();
38         Node_Pola.support = support;
39         Node_Pola.confidence = confidence;
40         int bantu_nomor = 2;
41         int bantu_index_kombinasi_obat_sebelumnya =
            Node_Kombinasi_Akhir_Sebelumnya.index_kombinasi_obat_seb
            elumnya;

42         NodeItemset Node_Itemset_Akhir_Bantu =
            Node_Itemset_Akhir.prev.prev;
43         Node_Pola.jumlah_obat = 2+(Node_Itemset_Akhir_Bantu
            !=null ?Node_Itemset_Akhir_Bantu.itemset:0);
44         Node_Pola.index_obat = new int [Node_Pola.jumlah_obat];
45         Node_Pola.nomor = new int [Node_Pola.jumlah_obat];

46         Node_Pola.index_obat[0]=Node_Kombinasi_Akhir.index_obat
            ;
47         Node_Pola.nomor[0] = 2;

```

48	Node_Pola.index_obat[1] =
	Node_Kombinasi_Akhir_Sebelumnya.index_obat;
49	Node_Pola.nomor[1] = 1;
50	while (Node_Itemset_Akhir_Bantu != null)
51	{
52	NodeKombinasi Node_Kombinasi_Akhir_Bantu =
	Node_Itemset_Akhir_Bantu.kombinasi.head;
53	for (int i = 0; i <
	bantu_index_kombinasi_obat_sebelumnya; i++)
55	{
56	Node_Kombinasi_Akhir_Bantu =
	Node_Kombinasi_Akhir_Bantu.next;
57	}
58	bantu_index_kombinasi_obat_sebelumnya =
	Node_Kombinasi_Akhir_Bantu.index_kombinasi_obat_seb
	elumnya;
59	Node_Pola.index_obat[bantu_nomor] =
	Node_Kombinasi_Akhir_Bantu.index_obat;
60	Node_Pola.nomor[bantu_nomor] = 1;
61	bantu_nomor++;
62	Node_Itemset_Akhir_Bantu =
	Node_Itemset_Akhir_Bantu.prev;
63	}
64	
65	

4.4.2 Perhitungan Rekomendasi Stok dengan Metode Double Exponential

Pada bagian ini akan membahas mengenai tahapan penentuan rekomendasi stok obat yang harus disediakan Apotek Anggita dalam satu bulan. Hal pertama yang dilakukan adalah menentukan jumlah obat yang terdapat pada tabel `dt_detail_pola` yang memiliki `kode_obat` yang berbeda dan `kode_obat` yang sama akan dihitung satu. Setelah itu dipilih `kode_obat` pada tabel `dt_detail_pola` yang memiliki kode obat yang berbeda dan yang sama dihitung satu. Data `kode_obat` yang telah dipilih kemudian disimpan dalam suatu *array* yang diberi nama `nama_obat[i]`.

Tabel 4. 7 Kode Pemilihan Data Obat dalam Pola Penjualan Obat

Baris	Kode
1	<code>rsDataTransaksi = stm.executeQuery("SELECT COUNT(DISTINCT</code>
2	<code>kode_obat) AS total FROM dt_detail_pola");</code>
3	<code>rsDataTransaksi.next();</code>
	<code>jumlah_obat = rsDataTransaksi.getInt("total");</code>

4	nama_obat = new String[jumlah_obat];
5	rsDataTransaksi = stm.executeQuery("SELECT DISTINCT kode_obat FROM dt_detail_pola");
6	for (int i = 0;i<jumlah_obat;i++)
7	{
8	rsDataTransaksi.next();
9	nama_obat[i] = rsDataTransaksi.getString("kode_obat");
10	}

Tahap kedua adalah penentuan rekomendasi stok dengan metode *double exponential smothing*. Pertama menentukan jumlah pembelian obat untuk masing – masing obat setiap bulan. Jumlah pembelian obat yang dihitung hanya data tiga bulan terakhir dari waktu analisis dilakukan. Setelah itu menentukan nilai dari S'1 yang dalam hal ini diberi nama variabel st1, nilai dari st1 sama dengan jumlah obat pada bulan ke 0 dan nilai S''1 yang dalam hal ini yang diberi nama variabel st2 memiliki nilai yang sama dengan st1. Tahap selanjutnya adalah menentukan nilai dari S'2 dan S'3 yang dihitung dengan rumus $\alpha * x[j] + (1 - \alpha) * st1[i][j-1]$ sedangkan untuk S''2 dan S''3 dihitung dengan rumus $\alpha * st1[i][j] + (1-\alpha) * st2[i][j-1]$. Nilai dari S'3 dan S''3 selanjutnya digunakan untuk menentukan nilai dari at dan bt. Tahap terakhir yang dilakukan adalah menentukan rekomendasi stok dengan cara mengurangi nilai at dan bt yang telah diperoleh sebelumnya.

Tabel 4. 8 Kode Perhitungan Metode Double Exponential Smothing

Baris	Kode
1	int x[] = new int[jumlah_bulan];
2	st1 = new double[jumlah_obat][jumlah_bulan];
3	st2 = new double[jumlah_obat][jumlah_bulan];
4	at = new double[jumlah_obat][jumlah_bulan];
5	bt = new double[jumlah_obat][jumlah_bulan];
6	sthasil = new double[jumlah_obat];
7	for (int i =0;i<jumlah_obat;i++)
8	{
9	for (int j=0;j<jumlah_bulan;j++)
10	{
11	x[j] = 0;
12	rsDataTransaksi = stm.executeQuery("SELECT jumlah
13	FROM
14	dt_detail_transaksi_penjualan,dt_transaksi_penjualan
15	WHERE dt_transaksi_penjualan.id_transaksi =

```

16      dt_detail_transaksi_penjualan.id_transaksi AND
17      kode_obat ='" +nama_obat[i]+"' AND tanggal_transaksi
18      LIKE '" +bulan[j]+"%";
19      while (rsDataTransaksi.next())
20      {
21          x[j]+=rsDataTransaksi.getInt("jumlah");
22      }
23      if (j==0)
24      {
25          st1[i][j] = x[j];
26          st2[i][j] = st1[i][j];
27      }
28      else
29      {
30          st1[i][j] = alpha * x[j] + (1 - alpha)
31          *st1[i][j-1];
32          st2[i][j] = alpha * st1[i][j] +(1-alpha)
33          *st2[i][j-1];
34      }
35      at[i][j] = 2 * st1[i][j] - st2[i][j];
36      bt[i][j] = (alpha/(1-alpha))*(st1[i][j] -
37      st2[i][j]);
38      }
39      sthasil[i] = at[i][jumlah_bulan - 1]
40      +bt[i][jumlah_bulan - 1];
41      }

```

4.4.3 Perhitungan Pembelian Obat terhadap Supplier dengan Metode EOQ

Pada bagian ini akan membahas mengenai penentuan pembelian obat terhadap *supplier* dengan menggunakan metode *Economic Order Quantity (EOQ)*. Pada perhitungan metode EOQ, hal pertama yang dilakukan adalah memilih biaya pesan dari tabel *dt_obat* dan kemudian disimpan dalam variabel *array* yang diberi nama *biaya_pesan[i]*. Nilai *st* hasil yang sebelumnya telah diperoleh dari perhitungan metode *double exponential smothing* disimpan dalam variabel *array* yang diberi nama *rekomendasi_stok[i]*. Pembelian obat terhadap *supplier* yang dalam hal ini diberi nama variabel *a* dihitung dengan rumus $\text{Math.round}(\text{Math.sqrt}((2 * \text{rekomendasi_stok}[i] * \text{biaya_pesan}[i]) / (0.25 * \text{biaya_pesan}[i])))$. Perhitungan selanjutnya adalah perhitungan nilai Reorder Point (ROP), ROP merupakan stok minimum yang tersedia di Apotek Anggita dan harus melakukan pemesanan kembali kepada *supplier*. Nilai ROP

dihitung dengan menggunakan rumus $rop = (int)(rsDataTransaksi.getDouble("lead_time") * (rekomendasi_stok[i]/30))$

Tabel 4. 9 Kode Perhitungan Metode EOQ

Baris	Kode
1	double[] biaya_pesan = new double[jumlah_obat];
2	double[] permintaan = new double [jumlah_obat];
3	double[] biaya_simpan = new double [jumlah_obat];
4	double[] rekomendasi_stok = new
5	double[jumlah_obat];
6	stok = new int[jumlah_obat];
7	for (int i=0;i<jumlah_obat;i++)
8	{
	rsDataTransaksi = stm.executeQuery("SELECT
	biaya_pesan FROM dt_obat WHERE kode_obat =
9	'"+nama_obat[i]+"");
10	rsDataTransaksi.next();
11	biaya_pesan[i] =
12	rsDataTransaksi.getDouble("biaya_pesan");
	rsDataTransaksi = stm.executeQuery("SELECT
	COUNT(*)AS total FROM
	dt_detail_transaksi_penjualan WHERE kode_obat
13	= '"+nama_obat[i]+"");
14	rsDataTransaksi.next();
	int jumlah_transaksi =
15	rsDataTransaksi.getInt("total");
16	permintaan[i] = 0;
	rsDataTransaksi = stm.executeQuery("SELECT
	jumlah FROM dt_detail_transaksi_penjualan
17	WHERE kode_obat = '"+nama_obat[i]+"");
18	for (int j=0;j<jumlah_transaksi;j++)
19	{
20	rsDataTransaksi.next();
	permintaan[i]+=rsDataTransaksi.getDouble("
21	jumlah");
	}
22	biaya_simpan[i] = biaya_pesan[i];
23	rekomendasi_stok[i] = sthasil[i];
24	rsDataTransaksi = stm.executeQuery("SELECT
	lead_time FROM dt_obat WHERE kode_obat =
	'"+nama_obat[i]+"");
26	rsDataTransaksi.next();

27	double a;
28	a =
29	Math.round(Math.sqrt((2*rekomendasi_stok[i]
	*biaya_pesan[i]) / (0.25*biaya_pesan[i])));
	stok[i] = (int)a;
	int rop =
	(int) (rsDataTransaksi.getDouble("lead_time") *
	(rekomendasi_stok[i]/30));
	double TCC = (rekomendasi_stok[i] /
	2)*biaya_pesan[i];

4.5 Hasil Penentuan Rekomendasi Pembelian Obat

Tabel 4.10 merupakan hasil rekomendasi pembelian obat terhadap *supplier*. Obat – obat yang ditentukan nilai rekomendasinya adalah obat – obat yang terdapat pada pola penjualan obat. Pada bagian ini terdapat tiga nilai yang ditentukan yaitu rekomendasi stok, jumlah pembelian dan nilai *reorder point*. Hasil rekomendasi pembelian obat ditentukan berdasarkan jumlah pembelian obat di masing – masing bulan berdasarkan data tiga bulan terakhir dari proses analisis dilakukan.

Rekomendasi stok adalah jumlah obat yang harus disediakan pihak apoeck dalam satu bulan. Nilai ini ditentukan dengan metode *Double Exponential Smoothing*. Jumlah pembelian adalah jumlah obat yang dibeli pihak apotek setiap kali melakukan pemesanan obat terhadap *supplier*. *Reorder point* adalah jumlah obat minimum yang menjadi patokan untuk melakukan pemesanan obat kembali terhadap *supplier*. Nilai jumlah pembelian dan *reorder point* ditentukan dengan menggunakan metode EOQ.

Tabel 4. 10 Hasil Rekomendasi Pembelian Obat Bulan April Tahun 2015

No	Nama Obat	Rekomendasi Stok	Jumlah Pembelian	Reorder Point
1	SARUNG TANGAN NON STERIL SENSI GLOVES S	249	45	24
2	SANMOL FORTE	225	42	22
3	ALKOHOL SWAB	150	35	15
4	SPUIT 3 CC ONE MED	118	31	11
5	SPUIT 3 CC TERUMO	105	29	10
6	SPUIT 5 CC ONE MED	100	28	10

7	SPUIT 5 CC TERUMO	93	27	9
8	INFUS NS SANBE	68	23	6
9	INFUS RL 500 ML SANBE	62	22	6
10	TAXEGRAM IJ	59	22	5
11	THREE WAY STOP COK JMS BEREKOR	55	21	5
12	SANMOL INFUS	49	20	4
13	INFUS FUTROLIT SANBE	49	20	4
14	IV CATHETER NO. 20 G TERUMO	45	19	4
15	COMBIVENT NEBUL	45	19	4
16	INFUS RL WIDA	41	18	4
17	INFUS D5 SANBE	39	18	3
18	IMUNOS TAB	38	17	3
19	ACRAN IJ	38	17	3
20	RANITIDIN IJ 25 MG	37	17	3
21	BIOCEF IJ	34	16	3
22	AQUA PI 25 ML	34	16	3
23	RYCEF IJ	33	16	3
24	NACL 500 ML WIDATRA	33	16	3
25	ID BAND ADULT BLUE W/CARD	31	16	3
26	INFUS SET DEWASA TERUMO	29	15	2
27	INVOMIT IJEKSI 4 MG	27	15	2
28	IV CATHETER NO. 22 TERUMO	27	15	2
29	WING NEEDLE 25G TERUMO	26	14	2
30	GASTRIDIN IJ	21	13	2
31	PANTOPUMP IJ	20	13	2
32	CEFOTAXIME IJ I GR	20	13	2
33	D5 500ML WIDA	20	13	2
34	LAPIXIME IJ	19	12	1
35	KETOROLAC IJ 30 MG	19	12	1
36	ID BAND ADULT PINK W/CARD	19	12	1
37	OTTOZOL IJ	17	12	1
38	OPIGRAN IJ 1 MG	17	12	1

39	ULCERANIN IJ	15	11	1
40	TOPAZOL IJ	15	11	1
41	SPUIT 1 CC TERUMO	14	11	1
42	PUMPISEL IJ	13	10	1
43	SPUIT 10 CC ONE MED	12	10	1
44	TROVENSIS 4 MG IJ	12	10	1
45	AMINOPHYLLIN IJ 10 ML	11	9	1
46	ANTRAIN IJ	11	9	1
47	LAMESON IJ	8	8	0
48	ROCER IJ	8	8	0
49	UNDER PAD 60 X 90 CM OM	7	7	0
50	OPIGRAN IJ 3 MG	7	7	0
51	NEUROSANBE IJ	7	7	0
52	SANTAGESIK IJ	6	6	0

4.6 Hasil Penentuan Pola Penjualan Obat

Tabel 4.11 merupakan beberapa hasil penentuan pola penjualan obat pada bulan April tahun 2015 berdasarkan hasil nilai *sup x conf*. Nilai *support* menunjukkan seberapa besar tingkat dominasi suatu *itemset* dari keseluruhan transaksi. Nilai *confidence* adalah suatu ukuran yang menunjukkan hubungan antar dua *item* secara *conditional*. Nilai *sup x conf* merupakan hasil final dari *assosiation rule* yang dihasilkan. Hasil – hasil perhitungan ini diperoleh dari perhitungan menggunakan algoritma m-Apriori

Tabel 4. 11 Hasil Penentuan Pola Penjualan Obat Bulan April Tahun 2015

No	Nama	Support	Confidence	Supp x conf
1	Jika membeli SARUNG TANGAN NON STERIL SENSI GLOVES S maka membeli ALKOHOL SWAB	12,73%	85,39%	10,87%
2	Jika membeli ALKOHOL SWAB maka membeli SARUNG TANGAN NON STERIL SENSI GLOVES S	12,73%	72,76%	9,26%

3	Jika membeli INFUS SET DEWASA TERUMO maka membeli THREE WAY STOP COK JMS BEREKOR	6,19%	89,22%	5,53%
4	Jika membeli ALKOHOL SWAB, SPUIT 5 CC TERUMO maka membeli SARUNG TANGAN NON STERIL SENSI GLOVES S	6,54%	80,67%	5,27%
5	Jika membeli THREE WAY STOP COK JMS BEREKOR maka membeli INFUS SET DEWASA TERUMO	6,19%	83,49%	5,17%
6	Jika membeli ALKOHOL SWAB, SPUIT 3 CC TERUMO maka membeli SARUNG TANGAN NON STERIL SENSI GLOVES S	4,97%	77,66%	3,86%
7	Jika membeli INFUS SET DEWASA TERUMO, IV CATHETER NO. 20 G TERUMO maka membeli THREE WAY STOP COK JMS BEREKOR	3,40%	98,04%	3,34%
8	Jika membeli IV CATHETER NO. 20 G TERUMO maka membeli INFUS SET DEWASA TERUMO	3,47%	89,47%	3,11%
9	Jika membeli IV CATHETER NO. 20 G TERUMO maka membeli THREE WAY STOP COK JMS BEREKOR	3,47%	89,47%	3,11%
10	Jika membeli SPUIT 3 CC ONE MED maka membeli SPUIT 5 CC ONE MED	5,65%	54,97%	3,11%
11	Jika membeli SPUIT 5 CC ONE MED maka membeli ALKOHOL SWAB	5,65%	50,61%	2,86%
12	Jika membeli SPUIT 5 CC ONE MED maka membeli SPUIT 3 CC ONE MED	5,65%	50,61%	2,86%
13	Jika membeli ALKOHOL SWAB, SPUIT 3 CC ONE MED maka membeli SPUIT 5 CC ONE MED	3,54%	80,00%	2,83%
14	Jika membeli ID BAND ADULT PINK W/CARD maka membeli THREE WAY STOP COK JMS BEREKOR	3,34%	84,48%	2,82%
15	Jika membeli ALKOHOL SWAB, SPUIT 3 CC TERUMO maka membeli SPUIT 5 CC TERUMO	4,22%	65,96%	2,78%

16	Jika membeli ALKOHOL SWAB, SPUIT 3 CC TERUMO, SPUIT 5 CC TERUMO maka membeli SARUNG TANGAN NON STERIL SENSI GLOVES S	3,27%	77,42%	2,53%
17	Jika membeli INFUS SET DEWASA TERUMO, SPUIT 3 CC TERUMO maka membeli THREE WAY STOP COK JMS BEREKOR	2,52%	92,50%	2,33%
18	Jika membeli ALKOHOL SWAB, SPUIT 5 CC ONE MED maka membeli SARUNG TANGAN NON STERIL SENSI GLOVES S	3,61%	63,86%	2,30%
19	Jika membeli THREE WAY STOP COK JMS BEREKOR maka membeli INFUS RL 500 ML SANBE	4,08%	55,05%	2,25%
20	Jika membeli INFUS SET DEWASA TERUMO, SPUIT 5 CC TERUMO maka membeli THREE WAY STOP COK JMS BEREKOR	2,25%	97,06%	2,18%

4.7 Pengujian Sistem

Pada tahapan ini, program secara keseluruhan akan diuji sebagai sistem yang lengkap untuk memastikan bahwa persyaratan perangkat lunak telah dipenuhi dan sesuai dengan kebutuhan serta keinginan pengguna. Pada penelitian ini teknik pengujian yang digunakan yaitu *Lift Ratio Testing*, pengujian peramalan stok, *Black Box Testing*, *White Box Testing* dan Uji Akurasi Hasil Rekomendasi.

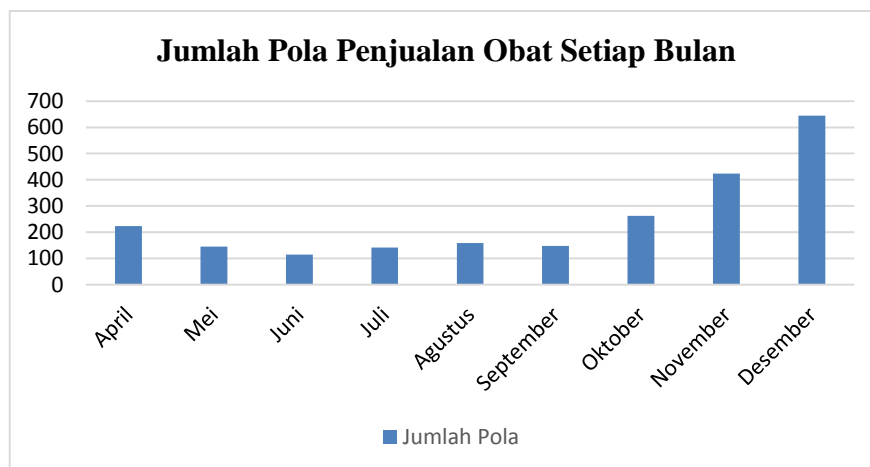
4.7.1 Pengujian *Lift Ratio*

Pada penelitian tentang penentuan pola penjualan obat dan rekomendasi pembelian obat terhadap *supplier*, digunakan data pada tahun 2015. Jumlah data transaksi penjualan obat yang dianalisis dalam penelitian ini adalah sebagai berikut.

Tabel 4. 12 Jumlah Data Transaksi Penjualan Obat

No	Nama Bulan Analisis	Nama Bulan Prediksi	Jumlah Transaksi
1	Januari, Februari, Maret	April	1469
2	Februari, Maret, April	Mei	1336
3	Maret, April, Mei	Juni	1210
4	April, Mei, Juni	Juli	1240
5	Mei, Juni, Juli	Agustus	1552
6	Juni, Juli, Agustus	September	1960
7	Juli, Agustus, September	Oktober	2046
8	Agustus, September, Oktober	November	2071
9	September, Oktober, November	Desember	2022

Gambar 4.24 merupakan grafik yang menggambarkan jumlah pola penjualan obat yang dihasilkan berdasarkan data transaksi penjualan obat dalam tiga bulan terakhir dari proses analisis dilakukan. Berdasarkan grafik tersebut jumlah pola penjualan paling banyak dihasilkan oleh sistem terdapat pada bulan Desember, hal ini disebabkan oleh banyak jumlah item yang dibeli bersamaan dengan item lain yang berada diatas minimum *support* dan diatas minimum *confidence*. Nilai minimum *support* dalam penelitian ini adalah 10 dan minimum *confidence* adalah 0,5.

**Gambar 4. 24 Grafik Jumlah Pola Penjualan Obat Setiap Bulan**

Tabel 4.13 merupakan tabel yang menampilkan hasil pengujian *lift ratio* tertinggi dan terendah dari pola penjualan obat yang dihasilkan. Berdasarkan hasil pengujian *lift ratio* pada semua pola penjualan dihasilkan pola penjualan yang memiliki aturan asosiasi yang kuat karena nilai *lift ratio* diatas 1. Semakin tinggi nilai *lift ratio* maka aturan asosiasi antara satu item dengan item lainnya pada pola penjualan semakin kuat. Berdasarkan nilai *lift ratio* pada masing – masing bulan, maka nilai *lift ratio* tertinggi terdapat pada bulan September. Semakin tinggi nilai *lift ratio* memiliki arti bahwa jumlah item yang dibeli bersamaan dengan item lain lebih besar dari jumlah *support* dari masing - masing item tersebut.

Tabel 4. 13 Hasil Pengujian *Lift Ratio* Setiap Bulan

No	Nama Bulan	<i>Lift Ratio</i> Terendah	<i>Lift Ratio</i> Tertinggi
1	April	1,43	45,91
2	Mei	1,95	28,43
3	Juni	1,62	17,79
4	Juli	1,27	44,29
5	Agustus	2,10	30,79
6	September	3,07	89,09
7	Oktober	4,02	77,50
8	November	5,03	73,02
9	Desember	4,95	59,47

4.7.2 Pengujian Akurasi Peramalan Stok

Pengujian ini digunakan untuk menentukan tingkat kesalahan sistem dalam memberikan rekomendasi stok obat.. Tabel 4.14 menggambarkan hasil pengujian akurasi peramalan stok. Berdasarkan hasil pengujian diperoleh presentase kesalahan sistem dalam melakukan peramalan rekomendasi stok adalah 31,17%, sehingga presentase akurasi sistem adalah 68,83%. Tingginya nilai kesalahan rekomendasi karena dalam teknik pengujian MAPE tidak menggunakan sistem interval namun sistem harus benar – benar dapat memprediksi secara akurat. Selain itu juga dipengaruhi oleh jenis satuan dalam data obat.

Tabel 4. 14 Pengujian Akurasi Peramalan Stok

No	Nama Bulan	MAPE
1	April	20,27%
2	Mei	39,8%
3	Juni	31,59%
4	Juli	34,96%
5	Agustus	25,16%
6	September	33,84%
7	Oktober	30,85%
8	November	22,32%
9	Desember	41,74%
Rata – Rata		31,17%

4.7.3 Performance Testing

Tabel 3.24 merupakan hasil dari *performance testing*. Pada pengujian ini dibandingkan waktu antara algoritma apriori dan m-Apriori. Berdasarkan hasil pengujian diperoleh bahwa waktu yang dibutuhkan algoritma m-Apriori dalam menentukan pola penjualan obat adalah 1040 s atau sama dengan 17 menit dan waktu yang dibutuhkan algoritma apriori dalam menentukan pola penjualan obat adalah 1225 s atau sama dengan 20 menit.

Tabel 3. 25 Hasil Performance Testing dalam Penentuan Pola Penjualan Obat

No	Bulan	Algoritma Apriori	Algoritma m-Apriori
1	April	1028 s	920 s
2	Mei	1001 s	900 s
3	Juni	898 s	886 s
4	Juli	925 s	898 s
5	Agustus	1268 s	1104 s
6	September	1455 s	1151 s
7	Oktober	1487 s	1169 s
8	November	1500 s	1180 s
9	Desember	1465 s	1157s
Rata – Rata		1225 s	1040 s

4.7.4 Black Box Testing

Black Box Testing dilakukan dengan menjalankan sistem dan melakukan apa yang bisa dikerjakan oleh sistem untuk menguji tingkah laku dari sistem apakah keluaran yang dihasilkan sistem sudah sesuai dengan masukan yang diberikan. Sebelum melakukan pengujian, ditentukan terlebih dahulu skenario pengujian untuk memaparkan urutan yang akan diuji pada sistem ini.

Berdasarkan tabel skenario *black box testing* yang telah dibentuk, tahap selanjutnya ialah melakukan pengujian sesuai dengan yang dijelaskan pada tabel 4.5. Hasil pengujian dari skenario pengujian tersebut adalah sebagai berikut.

a. Pengujian Fungsi *Login*

Tabel 4.15 Hasil Uji Fungsi *Login Happy Path*

Identifikasi	1-1	
Nama Kasus Uji	Pengujian fungsi <i>login</i>	
Deskripsi	<i>Login</i> kondisi normal	
Tujuan	Memeriksa apakah <i>user</i> dapat masuk ke sistem	
Kondisi Awal	<i>User</i> telah memiliki akun yang telah terdaftar pada sistem	
Tanggal Pengujian	06 November 2016	
Skenario		
1	<i>User</i> membuka sistem penentuan pola penjualan obat dan rekomendasi pembelian obat terhadap <i>supplier</i>	
2	<i>User</i> memasukkan <i>Username</i> : APT01 <i>Password</i> : 12345	
Hasil yang Diharapkan	Hasil yang Didapatkan	Kesimpulan
1. Berhasil masuk ke sistem 2. Muncul halaman utama admin	1. Berhasil masuk ke sistem 2. Muncul halaman utama admin	Sesuai

Tabel 4. 16 Hasil Uji Fungsi *Login Alternative Path#1*

Identifikasi	1-1	
Nama Kasus Uji	Pengujian fungsi <i>login</i>	
Deskripsi	<i>Login</i> menggunakan akun yang tidak terdaftar pada sistem	
Tujuan	Memeriksa apakah <i>user</i> dapat masuk ke sistem	
Kondisi Awal	<i>User</i> tidak memiliki akun yang telah terdaftar pada sistem	
Tanggal Pengujian	06 November 2016	
Skenario		
1	<i>User</i> membuka sistem penentuan pola penjualan obat dan rekomendasi pembelian obat terhadap <i>supplier</i>	
2	<i>User</i> menginputkan <i>username</i> dan <i>password</i> yang tidak terdaftar	
Hasil yang Diharapkan	Hasil yang Didapatkan	Kesimpulan
1. Tidak dapat masuk ke sistem 2. Muncul pesan untuk mengisi <i>username</i> dan <i>password</i> pada <i>form login</i> 3. Halaman <i>diredirect</i> kembali ke halaman <i>login</i>	1. Tidak dapat masuk ke sistem 2. Muncul pesan <i>error</i> (<i>username</i> atau <i>password</i> salah) 3. Halaman <i>diredirect</i> kembali ke halaman <i>login</i>	Sesuai

Tabel 4. 17 Hasil Uji Fungsi *Login Alternative Path#2*

Identifikasi	1-1	
Nama Kasus Uji	Pengujian fungsi <i>login</i>	
Deskripsi	<i>Login</i> tanpa menginputkan <i>username</i> atau <i>password</i>	
Tujuan	Memeriksa apakah <i>user</i> dapat masuk ke sistem	
Kondisi Awal	<i>User</i> tidak menginputkan <i>username</i> atau <i>password</i> pada sistem	
Tanggal Pengujian	06 November 2016	
Skenario		

1	<i>User</i> membuka sistem penentuan pola penjualan obat dan rekomendasi pembelian obat terhadap <i>supplier</i>	
2	<i>User</i> tidak menginputkan <i>username</i> dan <i>password</i> pada <i>form login</i>	
Hasil yang Diharapkan	Hasil yang Didapatkan	Kesimpulan
1. Tidak dapat masuk ke sistem 2. Sistem tetap berada pada halaman <i>login</i>	1. Tidak dapat masuk ke sistem 2. Sistem tetap berada pada halaman login	Sesuai

b. Pengujian Fungsi Menampilkan Data Rekomendasi Pembelian Obat

Tabel 4. 18 Hasil Uji Menampilkan Data Rekomendasi Pembelian Obat

Identifikasi	2-1	
Nama Kasus Uji	Pengujian menampilkan data rekomendasi pembelian obat	
Deskripsi	Pengoperasian dalam kondisi normal	
Tujuan	Memeriksa apakah sistem dapat menampilkan data rekomendasi pembelian obat yang tersimpan dalam <i>database</i>	
Kondisi Awal	<i>User</i> memilih tabmenu rekomendasi pembelian obat	
Tanggal Pengujian	06 November 2016	
Skenario		
1	<i>User</i> membuka <i>form</i> utama <i>user</i> pada sistem penentuan pola penjualan obat dan rekomendasi pembelian obat terhadap <i>supplier</i>	
2	<i>User</i> memilih tabmenu rekomendasi pembelian obat	
Hasil yang Diharapkan	Hasil yang Didapatkan	Kesimpulan
1. Sistem menampilkan data rekomendasi pembelian obat	1. Sistem menampilkan data rekomendasi pembelian obat	Sesuai

c. Pengujian Fungsi Menampilkan Data Pola Penjualan Obat

Tabel 4.19 Hasil Uji Menampilkan Data Pola Penjualan Obat

Identifikasi	2-2	
Nama Kasus Uji	Pengujian menampilkan data pola penjualan obat	
Deskripsi	Pengoperasian dalam kondisi normal	
Tujuan	Memeriksa apakah sistem dapat menampilkan data pola penjualan obat yang tersimpan dalam <i>database</i>	
Kondisi Awal	<i>User</i> memilih tabmenu pola penjualan obat	
Tanggal Pengujian	06 November 2016	
Skenario		
1	<i>User</i> membuka <i>form</i> utama <i>user</i> sistem penentuan pola penjualan obat dan rekomendasi pembelian obat	
2	<i>User</i> memilih tabmenu pola penjualan obat	
Hasil yang Diharapkan	Hasil yang Didapatkan	Kesimpulan
1. Sistem menampilkan data pola penjualan obat	1. Sistem menampilkan data pola penjualan obat	Sesuai

d. Pengujian Fungsi Tambah Data Obat

Tabel 4.20 Hasil Uji Tambah Data Obat *Happy Path*

Identifikasi	3-1
Nama Kasus Uji	Pengujian fungsi menambah data obat
Deskripsi	Pengoperasian dalam kondisi normal
Tujuan	Memeriksa apakah data obat berhasil disimpan ke dalam <i>database</i>
Kondisi Awal	<i>User</i> mengisi semua <i>field</i> pada <i>form</i> data obat
Tanggal Pengujian	06 November 2016
Skenario	
1	<i>User</i> membuka <i>form</i> utama admin
2	<i>User</i> memilih tabmenu data obat

3	<i>User</i> mengklik <i>button</i> “ <i>Insert Data</i> ”	
4	<i>User</i> mengisi semua <i>field</i> pada <i>form insert data</i> obat	
Hasil yang Diharapkan	Hasil yang Didapatkan	Kesimpulan
1. Sistem berhasil melakukan penginputan data obat kedalam <i>database</i>	1. Sistem berhasil melakukan penginputan data obat kedalam <i>database</i> .	Sesuai

Tabel 4. 21 Hasil Uji Tambah Data Obat *Alternative Path#1*

Identifikasi	3-1	
Nama Kasus Uji	Pengujian fungsi menambah data obat	
Deskripsi	Menginputkan data obat dengan mengosongkan beberapa <i>field</i> pada <i>form</i> data obat	
Tujuan	Memeriksa apakah data obat berhasil disimpan ke dalam <i>database</i>	
Kondisi Awal	<i>User</i> mengosongkan beberapa <i>field</i> pada <i>form</i> data obat	
Tanggal Pengujian	06 November 2016	
Skenario		
1	<i>User</i> membuka <i>form</i> utama admin	
2	<i>User</i> memilih tabmenu data obat	
3	<i>User</i> mengklik <i>button</i> “ <i>Insert Data</i> ”	
4	<i>User</i> tidak mengisi beberapa <i>field</i> pada <i>form insert</i> data obat	
Hasil yang Diharapkan	Hasil yang Didapatkan	Kesimpulan
1. Sistem tidak dapat melakukan penginputan data 2. Muncul pesan untuk mengisi beberapa <i>field</i> yang masih kosong 3. Halaman <i>diredirect</i> kembali ke <i>form insert</i> data	1. Sistem tidak dapat melakukan penginputan data ke dalam <i>database</i> 2. Muncul pesan untuk mengisi beberapa <i>field</i> yang masih kosong 3. Halaman <i>diredirect</i> kembali ke <i>form insert</i> data obat	Sesuai

e. Pengujian Fungsi *Edit* Data Obat**Tabel 4. 22 Hasil Uji Edit Data Obat *Happy Path***

Identifikasi	3-2	
Nama Kasus Uji	Pengujian fungsi <i>edit</i> data obat	
Deskripsi	Pengoperasian dalam kondisi normal	
Tujuan	Memeriksa apakah data obat yang <i>diupdate</i> berhasil disimpan ke dalam <i>database</i>	
Kondisi Awal	<i>User</i> mengisi semua <i>field</i> yang ingin <i>diupdate</i>	
Tanggal Pengujian	06 November 2016	
Skenario		
1	<i>User</i> membuka <i>form</i> utama admin pada sistem penentuan pola penjualan obat dan rekomendasi pembelian obat terhadap <i>supplier</i>	
2	<i>User</i> memilih tabmenu data obat	
3	<i>User</i> mengklik <i>icon</i> “ <i>Edit</i> ”	
4	<i>User</i> mengisi semua <i>field</i> pada form data obat yang ingin <i>diupdate</i>	
Hasil yang Diharapkan	Hasil yang Didapatkan	Kesimpulan
1. Sistem dapat menyimpan data yang telah <i>diedit</i> kedalam <i>database</i>	1. Sistem dapat menyimpan data yang telah <i>diedit</i> kedalam <i>database</i>	Sesuai

Tabel 4. 23 Hasil Uji Edit Data Obat *Alternatif Path#1*

Identifikasi	3-2
Nama Kasus Uji	Pengujian fungsi <i>edit</i> data obat
Deskripsi	Mengedit data obat dengan mengosongkan beberapa <i>field</i> pada <i>form</i> data obat
Tujuan	Memeriksa apakah data obat yang telah <i>diedit</i> berhasil disimpan ke dalam <i>database</i>
Kondisi Awal	<i>User</i> mengosongkan beberapa <i>field</i> pada <i>form</i> data obat yang <i>diedit</i>
Tanggal Pengujian	06 November 2016
Skenario	

1	<i>User</i> membuka <i>form</i> utama admin pada sistem penentuan pola penjualan obat dan rekomendasi pembelian obat terhadap <i>supplier</i>	
2	<i>User</i> memilih tabmenu data obat	
3	<i>User</i> mengklik <i>icon</i> “ <i>Edit</i> ”	
4	<i>User</i> tidak mengisi semua <i>field</i> pada form data obat yang ingin <i>diupdate</i>	
Hasil yang Diharapkan	Hasil yang Didapatkan	Kesimpulan
1. Sistem tidak dapat melakukan penginputan data obat yang telah <i>diupdate</i> ke dalam <i>database</i> 2. Muncul pesan untuk mengisi beberapa <i>field</i> yang masih kosong 3. Halaman <i>diredirect</i> kembali ke <i>form edit</i> data obat	1. Sistem tidak dapat melakukan penginputan data obat yang telah <i>diupdate</i> ke dalam <i>database</i> 2. Muncul pesan untuk mengisi beberapa <i>field</i> yang masih kosong 3. Halaman <i>diredirect</i> kembali ke <i>form edit</i> data obat	Sesuai

f. Pengujian Fungsi Hapus Data Obat

Tabel 4. 24 Hasil Uji Hapus Data Obat

Identifikasi	3-3
Nama Kasus Uji	Pengujian fungsi hapus data obat
Deskripsi	Pengoperasian dalam kondisi normal
Tujuan	Memeriksa apakah data obat yang dipilih berhasil dihapus dari <i>database</i>
Kondisi Awal	<i>User</i> memilih data obat yang akan dihapus
Tanggal Pengujian	06 November 2016
Skenario	

1	<i>User</i> membuka <i>form</i> utama admin pada sistem penentuan pola penjualan obat dan rekomendasi pembelian obat terhadap <i>supplier</i>	
2	<i>User</i> memilih tabmenu data obat	
3	<i>User</i> mengklik <i>icon</i> “Hapus”	
Hasil yang Diharapkan	Hasil yang Didapatkan	Kesimpulan
1. Data obat berhasil dihapus dari <i>database</i>	1. Data obat berhasil dihapus dari <i>database</i>	Sesuai

g. Pengujian Fungsi Menampilkan Data Obat

Tabel 4. 25 Hasil Uji Menampilkan Data Obat

Identifikasi	3-4	
Nama Kasus Uji	Pengujian menampilkan data obat	
Deskripsi	Pengoperasian dalam kondisi normal	
Tujuan	Memeriksa apakah sistem dapat menampilkan data obat yang tersimpan dalam <i>database</i>	
Kondisi Awal	<i>User</i> memilih tabmenu data obat	
Tanggal Pengujian	06 November 2016	
Skenario		
1	<i>User</i> membuka <i>form</i> utama admin	
2	<i>User</i> memilih tabmenu data obat	
Hasil yang Diharapkan	Hasil yang Didapatkan	Kesimpulan
1. Sistem dapat menampilkan data obat yang tersimpan dalam <i>database</i>	1. Sistem dapat menampilkan data obat yang tersimpan dalam <i>database</i>	Sesuai

h. Pengujian Fungsi Tambah Data Transaksi Penjualan Obat

Tabel 4. 26 Hasil Uji Tambah Data Transaksi Penjualan Obat *Happy Path*

Identifikasi	4-1	
Nama Kasus Uji	Pengujian fungsi menambah data transaksi penjualan obat	
Deskripsi	Pengoperasian dalam kondisi normal	
Tujuan	Memeriksa apakah data transaksi penjualan obat berhasil disimpan ke dalam <i>database</i>	
Kondisi Awal	<i>User</i> mengisi semua <i>field</i> pada <i>form</i> data transaksi penjualan obat	
Tanggal Pengujian	06 November 2016	
Skenario		
1	<i>User</i> membuka <i>form</i> utama admin	
2	<i>User</i> memilih tabmenu data transaksi penjualan	
3	<i>User</i> mengklik <i>button</i> “Insert Data”	
4	<i>User</i> mengisi semua <i>field</i> pada <i>form</i> insert data	
Hasil yang Diharapkan	Hasil yang Didapatkan	Kesimpulan
1. Data transaksi penjualan obat berhasil disimpan kedalam <i>database</i>	1. Data transaksi penjualan obat berhasil disimpan kedalam <i>database</i>	Sesuai

Tabel 4. 27 Hasil Uji Tambah Data Transaksi Penjualan *Alternatif Path#1*

Identifikasi	4-1
Nama Kasus Uji	Pengujian fungsi menambah data transaksi penjualan obat
Deskripsi	Menginputkan data transaksi penjualan obat dengan mengosongkan beberapa <i>field</i> pada <i>form</i> data transaksi penjualan
Tujuan	Memeriksa apakah data transaksi penjualan obat berhasil disimpan ke dalam <i>database</i>
Kondisi Awal	<i>User</i> mengosongkan beberapa <i>field</i> pada <i>form</i> data transaksi penjualan obat
Tanggal Pengujian	06 November 2016
Skenario	
1	<i>User</i> membuka <i>form</i> utama admin
2	<i>User</i> memilih tabmenu data transaksi penjualan

3	<i>User mengklik button “Insert Data”</i>	
4	<i>User tidak mengisi beberapa field pada form insert data transaksi penjualan</i>	
Hasil yang Diharapkan	Hasil yang Didapatkan	Kesimpulan
1. Sistem tidak dapat melakukan penginputan data ke dalam <i>database</i> 2. Muncul pesan untuk mengisi beberapa <i>field</i> yang masih kosong 3. Halaman <i>redirect</i> kembali ke <i>form insert</i> data transaksi penjualan obat	1. Sistem tidak dapat melakukan penginputan data ke dalam <i>database</i> 2. Muncul pesan untuk mengisi beberapa <i>field</i> yang masih kosong 3. Halaman <i>redirect</i> kembali ke <i>form insert</i> data transaksi penjualan obat	Sesuai

e. Pengujian Fungsi *Edit* Data Transaksi Penjualan Obat

Tabel 4. 28 Hasil Uji *Edit* Data Transaksi Penjualan Obat *Happy Path*

Identifikasi	4-2
Nama Kasus Uji	Pengujian fungsi <i>edit</i> data transaksi penjualan obat
Deskripsi	Pengoperasian dalam kondisi normal
Tujuan	Memeriksa apakah data transaksi penjualan obat yang <i>diupdate</i> berhasil disimpan ke dalam <i>database</i>
Kondisi Awal	<i>User</i> mengisi semua <i>field</i> yang ingin <i>diupdate</i>
Tanggal Pengujian	06 November 2016
Skenario	
1	<i>User</i> membuka <i>form</i> utama admin pada sistem penentuan pola penjualan obat dan rekomendasi pembelian obat terhadap <i>supplier</i>
2	<i>User</i> memilih tabmenu data transaksi penjualan
3	<i>User</i> mengklik icon “ <i>Edit</i> ”

4	<i>User</i> mengisi semua <i>field</i> pada form data transaksi penjualan yang ingin <i>diupdate</i>	
Hasil yang Diharapkan	Hasil yang Didapatkan	Kesimpulan
1. Sistem dapat menyimpan data yang telah <i>diedit</i> kedalam <i>database</i>	1. Sistem dapat menyimpan data yang telah <i>diedit</i> kedalam <i>database</i>	Sesuai

Tabel 4. 29 Hasil Uji Edit Data Transaksi Penjualan Obat Alternatif Path#1

Identifikasi	4-2	
Nama Kasus Uji	Pengujian fungsi <i>edit</i> data transaksi penjualan obat	
Deskripsi	Mengedit data transaksi penjualan obat dengan mengosongkan beberapa <i>field</i> pada <i>form</i> data transaksi penjualan obat	
Tujuan	Memeriksa apakah data transaksi penjualan obat yang telah <i>diedit</i> berhasil disimpan ke dalam <i>database</i>	
Kondisi Awal	<i>User</i> mengosongkan beberapa <i>field</i> pada <i>form</i> data transaksi penjualan obat yang <i>diedit</i>	
Tanggal Pengujian	06 November 2016	
Skenario		
1	<i>User</i> membuka <i>form</i> utama admin	
2	<i>User</i> memilih tabmenu data transaksi penjualan	
3	<i>User</i> mengklik <i>icon</i> “ <i>Edit</i> ”	
4	<i>User</i> tidak mengisi semua <i>field</i> pada form data transaksi penjualan obat yang ingin <i>diupdate</i>	
Hasil yang Diharapkan	Hasil yang Didapatkan	Kesimpulan

1. Sistem tidak dapat melakukan penginputan data transaksi penjualan obat	1. Sistem tidak dapat melakukan penginputan data transaksi penjualan obat yang telah <i>diupdate</i> ke dalam <i>database</i>	Sesuai
2. Muncul pesan untuk mengisi beberapa <i>field</i> yang masih kosong	2. Muncul pesan untuk mengisi beberapa <i>field</i> yang masih kosong	
3. Halaman <i>diredirect</i> kembali ke <i>form edit</i> data transaksi	3. Halaman <i>diredirect</i> kembali ke <i>form edit</i> data transaksi penjualan obat	

f. Pengujian Fungsi Hapus Data Transaksi Penjualan Obat

Tabel 4. 30 Hasil Uji Hapus Data Transaksi Penjualan Obat

Identifikasi	4-3	
Nama Kasus Uji	Pengujian fungsi hapus data transaksi penjualan obat	
Deskripsi	Pengoperasian dalam kondisi normal	
Tujuan	Memeriksa apakah data transaksi penjualan obat yang dipilih berhasil dihapus dari <i>database</i>	
Kondisi Awal	<i>User</i> memilih data transaksi penjualan obat yang akan dihapus	
Tanggal Pengujian	06 November 2016	
Skenario		
1	<i>User</i> membuka <i>form</i> utama admin	
2	<i>User</i> memilih tabmenu data transaksi penjualan	
3	<i>User</i> mengklik <i>icon</i> “Hapus”	
Hasil yang Diharapkan	Hasil yang Didapatkan	Kesimpulan
1. Data transaksi penjualan obat berhasil dihapus dari <i>database</i>	1. Data transaksi penjualan obat berhasil dihapus dari <i>database</i>	Sesuai

g. Pengujian Fungsi Menampilkan Data Transaksi Penjualan Obat

Tabel 4. 31 Hasil Uji Menampilkan Data Transaksi Penjualan Obat

Identifikasi	4-4	
Nama Kasus Uji	Pengujian menampilkan data transaksi penjualan obat	
Deskripsi	Pengoperasian dalam kondisi normal	
Tujuan	Memeriksa apakah sistem dapat menampilkan data transaksi penjualan obat yang tersimpan dalam <i>database</i>	
Kondisi Awal	<i>User</i> memilih tabmenu data transaksi penjualan	
Tanggal Pengujian	06 November 2016	
Skenario		
1	<i>User</i> membuka <i>form</i> utama admin pada sistem penentuan pola penjualan obat dan rekomendasi pembelian obat terhadap <i>supplier</i>	
2	<i>User</i> memilih tabmenu data transaksi penjualan	
Hasil yang Diharapkan	Hasil yang Didapatkan	Kesimpulan
1. Sistem dapat menampilkan data transaksi penjualan obat yang tersimpan dalam <i>database</i>	1. Sistem dapat menampilkan data transaksi penjualan obat yang tersimpan dalam <i>database</i>	Sesuai

4.7.5 White Box Testing

White box testing adalah teknik meramalkan cara kerja perangkat lunak secara rinci, karena *logical path* (jalur logika) perangkat lunak akan di *test* dengan menyediakan *test case* yang akan mengerjakan kumpulan kondisi atau pengulangan secara spesifik. Pada pengujian *white box testing* untuk menguji algoritma m-Apriori dibagi menjadi dua bagian yaitu pengujian untuk penentuan *itemset* dan pengujian untuk pembentukan kombinasi *itemset*.

4.7.5.1 Pengujian White Box Testing untuk Algoritma m-Apriori

Berikut ini adalah proses pengujian logika m-Apriori untuk menentukan pola penjualan obat.

Tabel 4. 32 *Flowgraph* Penentuan *Itemset*

No	Keterangan	<i>Flowgraph</i>
1	Input data transaksi dan data obat	<pre> graph TD 12((1,2)) --> 3((3)) 3 --> 4((4)) 4 --> 5((5)) 4 --> 6((6)) 5 --> 7((7)) 6 --> 7 </pre>
2	Inisialisasi nilai minimum <i>support</i> dan minimum <i>confidence</i>	
3	Hitung frekuensi <i>itemset</i>	
4	Bandingkan nilai frekuensi <i>itemset</i> dengan minimum <i>support</i>	
5	Hapus data <i>itemset</i>	
6	Simpan data <i>itemset</i>	
7	Perulangan sampai semua frekuensi <i>itemset</i> dihitung	

Perhitungan nilai *cyclomatic complexity* alur proses penentuan satu *itemset* berdasarkan data transaksi penjualan adalah sebagai berikut.

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 6 - 6 + 2 \\
 &= 2
 \end{aligned}$$

Berdasarkan jumlah *cyclomatic complexity* maka dapat dibuat jalur (*path*) pengujian sesuai *flow graph*.

$$Path\ 1 = 1 - 2 - 3 - 4 - 6 - 7$$

$$Path\ 2 = 1 - 2 - 3 - 4 - 5 - 7$$

Nilai yang didapat adalah 2, dimana terdapat 2 jalur pengujian yang didapat berdasarkan perhitungan dari *flowgraph* tersebut. Nilai *cyclomatic complexity* yang dihasilkan tergolong prosedur sederhana, tanpa banyak resiko. Tabel 4.29 menjelaskan kasus uji pada setiap *path* yang terbentuk.

Tabel 4. 33 Kasus Uji Penentuan *Itemset*

<i>Path</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Output</i>	<i>Keterangan</i>
1	Data Obat : {Opimer, Pamol, Trogyl} Transaksi 1 : {Opimer,pamol} Transaksi 2 : {Trogyl} Transaksi 3 : {Trogyl, Pamol, Opimer} Min support = 2 Min confidence = 0,5	Data <i>itemset</i> : Opimer Pamol Trogyl	Data <i>itemset</i> : Opimer Pamol Trogyl	Sesuai
2	Data Obat : {Opimer, Pamol, Trogyl, Nonemi} Transaksi 1 : {Opimer, Pamol} Transaksi 2 : {Pamol, Opimer, Trogyl} Transaksi 3 : {Pamol, Opimer} Transaksi 4 : {Nonemi} Min support = 2 Min confidence = 0,5	Data <i>itemset</i> : Opimer Pamol	Data <i>itemset</i> : Opimer Pamol	Sesuai

Tabel 4. 34 *Flowgraph* Pembentukan Kombinasi *Itemset*

No	Keterangan	Flowgraph
1	Input data <i>itemset</i>	<pre> graph TD 1((1)) --> 2((2)) 2 --> 3((3)) 3 --> 456((4,5,6)) 456 --> 7((7)) 7 --> 8((8)) 8 --> 9((9)) 9 --> 10((10)) 10 --> 11((11)) 11 --> 12((12)) 8 --> 3 11 --> 2 </pre>
2	Pembentukan kombinasi <i>itemset</i>	
3	Pecah kombinasi <i>itemset</i> menjadi item tunggal	
4	Bandingkan frekuensi <i>iteemset</i> dalam satu kombinasi	
5	Pilih transaksi yang mengandung <i>itemset</i> dengan nilai terkecil	
6	Hitung frekuensi kombinasi	
7	Membandingkan nilai frekuensi kombinasi dengan nilai minimum <i>support</i>	
8	Hapus data kombinasi	
9	Simpan data kombinasi <i>itemset</i>	
10	Perulangan sampai semua kombinasi ditentukan frekuensinya	
11	Perulangan sampai tidak dapat dibentuk kombinasi kembali	
12	Buat pola dari kombinasi <i>itemset</i> terpilih	

Perhitungan nilai *cyclomatic complexity* alur proses penentuan kombinasi *itemset* berdasarkan data *itemset* yang telah ditentukan pada tabel 4.28 adalah sebagai berikut.

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 12 - 10 + 2 \\
 &= 4
 \end{aligned}$$

Berdasarkan jumlah *cyclomatic complexity* maka dapat dibuat jalur (*path*) pengujian sesuai *flow graph*.

Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 9 – 10 – 11 – 12

Path 2 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 10 – 11 – 12

Path 3 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 9 – 10 – 3 – 4 – 5 – 6 – 7 – 9 – 10 – 11 – 12

Path 4 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 9 – 10 – 11 – 2 – 3 – 4 – 5 – 6 – 7 – 9 – 10 – 11 – 12

Nilai yang didapat adalah 4, dimana terdapat 4 jalur pengujian yang didapat berdasarkan perhitungan dari *flowgraph* tersebut. Nilai *cyclomatic complexity* yang dihasilkan tergolong prosedur sederhana, tanpa banyak resiko. Tabel 4.31 menjelaskan kasus uji pada setiap *path* yang terbentuk.

Tabel 4. 35 Kasus Uji Penentuan Kombinasi *Itemset*

<i>Path</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Output</i>	Keterangan
1	Data <i>itemset</i> = {Opimer,Pamol} min_sup = 2 min_conf = 0,5	Kombinasi 2 <i>itemset</i> = {Opimer, Pamol}	Kombinasi 2 <i>itemset</i> = {Opimer, Pamol}	Sesuai
2	Data <i>itemset</i> = {Opimer, Pamol, TrogyI} min_sup = 2 min_conf = 0,5	Kombinasi 2 <i>itemset</i> = {Opimer, Pamol}	Kombinasi 2 <i>itemset</i> = {Opimer, Pamol}	Sesuai
3	Data <i>itemset</i> = {Opimer, Pamol, TrogyI}	Kombinasi 2 <i>itemser</i> = {Opimer,Pamol} {TrogyI,Pamol}	Kombinasi 2 <i>itemser</i> = {Opimer,Pamol} {TrogyI,Pamol}	Sesuai
4	Data <i>itemset</i> = {Opimer, Pamol,TrogyI}	Kombinasi 2 <i>itemset</i> = {Opimer,Pamol} {Opimer,TrogyI} Kombinasi 3 <i>itemset</i> = {Opimer, Pamol, TrogyI}	Kombinasi 2 <i>itemset</i> = {Opimer, Pamol} {Opimer, TrogyI} Kombinasi 3 <i>itemset</i> = {Opimer, Pamol, TrogyI}	Sesuai