

## **BAB III**

### **METODOLOGI**

#### **3.1 Studi Pustaka**

Studi pustaka dalam penelitian ini dilakukan dengan cara membaca dan menganalisis literatur yang berkaitan dengan pengendalian persediaan barang, metode klasifikasi ABC, metode peramalan moving average serta metode peramalan weighted moving average yang pernah dilakukan sebelumnya yang bersumber dari jurnal, paper, situs web, dan *e-book* yang berkaitan.

##### **3.1.1 Analisis Permasalahan**

Apotek UII Farma tidak mempunyai sistem pengadaan barang sehingga pengadaan obat dilakukan secara tidak terencana. Pengadaan obat yang dilakukan pihak apotek tergantung dari hasil pengecekan stok yang dilakukan setiap minggunya, apabila stok dirasa menipis maka saat itu juga pihak apotek akan memesan obat untuk menambah stok. Selain pengadaan obat yang dilakukan tidak menentu waktunya, banyaknya obat yang dipesan juga hanya berdasarkan dari perkiraan tanpa adanya perhitungan khusus. Waktu pemesanan obat yang tidak menentu akan berdampak semakin besarnya biaya transportasi yang harus ditanggung, dan banyaknya obat yang dipesan hanya sebatas perkiraan dapat berdampak suatu obat memiliki sisa stok yang banyak sekali dan pada saat yang bersamaan obat yang lain kekurangan stok karena habis terlebih dahulu. Kedua hal ini dapat menjadi penyebab semakin lambatnya perputaran dana yang ada di Apotek UII Farma itu sendiri.

Apotek UII Farma memiliki tiga golongan obat yang terbagi menurut hukum peredarannya yaitu obat bebas, obat bebas terbatas, dan obat keras. Akan tetapi selama ini pihak apotek tidak pernah membedakan golongan obat pada saat pengadaan obat. Padahal terkait ketiga golongan tersebut sebagaimana dijelaskan pada subbab 2.2.1 tentang hukum peredarannya, obat keras merupakan obat yang paling sulit untuk diedarkan karena pihak apotek hanya diperbolehkan menjual obat golongan ini kepada pembeli yang memiliki resep dokter saja. Oleh karenanya akan kurang tepat apabila mengabaikan golongan obat pada saat pengadaan obat.

Karena obat keras dinilai memiliki tingkat peredaran yang paling sulit, maka sistem pengadaan obat yang akan dibangun ini akan difokuskan untuk meramalkan kebutuhan obat keras saja. Sejalan dengan hukum pareto yang meyakini bahwa setiap golongan pasti ada suatu

kelompok kecil yang memiliki dampak terbesar, sehingga pada penelitian ini dari obat keras yang terbilang paling sulit untuk diedarkan akan dicari tahu terlebih dahulu obat keras apa saja yang memiliki dampak terbesar dengan cara mengelompokkan obat keras kedalam kelas-kelas tertentu menggunakan metode ABC seperti yang pernah diterapkan peneliti terdahulu terhadap kasus yang serupa. Penerapan metode ABC pada penelitian ini dilakukan sebelum proses peramalan karena bertujuan agar peramalan yang nantinya akan dilakukan hanya berpusat pada obat keras yang memiliki prioritas tertinggi saja. Berbeda dengan penelitian yang dilakukan oleh (Effendi, Ong, & Gunawan, 2011) pada penelitiannya proses klasifikasi menggunakan metode ABC dilakukan setelah hasil peramalan didapatkan karena tujuannya adalah hanya ingin mencari tahu barang yang dapat menghasilkan laba tertinggi. Lalu untuk peramalan yang akan dilakukan pada penelitian ini menggunakan metode WMA seperti yang diterapkan oleh (Shinta, Susanto, & Wivia, 2015) pada penelitiannya.

Penelitian ini mencoba untuk menerapkan kedua metode yang telah disebutkan sebelumnya terhadap pengadaan obat yang dilakukan oleh Apotek UII Farma agar pengadaan obat yang sebelumnya dilakukan dengan tidak terencana menjadi pengadaan obat yang terencana. Terkait perhitungan yang dilakukan pada kedua metode tersebut sebenarnya dapat dilakukan menggunakan aplikasi yang umum digunakan yaitu Microsoft Excel (MS Excel). Akan tetapi karyawan apotek yang memiliki kewenangan untuk mengelola logistik obat hanya bisa menggunakan MS Excel hanya untuk perhitungan sederhana saja. Proses pengadaan obat terkait pengklasifikasian dan peramalan kebutuhan obat memerlukan rumus-rumus yang terbilang rumit untuk karyawan yang tidak mahir menggunakan MS Excel. Oleh karenanya diperlukan sebuah aplikasi perhitungan yang dikhususkan untuk pengadaan obat guna membantu karyawan tersebut dalam perhitungannya.

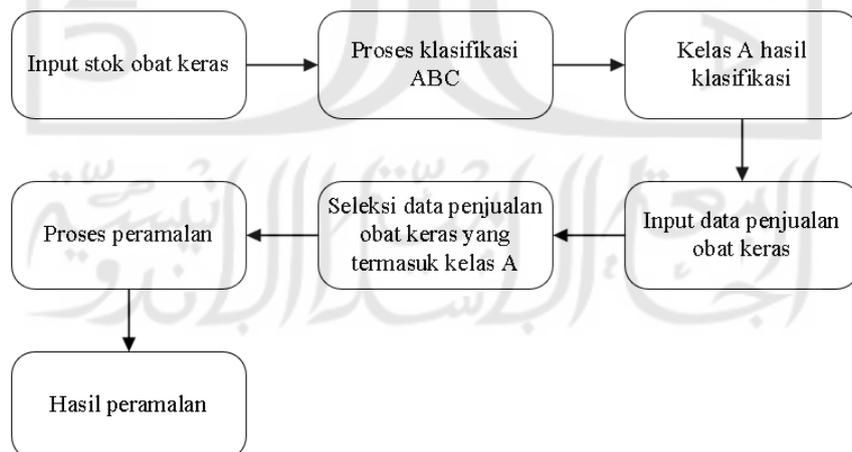
### **3.1.2 Pengumpulan Data**

Pengumpulan data yang ingin dilakukan adalah dengan cara mengambil data primer yang bersumber langsung dari *database* obat yang dimiliki Apotek UII Farma terkait daftar obat golongan obat keras, stok terbaru obat keras, dan data penjualannya. Dikarenakan hak akses agar dapat mengakses langsung *database* obat tidak berhasil diperoleh, sehingga data yang berhasil diambil peneliti merupakan data *output* dari sistem yang digunakan oleh pihak apotek dan tersimpan kedalam *file excel*.

### 3.2 Gambaran Umum Sistem

Pada awalnya sistem yang ingin dibangun adalah sistem yang dapat diintegrasikan dengan sistem pengelolaan barang yang digunakan oleh Apotek UII Farma, sehingga sistem yang akan dibangun ini dapat secara langsung mengambil data yang diperlukan dari *database* yang ada. Seperti yang telah dijelaskan pada subbab 3.1.2, peneliti tidak berhasil mendapatkan hak untuk mengakses *database* apotek secara langsung sehingga harus merubah cara memasukan data pada sistem yang akan dibangun ini. Pada akhirnya proses memasukkan data akan dilakukan dengan cara menyimpan data dalam *file* excel agar bisa terbaca oleh sistem yang akan dibangun.

Sistem pengadaan obat yang akan dibangun pada penelitian ini diawali dengan memasukkan data stok obat keras yang akan diklasifikasikan kedalam kelas ABC. Setelah proses klasifikasi ABC akan menghasilkan data obat yang termasuk dalam kelas kategori A. Apabila *user* ingin melanjutkan ke proses peramalan kebutuhan maka *user* diharuskan untuk memasukkan data penjualan bulanan kedalam sistem. Data obat kelas kategori A hasil klasifikasi sebelumnya akan digunakan sebagai *filter* untuk data penjualan obat yang akan diramalkan permintaannya. Setelah didapatkan data penjualan obat yang termasuk kedalam kelas kategori A maka proses yang akan berjalan setelahnya adalah proses peramalan kebutuhan kepada data penjualan bulanan obat yang termasuk kedalam kelas kategori A. Hasil dari peramalan akan ditampilkan pada sistem. **Error! Reference source not found.** berikut ini merupakan gambaran dari arsitektur sistem yang akan dibuat. (contoh)



Gambar 3.1 Diagram alir sistem.

## Pengklasifikasian

Proses klasifikasi yang dimaksudkan pada penelitian ini adalah pengelompokan data dengan cara memberikan label kelas pada data itu menggunakan metode ABC. Data yang dikelompokkan adalah data stok obat keras terbaru yang ada di Apotek UII Farma. Kelas yang digunakan pada metode tersebut terdiri dari tiga yaitu kelas A yang memiliki prioritas tertinggi, kelas B yang memiliki prioritas sedang, dan kelas C yang memiliki prioritas rendah. Ilustrasi dari alir klasifikasi secara umum bisa dilihat pada Gambar 3.2 berikut:



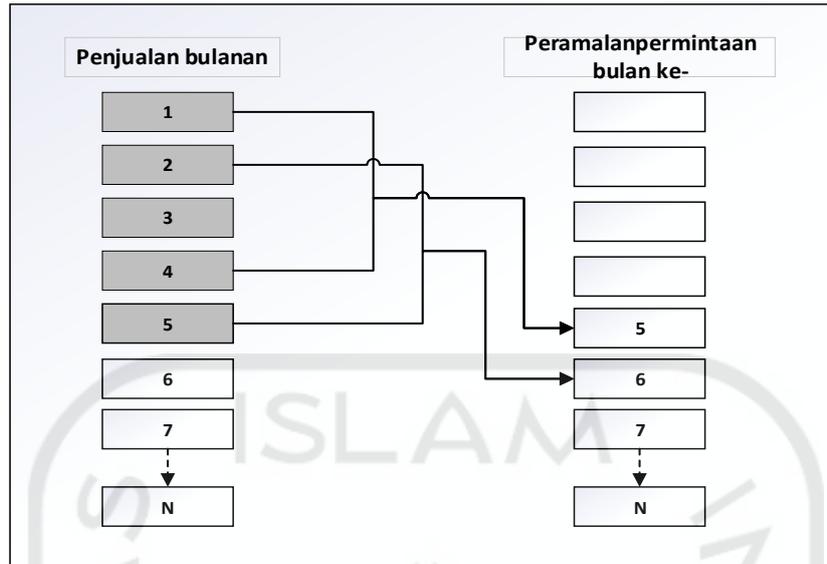
Gambar 3.2 Ilustrasi pengklasifikasian

Pada proses di atas data obat yang dibutuhkan adalah daftar obat keras yang berisikan nama, stok dan harga obat keras. Setelah proses klasifikasi selesai dilakukan sistem akan mengambil daftar obat yang memiliki label “A” (kelas teratas) sebagai acuan obat apa saja yang akan dimasukkan kedalam proses peramalan kebutuhan.

Besaran persentase kelas ABC menggunakan perbandingan 20:30:50 yang mana 20% dari jenis barang diklasifikasikan kedalam kelas A, 30% dari banyaknya jenis barang diklasifikasikan kedalam kelas B, dan 50% dari banyaknya jenis barang diklasifikasikan kedalam kelas C (Gaspersz, 2006).

## Peramalan

Proses peramalan kebutuhan menggunakan metode WMA didasarkan dari data-data penjualan periode sebelumnya. Apabila ingin melakukan peramalan berdasarkan 4 periode maka untuk setiap periode yang ingin diramalkan harus menggunakan 4 periode sebelumnya sebagai acuan. Gambaran umum alur perhitungan peramalan kebutuhan bisa dilihat pada Gambar 3.3 Ilustrasi peramalan kebutuhan berikut:



Gambar 3.3 Ilustrasi peramalan kebutuhan

Pada ilustrasi di atas kotak berwarna abu-abu adalah periode yang sudah berlalu dan memiliki data penjualan untuk periode tersebut. Karena peramalan kebutuhan menggunakan acuan 4 periode maka proses perhitungan dimulai dari periode kelima.

### 3.3 Analisis Kebutuhan Sistem

#### 3.3.1 Kebutuhan Data

Untuk dapat mengklasifikasi serta meramalkan permintaan obat keras maka diperlukan beberapa data yang berhubungan dengan penghitungan saat klasifikasi serta peramalan kebutuhan obat. Data yang diperlukan adalah sebagai berikut:

a. Data obat keras

Penelitian ditujukan kepada golongan obat keras, maka dibutuhkan daftar dari obat yang termasuk dalam golongan obat keras. Sedangkan data yang berhasil diperoleh dari pihak apotek merupakan data yang berisi daftar dari semua obat yang berjumlah 2.454 macam obat, dan tidak ada keterangan yang secara langsung menggolongkan obat berdasarkan golongan obat bebas, obat bebas terbatas, atau obat keras. Namun menurut karyawan apotek, obat yang diletakkan pada rak keenam hanyalah obat keras. Sehingga daftar obat keras berhasil didapatkan dengan jumlah 183 macam obat.

b. Data stok obat

Karena pengklasifikasian hanya dilakukan terhadap obat berdasarkan harga total untuk setiap unit, maka dibutuhkan data stok dan harga obat dari obat yang termasuk golongan obat keras sebagai data masukan.

c. Data penjualan bulanan

Peramalan kebutuhan obat yang ingin dilakukan berdasarkan penjualan obat periode sebelumnya, maka data penjualan obat juga diperlukan sebagai data masukan.

### 3.3.2 Kebutuhan Masukan

Masukan yang dibutuhkan oleh sistem pengadaan obat berupa data-data obat yang telah disebutkan pada *subbab* 3.2 yang mana data obat tersebut ditulis menggunakan *format* penulisan tertentu dan telah tersimpan kedalam *file* bertipe “.xls”. Tabel 3.1 berikut ini merupakan tabel kebutuhan *input* data berdasarkan proses yang berjalan:

Tabel 3.1 Kebutuhan masukan

Proses	Masukan
Klasifikasi	Data stok obat keras
Peramalan	Data penjualan bulanan

Data stok obat keras yang diperlukan untuk proses klasifikasi terdiri dari nama, stok, dan harga dari obat keras terbaru. Sedangkan data yang diperlukan pada proses peramalan berisikan nama, data penjualan bulan ke-1, data penjualan bulan ke-2, data penjualan bulan ke-3, data penjualan bulan ke-4, data penjualan bulan ke-5 dan data penjualan bulan ke-6. Data penjualan tersebut merupakan data penjualan enam bulan terakhir dimana data penjualan bulan ke-6 merupakan data penjualan terbaru dan data penjualan bulan ke-1 adalah data penjualan yang paling lama.

### 3.3.3 Kebutuhan Proses

Berikut ini merupakan proses yang diperlukan pada saat pengadaan obat:

a. Proses klasifikasi

Pada saat pengklasifikasian obat terdapat beberapa sub-proses yang akan dijalankan nantinya. Proses-proses itu sendiri adalah sebagai berikut:

1. Proses *import* data obat keras  
Pada tahapan ini *user* diminta untuk memilih *file* excel yang berisi data dari stok obat keras dan memasukkannya ke dalam sistem.
2. Proses membaca data  
Setelah *file* excel dimasukkan, pada tahapan ini isi dari *file* tersebut akan dibaca oleh sistem.
3. Proses pengambilan nilai  
Data stok obat keras yang sudah berhasil dibaca akan disimpan ke dalam sistem.
4. Proses perhitungan harga total untuk setiap obat  
Pada proses ini total harga dari setiap data obat keras yang sudah berhasil disimpan akan dihitung dengan cara mengalikan stok dengan harga satuan yang dimilikinya.
5. Proses *sorting* data  
Pada proses ini data obat keras akan diurutkan dari obat keras yang memiliki harga total terbesar hingga terkecil.
6. Proses penjumlahan total keseluruhan harga  
Pada proses ini akan harga dari semua obat keras yang telah disimpan oleh sistem akan dijumlahkan.
7. Proses perhitungan banyaknya macam obat.  
Pada tahapan ini akan dihitung seberapa banyak macam obat keras yang datanya berhasil disimpan kedalam sistem.
8. Proses pemberian besaran persentase anggota kelas  
Besaran persentase anggota kelas A, B, dan C ditentukan pada tahapan ini.
9. Proses perhitungan banyaknya anggota kelas  
Setelah besaran persentase anggota untuk setiap kelas diketahui, pada tahap ini akan dilakukan perhitungan besaran persentase tersebut terhadap banyaknya obat yang ada sehingga diketahui berapa banyak anggota untuk kelas A, B dan C.
10. Proses pemberian label kelas
11. Proses perhitungan persentase kumulatif harga kelas  
Pada tahap ini akan dihitung seberapa besar persentase harga yang dimiliki setiap kelas terhadap total keseluruhan harga.

b. Proses peramalan

Pada proses peramalan daftar nama obat dari kelas kategori A akan dijadikan sebagai acuan perhitungan. Oleh sebab itu sub-proses yang akan dijalankan pada saat peramalan adalah sebagai berikut:

1. Proses *input* data penjualan

Pada tahap ini *user* akan diminta untuk memasukkan *file* excel yang berisi data penjualan obat keras 6 selama bulan terakhir.

2. Proses baca data penjualan

Setelah *file* excel dimasukkan, pada tahapan ini isi dari *file* tersebut akan dibaca oleh sistem.

3. Proses pengambilan data penjualan

Data penjualan obat dari file excel yang telah dipilih akan disimpan ke dalam sistem.

4. Proses baca data obat hasil klasifikasi

Hasil klasifikasi obat keras yang termasuk kedalam kelas A akan diambil.

5. Proses *filter* data

Data penjualan bulanan yang tergolong ke dalam obat dari kelas A dan selalu terjual setiap bulannya akan disimpan oleh sistem.

6. Proses penentuan periode acuan

Pada tahap ini ditentukan seberapa banyak periode yang nantinya digunakan sebagai acuan peramalan.

7. Proses pemberian bobot untuk setiap periode acuan.

8. Proses perhitungan rata-rata berbobot

Pada tahap ini data penjualan akan dikalikan dengan bobot sesuai dengan urutannya dari data yang akan diramalkan.

9. Proses perhitungan *error*

10. Proses *export* data

Untuk melihat data hasil peramalan *user* harus memberikan perintah *export* data agar hasil perhitungan peramalan yang telah dilakukan akan ditulis dan disimpan ke dalam *file* excel.

### 3.3.4 Kebutuhan Keluaran

Keluaran yang dibutuhkan pada saat sistem dijalankan dapat dilihat pada Tabel 3.2 berikut:

Tabel 3.2 Kebutuhan keluaran

Proses	Keluaran	Keterangan
Klasifikasi	Informasi stok obat keras	Menampilkan data stok obat yang telah <i>diinput</i> kedalam tabel.
	Informasi hasil klasifikasi	Menampilkan hasil dari proses klasifikasi.
Peramalan	Data hasil peramalan	Mengekspor hasil perhitungan dari proses peramalan yang dilakukan kedalam <i>file excel</i> .

### 3.3.5 Kebutuhan Antarmuka

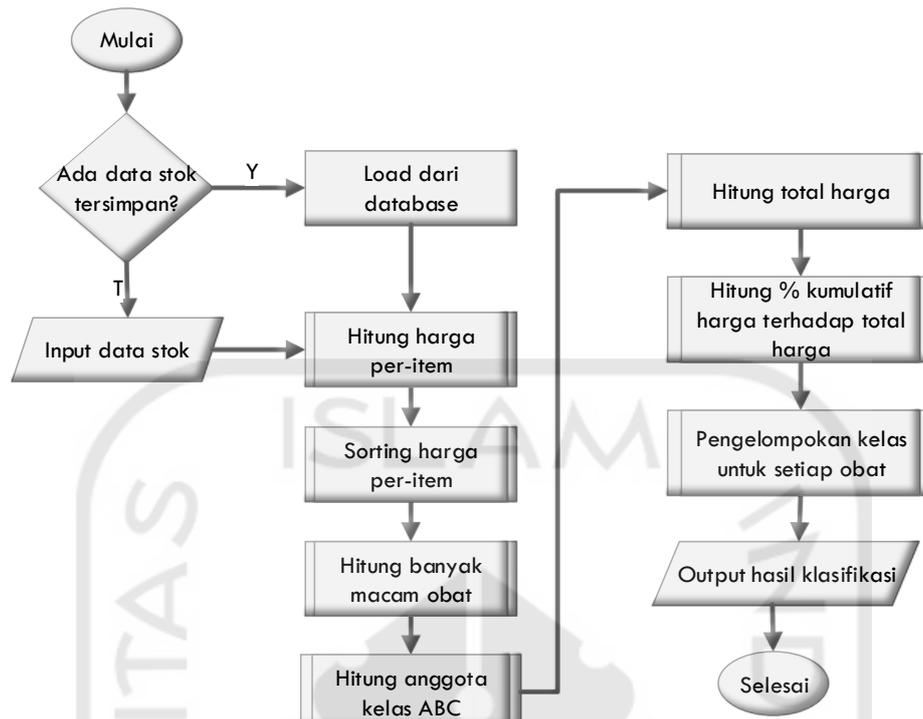
Antarmuka dari sistem perlu dibuat untuk menjadi perantara di antara *user* dan sistem pengadaan obat. Dengan adanya antarmuka *user* akan lebih mudah melakukan pengadaan obat karena hanya perlu menekan beberapa tombol saja. Antarmuka sistem juga akan memberikan informasi dari obat keras yang termasuk kedalam kelas A yang merupakan hasil dari proses klasifikasi.

## 3.4 Desain

### 3.4.1 Proses Klasifikasi ABC

Pada tahapan ini sistem akan meminta masukan berupa data stok obat keras yang ada. Data yang akan dimasukkan tersebut harus disesuaikan formatnya sebagaimana yang akan didefinisikan pada subbab 3.3.2 poin b. Setelah itu sistem akan mengurutkan data dari obat yang memiliki nilai terbesar ke nilai terkecil. Proses selanjutnya adalah proses perhitungan berapa banyak obat yang ada. Setelah diketahui berapa banyaknya obat yang ada akan ditentukan berapa banyak anggota untuk setiap kelas A, B dan C. Proses yang akan berjalan setelahnya adalah proses perhitungan harga keseluruhan dari semua obat yang ada. Setelah jumlah keseluruhan harga obat diketahui, sistem akan menghitung berapa persentase harga setiap obat terhadap harga keseluruhan obat. Setelah sistem akan memberikan label kelas pada setiap obat yang ada.

Proses klasifikasi ABC ini bertujuan untuk mencari tahu obat mana saja yang termasuk kedalam kelas kategori A. Gambar 3.4 berikut ini merupakan diagram alir dari proses klasifikasi ABC.



Gambar 3.4 Diagram alir proses klasifikasi ABC.

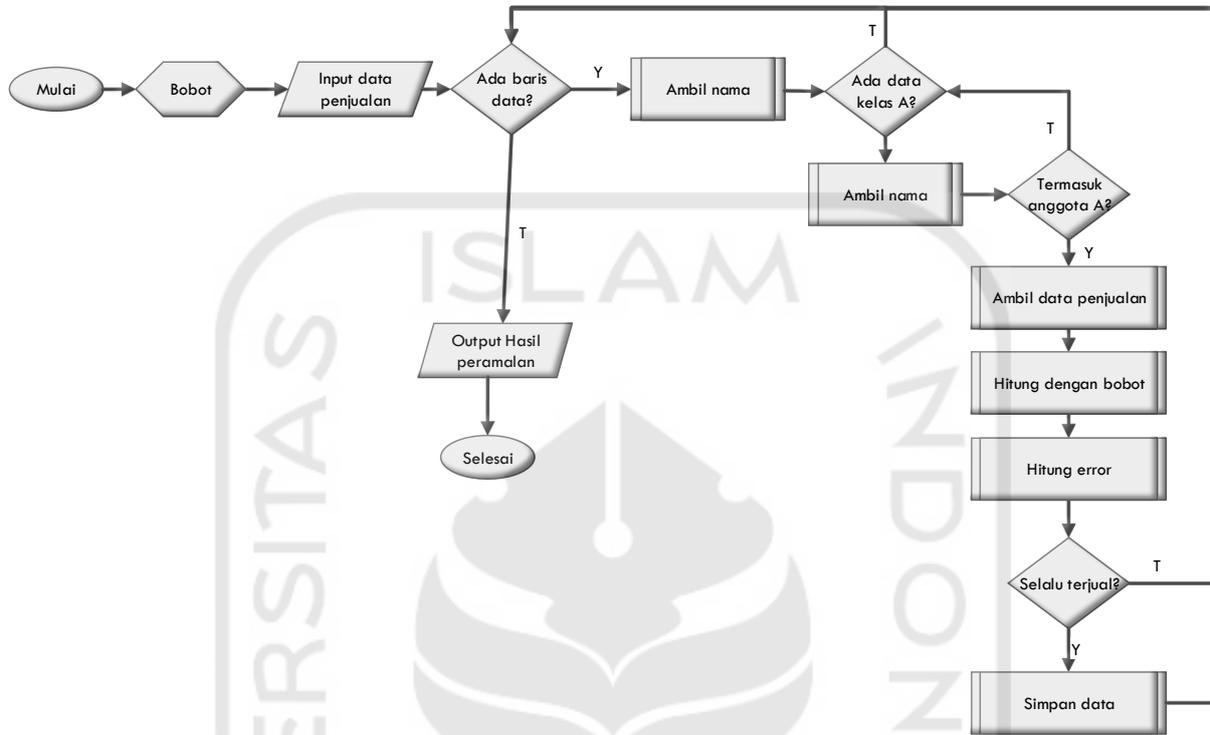
### 3.4.2 Proses Peramalan

Pada tahapan ini akan dilakukan permintaan masukan data penjualan bulanan obat keras. Karena proses peramalan ditujukan untuk obat keras kelas kategori A maka proses peramalan hanya bisa dijalankan setelah proses klasifikasi. Saat proses peramalan berjalan hal yang pertama kali dilakukan adalah *reload* kembali data kelas kategori A hasil klasifikasi. Setelah itu *user* diminta untuk memasukkan data penjualan bulanan.

Sistem akan mengambil data nama obat dari kelas kategori A. Setelah itu sistem akan melakukan penyaringan data penjualan obat keras yang termasuk kedalam kelas kategori A. Apabila ada data penjualan obat keras yang tergolong kelas kategori A, maka sistem akan mengambil nilai dari data penjualan bulanan obat itu dan dikalikan dengan bobot yang telah ditentukan. Data yang akan digunakan sebagai acuan adalah data penjualan 4 bulan sebelumnya. Bobot yang akan digunakan adalah 0.1, 0.2, 0.3 dan 0.4 berurutan dari data penjualan terlama hingga data penjualan terbaru. Setelah data dikalikan dengan masing-masing bobot, data tersebut akan dijumlahkan lalu dibagi dengan total bobot.

Hasil dari perhitungan di atas merupakan nilai peramalan kebutuhan obat untuk periode penjualan ke- $x$  ( $x$  merupakan banyaknya periode acuan ditambah 1). Perhitungan itu akan terus diulang hingga tidak lagi dibutuhkan peramalan untuk obat itu. Sistem akan mengulangi

proses-proses perhitungan di atas untuk setiap data penjualan obat yang tergolong kedalam kelas kategori A. Gambar 3.5 berikut ini merupakan ilustrasi diagram alir dari proses peramalan kebutuhan menggunakan metode WMA.



Gambar 3.5 Diagram alir proses peramalan.

### 3.4.3 Rancangan Antarmuka

Perancangan antarmuka pada penelitian ini diperlukan untuk mempermudah pengguna pada saat mengoperasikan sistem nantinya. Berikut ini adalah rancangan yang akan digunakan sebagai dasar pembuatan *interface* dari sistem yang akan dibuat. Rancangan *interface* itu sendiri secara garis besar terbagi menjadi 2 menurut proses yang dijalankan, yakni *interface* untuk proses klasifikasi dan *interface* untuk proses peramalan.

Berikut ini beberapa rancangan interface yang akan menjadi dasar pembuatan sistem:

a. Halaman *Input* Klasifikasi

Halaman *input* klasifikasi berfungsi untuk memasukkan serta menampilkan data obat keras yang sudah dimasukkan untuk proses klasifikasi. Rancangan *interface* untuk halaman ini bisa dilihat pada Gambar 3.6 berikut ini:

Gambar 3.6 Rancangan halaman *input* klasifikasi.

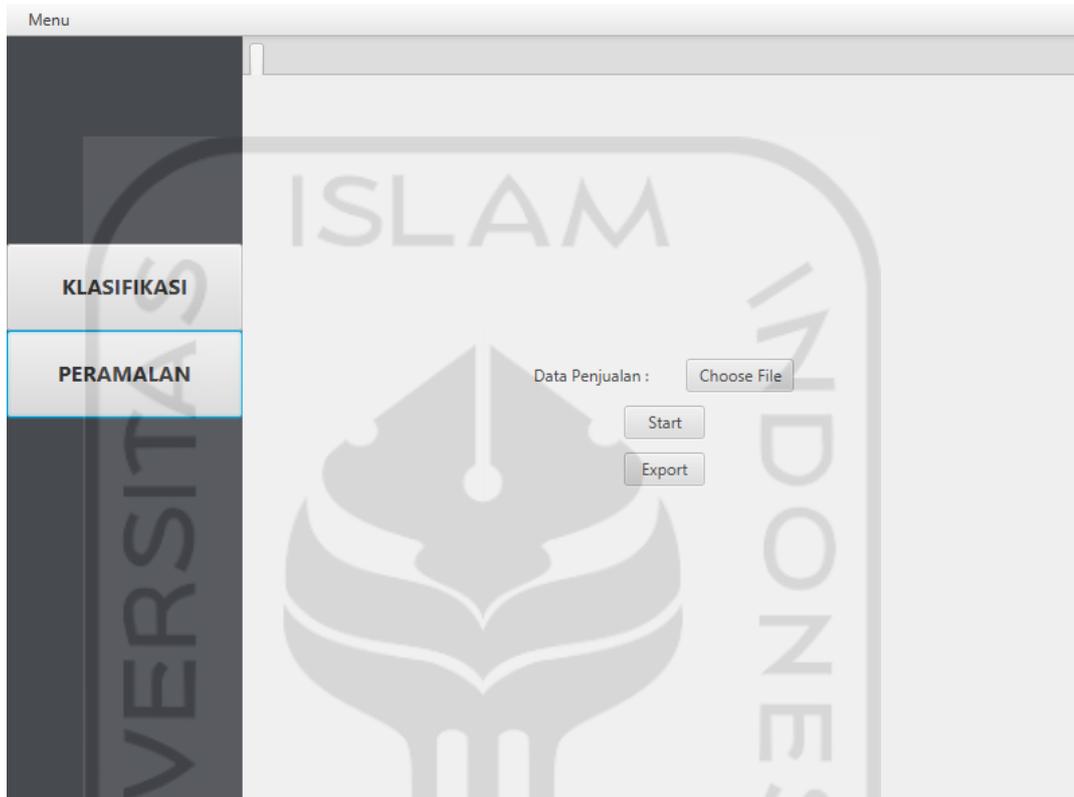
b. Halaman Hasil Klasifikasi

Halaman hasil klasifikasi berfungsi untuk menampilkan hasil pengklasifikasian dari data obat keras yang sudah dimasukkan di halaman sebelumnya. Rancangan *interface* untuk halaman ini bisa dilihat pada Gambar 3.7 berikut ini:

Gambar 3.7 Rancangan halaman hasil klasifikasi.

c. Halaman Peramalan

Halaman Peramalan berfungsi untuk memasukkan data penjualan obat keras serta menampilkan hasil dari proses peramalan. Rancangan *interface* untuk halaman ini bisa dilihat pada Gambar 3.8 berikut ini:



Gambar 3.8 Rancangan halaman peramalan.

### 3.5 Pengkodean

Pada tahapan ini akan dilakukan pembuatan program. Bahasa pemrograman yang akan digunakan adalah Java dan JavaFXML. Tahapan ini merupakan tahapan pembuatan sistem pengadaan obat sebagaimana yang sudah direncanakan sebelumnya. Dari hasil perancangan lalu diwujudkan sebagai perangkat lunak yang sebenarnya yang memiliki tampilan sistem.

#### 3.5.1 Proses Klasifikasi

Berikut ini merupakan implementasi dari subproses yang ada pada proses klasifikasi:

a. Proses *input* data

Sebelum proses klasifikasi dimulai, *user* diharuskan memasukkan data stok obat keras kedalam sistem. Cara *input* itu sendiri terbagi menjadi 2 yaitu *input* dari *file* excel dan *input* manual satu-persatu. Sebagaimana yang telah disebutkan pada Tabel 3.1, data yang

dibutuhkan pada proses ini adalah data stok obat keras. Data stok obat keras ini terdiri dari nama, stok, dan harga.

Tabel 3.3 berikut ini merupakan contoh format penulisan data stok obat pada *file excel*:

Tabel 3.3 Contoh data stok dan harga obat

Nama	Qty	Harga Satuan
Abate 1 GR 10 g/Sachet	46.00	2,010.00
Abbocath 20	1.00	15,000.00
ABBOCATH No 22	2.00	15,000.00
Absolute Feminine Hygiene 60 ml	1.00	15,145.90
Acanthe Lightening Gel 10 ml	0.40	157,181.90
Acanthe SPF 30 30 gr	3.21	63,276.26
Acarbose 50 mg	20.00	1,078.00
Ace Max	1.00	172,500.00
Aceton 50 cc	5.00	6,700.00
Acetylcysteine Cap 200 mg	10.00	1,000.00

*Syntax* dari proses pemilihan *file* bisa dilihat pada Gambar 3.9 berikut:

```
@FXML
private String pilihFileIn(ActionEvent event) throws IOException {
    FileChooser chooser = new FileChooser();
    File f = chooser.showOpenDialog(null);
    lokasi = f.getCanonicalPath();
    nmFile.setText(f.getName());
    return lokasi; }

```

Gambar 3.9 *Syntax* untuk memilih *file*.

#### b. Proses baca data

Proses baca data merupakan proses yang berfungsi untuk membaca nilai setiap *cell* dari *file excel* yang telah dipilih. Untuk dapat membaca isi dari *cell*, sistem akan membaca baris dan kolom dari data yang telah dimasukkan. *Syntax* dari proses baca data ini dapat dilihat pada Gambar 3.10 berikut:

```
public void bacaFile() {
    File f = new File(lokasi);
    try {
        Workbook wb = Workbook.getWorkbook(f);
        Sheet s = wb.getSheet(0);
        Cell c;

        int row = s.getRows();
        int col = s.getColumns();

        for (int i = 0; i < row; i++) {
            String nm1 = null;

```

```

double qt1 = 0, hr1 = 0;

for (int j = 0; j < col; j++) {

    c = s.getCell(j, i);
    if (j == 0) {
        nml = c.getContents();
    } else if (j == 1) {
        qt1 = Double.valueOf(c.getContents());
    } else if (j == 2) {
        hr1 = Double.valueOf(c.getContents());
    }
}

Obat io = new Obat(nml, qt1, hr1);
io.setHargaPerItem();
arrObt.add(io);
}

} catch (IOException | BiffException ex) {
    Logger.getLogger(FXMLHomeController.class.getName()).log(Level.SEVERE,
null, ex);
}
}

```

Gambar 3.10 *Syntax* untuk membaca isi *file*.

c. Proses pengambilan nilai

Proses pengambilan data merupakan proses yang berfungsi unruk mengambil nilai dari data yang sudah dibaca serta merubah tipe data yang sebelumnya terbaca sebagai *String* menjadi tipe data *double*. Perubahan data ini diperlukan sebab akan dilakukan perhitungan terkait data yang sudah diambil nantinya. *Syntax* dari proses pengambilan nilai data ini dapat dilihat pada Gambar 3.11 berikut:

```

for (int i = 0; i < row; i++) {
String nml = null;
double qt1 = 0, hr1 = 0;
    for (int j = 0; j < col; j++) {
        c = s.getCell(j, i);
        if (j == 0) {
            nml = c.getContents();
        } else if (j == 1) {
            qt1 = Double.valueOf(c.getContents());
        } else if (j == 2) {
            hr1 = Double.valueOf(c.getContents());
        }
    }
    Obat io = new Obat(nml, qt1, hr1);
    io.setHargaPerItem();
    arrObt.add(io);
}
}

```

Gambar 3.11 *Syntax* untuk mengambil nilai data.

Pada penerapannya proses pengambilan data ini dilakukan bersamaan dengan proses baca data.

d. Proses perhitungan harga obat

Proses penghitungan harga obat berfungsi untuk menghitung harga untuk setiap obat yang ada. Harga obat didapatkan dari mengalikan banyaknya stok dengan harga satuan obat.

*Syntax* dari proses ini dapat dilihat pada Gambar 3.12 berikut:

```
public void totalPerItem(){
    DecimalFormat df = new DecimalFormat("#.00");
    double t = this.qty*this.harga;
    this.hargaPerItem = Double.valueOf(df.format(t));
}
```

Gambar 3.12 *Syntax* perhitungan harga obat.

e. Proses penjumlahan harga seluruh obat

Proses ini berfungsi untuk menghitung total harga dari seluruh obat yang ada. *Syntax* dari proses perhitungan banyaknya obat dapat dilihat pada Gambar 3.13 berikut:

```
public void hitungTotal() {
    double t = 0;
    for (int i = 0; i < arrObt.size(); i++) {
        t = t + (arrObt.get(i).getHargaPerItem());
    }
    total = t;
}
```

Gambar 3.13 *Syntax* perhitungan total harga obat.

f. Proses perhitungan banyaknya obat

Proses ini berfungsi untuk mencari tahu berapa banyak obat keras yang telah dimasukkan kedalam sistem. Banyaknya obat bisa dicari tahu dari berapa banyak baris data yang telah diambil pada proses sebelumnya. *Syntax* dari proses perhitungan banyaknya obat dapat dilihat pada Gambar 3.14 berikut:

```
arrObt.size()
```

Gambar 3.14 *Syntax* perhitungan harga obat.

g. Proses perhitungan besaran persentase kumulatif harga

Proses ini berfungsi untuk menghitung berapa besar persentase harga obat terhadap harga total keseluruhan obat. *Syntax* dari proses perhitungan ini dapat dilihat pada Gambar 3.15 berikut:

```
public void hitungPersen() {
    for (int i = 0; i < arrObt.size(); i++) {
        double ph = (arrObt.get(i).getHargaPerItem() / total) * 100;
        double ph2 = (double) Math.round(ph * prs) / prs;
        arrObt.get(i).setPersenHarga(ph2);
    }
}
```

Gambar 3.15 *Syntax* perhitungan persentase harga obat.

h. Proses perhitungan anggota kelas ABC

Proses ini berfungsi untuk menghitung berapa banyak anggota untuk setiap kelas ABC. Sebagaimana yang telah dirancangkan, anggota dari kelas A adalah 20% dari banyaknya

obat, anggota dari kelas A adalah 30% dari banyaknya obat dan anggota dari kelas A adalah 50% dari banyaknya obat. *Syntax* dari proses perhitungan ini dapat dilihat pada Gambar 3.16 berikut:

```
public void hitungAnggota() {
    anggA = Math.round((arrObt.size() * 20) / 100);
    anggB = Math.round((arrObt.size() * 30) / 100);
    anggC = arrObt.size()-(anggA+anggB);
}
```

Gambar 3.16 *Syntax* perhitungan anggota kelas ABC.

i. Proses pemberian label kelas

Pemberian label kelas dilakukan agar memudahkan saat pemanggilan nilai dari data obat yang termasuk kedalam kelas tertentu. *Syntax* dari proses pelabelan ini dapat dilihat pada Gambar 3.17 berikut:

```
public void beriLabel() {
    for (int i = 0; i < arrObt.size(); i++) {
        if (i < anggA) {
            arrObt.get(i).setKelas('A');
            persenA = persenA + arrObt.get(i).getPersenHarga();
        } else if (i >= anggA && i < (anggA + anggB)) {
            arrObt.get(i).setKelas('B');
            persenB = persenB + arrObt.get(i).getPersenHarga();
        } else {
            arrObt.get(i).setKelas('C');
            persenC = persenC + arrObt.get(i).getPersenHarga();
        }
    }
}
```

Gambar 3.17 *Syntax* pemberian label kelas ABC.

j. Proses perhitungan persentase kumulatif harga total dari kelas ABC

Proses perhitungan ini dilakukan untuk mengetahui seberapa besar total harga yang dimiliki oleh kelas A terhadap total harga keseluruhan obat. *Syntax* dari proses perhitungannya dapat dilihat pada Gambar 3.18 berikut:

```
for (int i = 0; i < arrObt.size(); i++){
    hob.add(new Obat(arrObt.get(i).getNama(), arrObt.get(i).getHargaPerItem(),
arrObt.get(i).getPersenHarga(), arrObt.get(i).getKelas()));
    if (arrObt.get(i).getKelas()=='A'){
        persenA = persenA + arrObt.get(i).getPersenHarga();
    }else if (arrObt.get(i).getKelas()=='B'){
        persenB = persenB + arrObt.get(i).getPersenHarga();
    }else {
        persenC = persenC + arrObt.get(i).getPersenHarga();
    }
}
```

Gambar 3.18 *Syntax* perhitungan kumulatif harga kelas ABC.

### 3.5.2 Proses Peramalan

Setelah proses klasifikasi selesai, proses yang akan dijalankan selanjutnya adalah proses peramalan obat. Seperti halnya proses klasifikasi, proses ini juga mempunyai beberapa sub-proses. Berikut ini merupakan sub-proses dari proses peramalan:

a. Proses baca data obat kelas kategori A

Proses ini berfungsi untuk membaca anggota kelas A dari proses klasifikasi yang telah dilakukan sebelumnya. *Syntax* dari proses baca data dapat dilihat pada Gambar 3.19 berikut:

```
for (Obat ob : arrObt) {
    if (ob.getNama().equalsIgnoreCase(cek)) {
    }
}
```

Gambar 3.19 *Syntax* baca anggota kelas kategori A.

b. Proses *input* data penjualan

Proses ini berfungsi untuk memasukkan data penjualan obat 6 bulan terakhir. Eksekusi dari proses ini sama persis dengan proses *input* data stok obat. Perbedaan dari kedua proses input tersebut hanyalah isi dari data yang dimasukkan. Contoh format penulisan data penjualan dapat dilihat pada Tabel 3.4 berikut ini:

Tabel 3.4 Contoh data penjualan obat

Nama	April	Mei	Juni	Juli	Agustus	September
Tremenza Tab	1145	1493	854	1364	1094	1051
Rhinos SR	505	527	340	642	534	506
Ericaf	68	113	133	82	83	85
Rhinofed	65	82	52	97	35	96
Lapifed Tablet	27	39	36	146	98	61
Gabapentin Tab 300 mg	35	116	35	107	52	50
Primolut N(AT)	95	10	60	72	30	0
Glucovance 250mg/1,25mg	20	40	30	20	10	60
Zyloric 300	14	10	39	46	6	33
Daonil 5mg	20	30	40	0	30	20
Glucovance 500mg/2.5mg	0	60	10	20	0	10
Fluimucil 200 Mg	11	30	20	23	10	0
Amaryl 3 mg	15	18	38	0	0	10
Leparson	0	24	0	0	20	30
Myonep Tab	10	16	10	20	8	5

Meptin Tablet 0,05 mg	9	38	0	2	8	8
Myonal	11	10	0	10	5	0
Diane - 35 /Strip box	7	9	5	5	4	3
Euphyllin Retard Mite	0	0	0	23	0	0
Duphaston 10mg tab	0	10	0	0	0	0

c. Proses baca data penjualan

Proses ini berfungsi untuk membaca isi data penjualan pada proses sebelumnya. *Syntax* dari proses baca data dapat dilihat pada Gambar 3.20 berikut:

```
public void bacaFile2() {
    File f = new File(lokasi);
    try {
        Workbook wb = Workbook.getWorkbook(f);
        Sheet s = wb.getSheet(0);
        Cell c1, c2;
        int row = s.getRows();
        int col = s.getColumns();
        for (int i = 0; i < row; i++) {
            String cek = c1.getContents();
            for (Obat ob : arrObt) {
                if (ob.getNama().equalsIgnoreCase(cek)) {
                    for (int j = 0; j < col; j++) {
                        c2 = s.getCell(j, i);
                    }
                }
            }
        }
    } catch (IOException | BiffException ex) {
        Logger.getLogger(FXMLHomeController.class.getName()).log(Level.SEVERE,
            null, ex);
    }
}
```

Gambar 3.20 *Syntax* baca data penjualan.

d. Proses pengambilan nilai data penjualan

Proses ini berfungsi untuk mengambil nilai dari data penjualan yang telah dimasukkan.

*Syntax* dari proses ini dapat dilihat pada Gambar 3.21 berikut:

```
String nm1 = null;
double b11 = 0;
double b12 = 0;
double b13 = 0;
double b14 = 0;
double b15 = 0;
double b16 = 0;
String cek = c1.getContents();
for (Obat ob : arrObt) {
    if (ob.getNama().equalsIgnoreCase(cek)) {
        for (int j = 0; j < col; j++) {
            c2 = s.getCell(j, i);
            if (j == 0) {nm1 = c2.getContents();}
            } else if (j == 1) {
                b11 = Double.valueOf(c2.getContents());
            } else if (j == 2) {
                b12 = Double.valueOf(c2.getContents());
            } else if (j == 3) {
                b13 = Double.valueOf(c2.getContents());
            } else if (j == 4) {
```

```

        b14 = Double.valueOf(c2.getContents());
    } else if (j == 5) {
        b15 = Double.valueOf(c2.getContents());
    } else if (j == 6) {
        b16 = Double.valueOf(c2.getContents());
    }
    f5 = ((b11 * bobot1) + (b12 * bobot2) + (b13 * bobot3) + (b14 * bobot4))
/ totalBobot;
    f6 = ((b12 * bobot1) + (b13 * bobot2) + (b14 * bobot3) + (b15 * bobot4))
/ totalBobot;
    f7 = ((b13 * bobot1) + (b14 * bobot2) + (b15 * bobot3) + (b16 * bobot4))
/ totalBobot;
    }
    Penjualan pj = new Penjualan(nml, b11, b12, b13, b14, b15, b16, f5, f6, f7);
    pj.setMad(mad1);
    forObt.add(pj);

```

Gambar 3.21 *Syntax* pengambilan nilai data penjualan.

e. Proses filter data

Proses ini berfungsi sebagai penyaring agar data penjualan yang akan diambil hanya data penjualan obat yang termasuk kedalam hasil klasifikasi dan juga terjual pada setiap bulan.

*Syntax* dari proses ini dapat dilihat pada Gambar 3.22 berikut:

```

c1 = s.getCell(0, i);
String cek = c1.getContents();
for (Obat ob : arrObt) {
    if (ob.getNama().equalsIgnoreCase(cek)) {
}}

```

Gambar 3.22 *Syntax filter* data.

Pada penerapannya proses baca data obat kelas A, proses baca data penjualan, proses pengambilan nilai data penjualan dan proses *filter* data tergabung kedalam sebuah *method*.

f. Proses penentuan banyaknya periode acuan

Proses ini berfungsi untuk menentukan berapa banyak periode penjualan yang nantinya akan digunakan sebagai acuan peramalan. Seperti yang telah direncanakan, periode yang akan digunakan sebagai acuan adalah 4 bulan.

g. Proses pemberian bobot

Proses ini berfungsi untuk menentukan berapa bobot yang akan diberikan untuk setiap periode acuan. Besaran bobot yang akan diberikan akan disesuaikan dengan yang telah direncanakan yaitu 0.1, 0.2, 0.3 dan 0.4 berurutan dari data penjualan terlama hingga data penjualan terbaru. *Syntax* dari proses ini dapat dilihat pada Gambar 3.23 berikut:

```

double bobot1, bobot2, bobot3, bobot4, totalBobot;

bobot1 = 0.1;
bobot2 = 0.2;
bobot3 = 0.3;
bobot4 = 0.4;
totalBobot = bobot1 + bobot2 + bobot3 + bobot4;

```

Gambar 3.23 *Syntax* pemberian bobot.

h. Proses perhitungan rata-rata berbobot

Pada proses ini akan dilakukan perhitungan terkait bobot yang telah diberikan kepada setiap periode acuan. Setiap data penjualan dari periode acuan akan dikalikan dengan bobot yang telah diberikan. Hasil perkalian untuk setiap data akan dijumlahkan lalu dibagi dengan total bobot. Syntax dari proses ini dapat dilihat pada Gambar 3.24 berikut:

```
double f5, f6, f7;

f5 = ((b11 * bobot1) + (b12 * bobot2) + (b13 * bobot3) + (b14 * bobot4)) /
totalBobot;
f6 = ((b12 * bobot1) + (b13 * bobot2) + (b14 * bobot3) + (b15 * bobot4)) /
totalBobot;
f7 = ((b13 * bobot1) + (b14 * bobot2) + (b15 * bobot3) + (b16 * bobot4)) /
totalBobot;
```

Gambar 3.24 Syntax perhitungan rata-rata berbobot.

#### i. Proses perhitungan *error*

Pada proses ini akan dilakukan perhitungan untuk mengetahui berapa besaran nilai *error*. *Error* yang dimaksud adalah nilai absolut selisih hasil peramalan dengan data aktual. Syntax dari proses ini dapat dilihat pada Gambar 3.25 berikut:

```
public void hitungMad() {
    DecimalFormat df = new DecimalFormat("#.00");
    double x = (Math.abs(this.real5-this.forc1));
    double y = (Math.abs(this.real6-this.forc2));
    double m = ((x + y) / 2);
    this.mad = Double.valueOf(df.format(m));}

public void hitungMape(){
    DecimalFormat df = new DecimalFormat("#.00");
    double k = (Math.abs(this.real5- this.forc1))/this.real5*100;
    double l = (Math.abs(this.real6- this.forc2))/this.real6*100;
    double m = (k+l)/2;
    this.mape = Double.valueOf(df.format(m)); }
```

Gambar 3.25 Syntax perhitungan *error*

#### j. Proses ekspor data

Proses ekspor data ini berfungsi untuk mengekspor tabel perhitungan lengkap terkait peramalan yang dilakukan kedalam *file* excel. Syntax dari proses ini dapat dilihat pada Gambar 3.26 berikut:

```
public void exportToExcel() throws IOException, WriteException {

    String filePath = "C:\\Users\\WIndows 10\\Desktop\\Peramalan.xls";
    WritableWorkbook workBook = null;

    try {
        workBook = Workbook.createWorkbook(new File(filePath)); //inisialisasi
        workBook
        WritableSheet sheet = workBook.createSheet("Peramalan", 0); //membuat
        sheet peramalan
        WritableFont headerCellFont = new WritableFont(WritableFont.COURIER,
        10,
            WritableFont.BOLD, true); //font style dari headernya
        WritableCellFormat headerCellFormat = new
        WritableCellFormat(headerCellFont); //membuat format header
```

```

//membuat nama kolom
Label headerCell1 = new Label(0, 0, "Nama", headerCellFormat);
Label headerCell2 = new Label(1, 0, "Bulan 1", headerCellFormat);
Label headerCell3 = new Label(2, 0, "Bulan 2", headerCellFormat);
Label headerCell4 = new Label(3, 0, "Bulan 3", headerCellFormat);
Label headerCell5 = new Label(4, 0, "Bulan 4", headerCellFormat);
Label headerCell6 = new Label(5, 0, "Bulan 5", headerCellFormat);
Label headerCell7 = new Label(6, 0, "Bulan 6", headerCellFormat);
Label headerCell8 = new Label(8, 0, "Forecast 5", headerCellFormat);
Label headerCell9 = new Label(9, 0, "Forecast 6", headerCellFormat);
Label headerCell10 = new Label(10, 0, "Forecast 7", headerCellFormat);
Label headerCell11 = new Label(11, 0, "MAD", headerCellFormat);
Label headerCell12 = new Label(12, 0, "MAPE", headerCellFormat);

//tambahkan nama kolom ke sheet
sheet.addCell(headerCell1);
sheet.addCell(headerCell2);
sheet.addCell(headerCell3);
sheet.addCell(headerCell4);
sheet.addCell(headerCell5);
sheet.addCell(headerCell6);
sheet.addCell(headerCell7);
sheet.addCell(headerCell8);
sheet.addCell(headerCell9);
sheet.addCell(headerCell10);
sheet.addCell(headerCell11);
sheet.addCell(headerCell12);
int j = 1;
for (Penjualan pj : forObt) { //panjang barisdata sama dengan arraylist

    Label rmlNama = new Label(0, j, pj.getNama());
    Label rmlBln1 = new Label(1, j, df.format(pj.getReal1()));
    Label rmlBln2 = new Label(2, j, df.format(pj.getReal2()));
    Label rmlBln3 = new Label(3, j, df.format(pj.getReal3()));
    Label rmlBln4 = new Label(4, j, df.format(pj.getReal4()));
    Label rmlBln5 = new Label(5, j, df.format(pj.getReal5()));
    Label rmlBln6 = new Label(6, j, df.format(pj.getReal6()));
    Label rmlFrcBln5 = new Label(8, j, df.format(pj.getForc1()));
    Label rmlFrcBln6 = new Label(9, j, df.format(pj.getForc2()));
    Label rmlFrcBln7 = new Label(10, j, df.format(pj.getForc3()));
    Label rmlMad = new Label(11, j, df.format(pj.getMad()));
    Label rmlMape = new Label(12, j, df.format(pj.getMape()) + "%");

    //tambah cell ke sheet
    sheet.addCell(rmlNama);
    sheet.addCell(rmlBln1);
    sheet.addCell(rmlBln2);
    sheet.addCell(rmlBln3);
    sheet.addCell(rmlBln4);
    sheet.addCell(rmlBln5);
    sheet.addCell(rmlBln6);
    sheet.addCell(rmlFrcBln5);
    sheet.addCell(rmlFrcBln6);
    sheet.addCell(rmlFrcBln7);
    sheet.addCell(rmlMad);
    sheet.addCell(rmlMape);

    j++;
}

//atur lebar kolom
sheet.setColumnView(0, 23);
sheet.setColumnView(1, 13);
sheet.setColumnView(2, 13);
sheet.setColumnView(3, 13);
sheet.setColumnView(4, 13);

```

```

        sheet.setColumnView(5, 13);
        sheet.setColumnView(6, 13);
        sheet.setColumnView(7, 13);
        sheet.setColumnView(8, 13);
        sheet.setColumnView(9, 13);
        sheet.setColumnView(10, 13);
        sheet.setColumnView(11, 13);
        sheet.setColumnView(12, 13);

        //write workbook
        workBook.write();
        keteranganExport.setText("Data sudah ter-ekspor.");
    } catch (IOException e) {
        e.printStackTrace();
    } catch (RowsExceededException e) {
        e.printStackTrace();
    } catch (WriteException e) {
        e.printStackTrace();
    } finally {
        //tutup workbook
        workBook.close();
    }
}

```

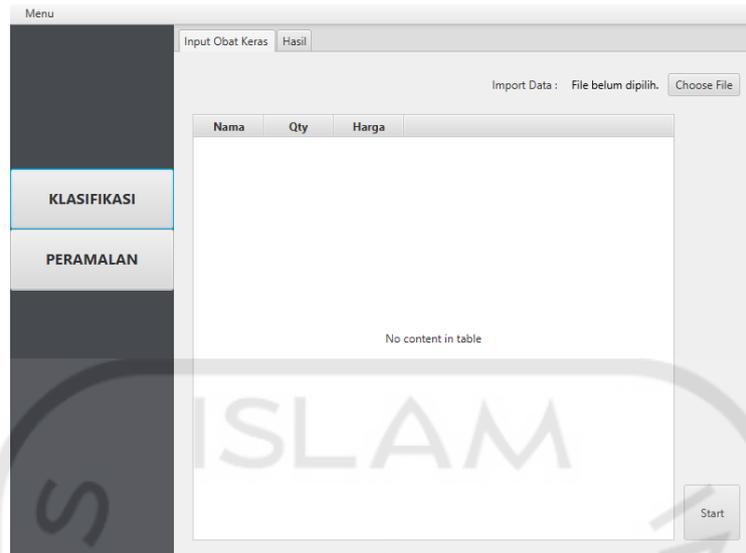
Gambar 3.26 *Syntax* ekspor perhitungna peramalan

### 3.5.3 Tampilan Halaman

Terdapat tiga tampilan halaman pada sistem yang di buat, yakni halaman *input* untuk proses klasifikasi, halaman hasil klasifikasi dan halaman untuk melakukan proses peramalan. Tampilan tersebut dibuat menggunakan aplikasi *scene builder* yaitu gluon, tampilan ini tersimpan kedalam *file* javaFXML. Hasil implementasi perangkat lunak adalah sebagai berikut:

#### a. Halaman *input* klasifikasi

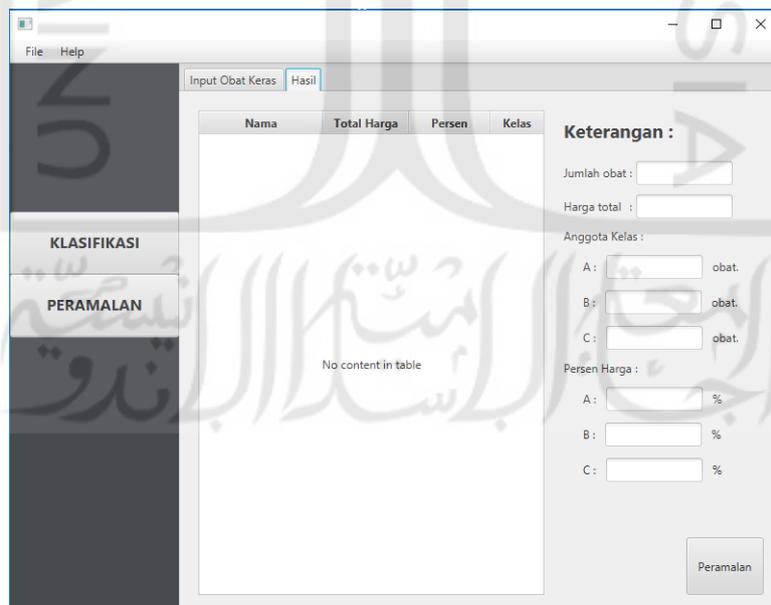
Halaman *input* klasifikasi merupakan halaman yang mengharuskan *user* untuk memasukkan data stok obat keras yang ada. Terkait *input* data stok obat, *user* bisa memasukkan data stok langsung dari *file* bertipe “.xls” atau secara manual memasukkan data satu persatu. Pada halaman ini *user* bisa menyimpan data stok obat keras yang sudah di-*input* kedalam database. Fitur ini berguna apabila proses klasifikasi yang akan dijalankan harus ditunda karena alasan tertentu. Jika *user* sudah menyimpan data stok obat, maka pada saat ingin melanjutkan proses klasifikasi kembali, *user* hanya perlu me-load kembali data stok itu. Terdapat tombol reset yang berfungsi untuk menghapus data stok yang sudah disimpan. Proses klasifikasi akan berjalan saat *user* menekan tombol start. Tampilan halaman dapat dilihat pada Gambar 3.27 berikut:



Gambar 3.27 Halaman *input* klasifikasi.

b. Halaman hasil klasifikasi

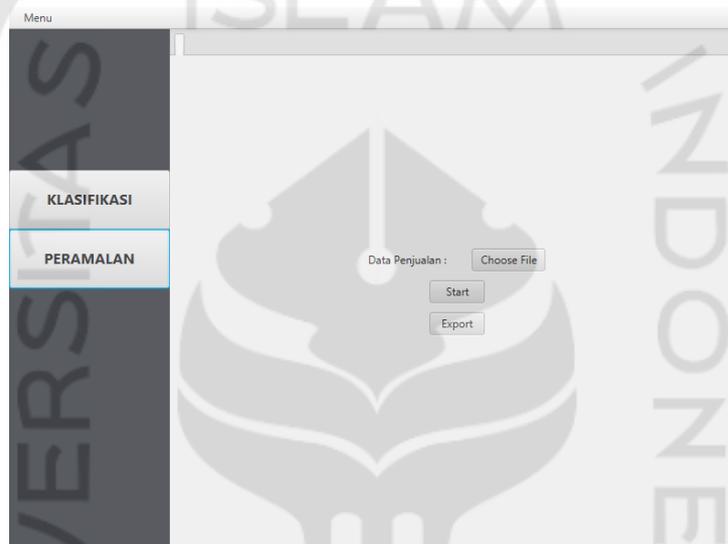
Halaman hasil klasifikasi merupakan halaman yang berfungsi untuk menampilkan hasil yang didapat dari proses klasifikasi. Halaman ini akan menampilkan data obat yang sudah digolongkan kedalam masing-masing kelas ABC, jumlah obat yang ada, jumlah harga total keseluruhan obat, berapa banyak anggota untuk setiap kelas ABC dan berapa persentase harga dari setiap kelas ABC terhadap harga total keseluruhan obat. Tampilan dari halaman dapat dilihat pada Gambar 3.28 berikut:



Gambar 3.28 Halaman hasil klasifikasi.

c. Halaman peramalan

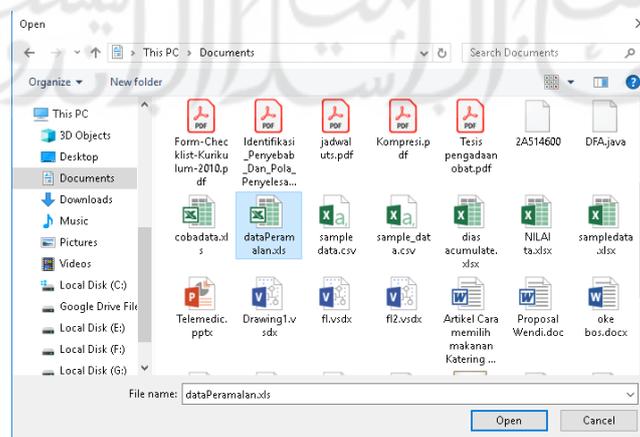
Halaman peramalan merupakan halaman yang berfungsi untuk meminta *input* berupa data penjualan bulanan obat keras, melakukan proses peramalan, sekaligus menampilkan hasil dari peramalan. Data hasil peramalan berisi nama, data penjualan 6 bulan terakhir, hasil peramalan untuk 2 bulan terakhir sekaligus hasil peramalan untuk bulan depan, MAD dan MAPE dari setiap obat keras yang tergolong kedalam kelas kategori A. pada halaman ini juga terdapat fitur untuk export hasil peramalan kedalam *file* bertipe “.xls”. Tampilan halaman dapat dilihat pada Gambar 3.29 berikut:



Gambar 3.29 Halaman peramalan.

d. Halaman Pilih *File*

Halaman pilih *file* akan ditampilkan ketika *user* menekan tombol *Choose File* yang ada pada halaman *input* klasifikasi dan halaman peramalan. Halaman ini berfungsi untuk memilih *file external* yang akan di-*input* kedalam sistem. Tampilan halaman dapat dilihat pada Gambar 3.30 berikut:



Gambar 3.30 Halaman pilih *file*.

### 3.6 Pengujian

Pengujian sistem merupakan tahap yang harus dilakukan untuk memastikan apakah perangkat lunak yang dibangun sudah sesuai dengan apa yang telah direncanakan sebelumnya atau tidak. Pengujian yang akan dilakukan terhadap perangkat lunak yang telah dibangun menggunakan *black box testing*. Hasil pengujian bisa dilihat pada Tabel 3.5 berikut:

Tabel 3.5 Pengujian sistem

No	Fungsi/Proses	Aktivitas	Hasil	Kesimpulan
1	Choose <i>File</i>	Memilih <i>file</i> external.	Mengambil data dari <i>file</i> yang dipilih.	Berhasil.
4	Load	Memanggil data yang telah disimpan.	Memanggil data dari database dan menampilkannya pada table view.	Berhasil.
5	Reset	Mengulang proses klasifikasi.	Mengulang proses klasifikasi dari pengambilan data yang akan diklasifikasikan	Berhasil.
6	Table view – masukan klasifikasi	Menampilkan data pada tabel.	Menampilkan data stok obat keras yang telah dimasukkan.	Berhasil.
7	Start Klasifikasi	Membaca data masukan.	Membaca dan mengambil nilai dari data stok obat yang telah dimasukkan dari <i>file excel</i> maupun <i>database</i> .	Berhasil.
		Menjalankan proses klasifikasi.	Melakukan proses klasifikasi terhadap nilai yang sudah diambil dari data masukan.	Berhasil.
8	Table view – hasil klasifikasi	Menampilkan data pada tabel.	Menampilkan data hasil klasifikasi.	Berhasil.
9	merangkum hasil klasifikasi	Menjumlahkan banyaknya obat yang dimasukkan.	Menampilkan jumlah obat.	Berhasil.
		Menjumlahkan harga dari keseluruhan obat.	Menampilkan harga keseluruhan obat.	Berhasil.

		Menghitung anggota untuk masing-masing kelas ABC.	Menampilkan banyaknya anggota untuk setiap kelas ABC	Berhasil.
		Menjumlahkan persentase harga obat untuk masing-masing kelas ABC.	Menampilkan total persentase harga obat untuk setiap kelas ABC	Berhasil.
10	Switch pane	Beralih halaman.	Menampilkan tampilan halaman yang dipilih.	Berhasil.
11	Start Peramalan	Membaca data obat kelas kategori A.	Mengambil data nama obat yang termasuk kelas kategori A.	Berhasil.
		Membaca data penjualan bulanan.	Mengambil nilai dari data penjualan bulanan yang telah dimasukkan.	Berhasil.
		Filtrasi obat.	Mengambil data penjualan yang termasuk kedalam kelas kategori A yang selalu terjual.	Berhasil.
		Menjalankan proses peramalan	Melakukan perhitungan peramalan terhadap data penjualan obat yang termasuk kelas kategori A.	Berhasil.
12	Table view – peramalan	Menampilkan data pada tabel.	Menampilkan data hasil peramalan kedalam tabel.	Berhasil
13	Export	Output data.	Membuat <i>file</i> baru bertipe .xls yang berisi data hasil peramalan.	Berhasil

Berdasarkan hasil pengujian perangkat lunak pada Tabel 3.5 di atas, maka dapat disimpulkan bahwa proses dan fitur yang ada di dalamnya berjalan sesuai alur dan hasil dari setiap proses yang ada belum sesuai dengan rancangan.