

## BAB V

### SIMULASI PERHITUNGAN PEMBEBANAN EKONOMIS PLTGU DI PT. PJB UP. GRESIK

#### 5.1 Perancangan *Software*

Dalam melakukan studi perhitungan pembebanan ekonomis pada Pembangkit Listrik Tenaga Gas dan Uap (PLTGU) di PT. PJB UP. Gresik dirancnglah suatu *software* sistem manajemen energi dengan menggunakan *software* JAVA FX 2 dengan pertimbangan :

1. Sederhana

Bahasa pemrograman Java menggunakan sintaks mirip dengan C++ namun sintaks pada Java telah banyak diperbaiki terutama menghilangkan penggunaan pointer yang rumit dan *multiple inheritance*. Java juga menggunakan *automatic memory allocation* dan *memory garbage collection* (Isak Rickyanto, 2005).

2. Berorientasi objek (*Object Oriented*)

Java menggunakan pemrograman berorientasi objek yang membuat program dapat dibuat secara modular dan dapat dipergunakan kembali. Pemrograman berorientasi objek memodelkan dunia nyata kedalam objek dan melakukan interaksi antar objek-objek tersebut.

3. Dapat didistribusi dengan mudah

Java dibuat untuk membuat aplikasi terdistribusi secara mudah dengan adanya *libraries networking* yang terintegrasi pada Java.

#### 4. Interpreter

Program Java dijalankan menggunakan interpreter yaitu *Java Virtual Machine* (JVM). Hal ini menyebabkan *source code* Java yang telah dikompilasi menjadi *Java bytecodes* dapat dijalankan pada platform yang berbeda-beda.

#### 5. Robust

Java mempunyai reliabilitas yang tinggi. Compiler pada Java mempunyai kemampuan mendeteksi error secara lebih teliti dibandingkan bahasa pemrograman lain. Java mempunyai *runtime-Exception handling* untuk membantu mengatasi error pada pemrograman.

#### 6. Aman

Sebagai bahasa pemrograman untuk aplikasi internet dan terdistribusi, Java memiliki beberapa mekanisme keamanan untuk menjaga aplikasi tidak digunakan untuk merusak sistem komputer yang menjalankan aplikasi tersebut.

#### 7. Architecture Neutral

Program Java merupakan *platform independent*. Program cukup mempunyai satu buah versi yang dapat dijalankan pada platform yang berbeda dengan *Java Virtual Machine*.

#### 8. Portabel

Source code maupun program Java dapat dengan mudah dibawa ke platform yang berbeda-beda tanpa harus dikompilasi ulang.

## 9. Performance

Performance pada Java sering dikatakan kurang tinggi. Namun performance Java dapat ditingkatkan menggunakan kompilasi Java lain seperti buatan Inprise, Microsoft ataupun Symantec yang menggunakan *Just In Time Compilers (JIT)*.

## 10. Multithreaded

Java mempunyai kemampuan untuk membuat suatu program yang dapat melakukan beberapa pekerjaan secara sekaligus dan simultan.

## 11. Dinamis

Java didesain untuk dapat dijalankan pada lingkungan yang dinamis. Perubahan pada suatu *class* dengan menambahkan properties ataupun method dapat dilakukan tanpa mengganggu program yang menggunakan *class* tersebut.

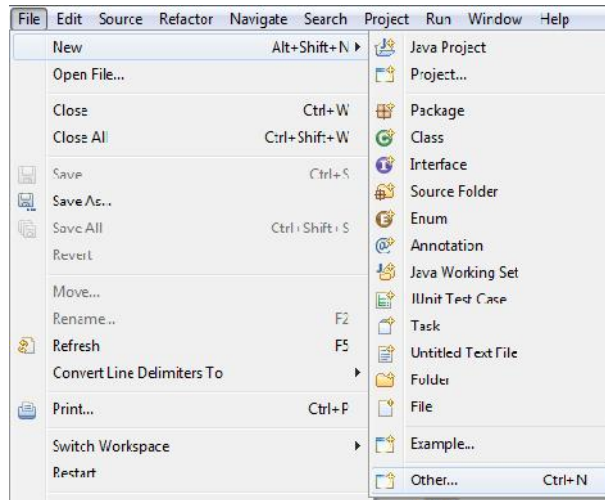
*Software* JAVA FX 2 digunakan untuk merancang optimasi yang dibutuhkan, yaitu model *Unit Commitment* dan *Economic Dispatch*.

### **5.1.1 Pemrograman dengan menggunakan JavaFX 2**

#### **5.1.1.1 Membuat project melalui Java FX**

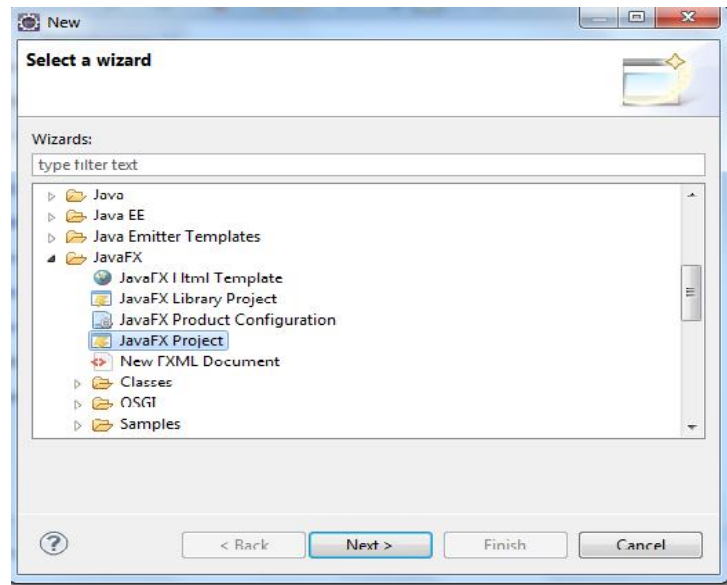
Adapun langkah-langkah membuat project sebagai berikut :

1. Klik file → new → other



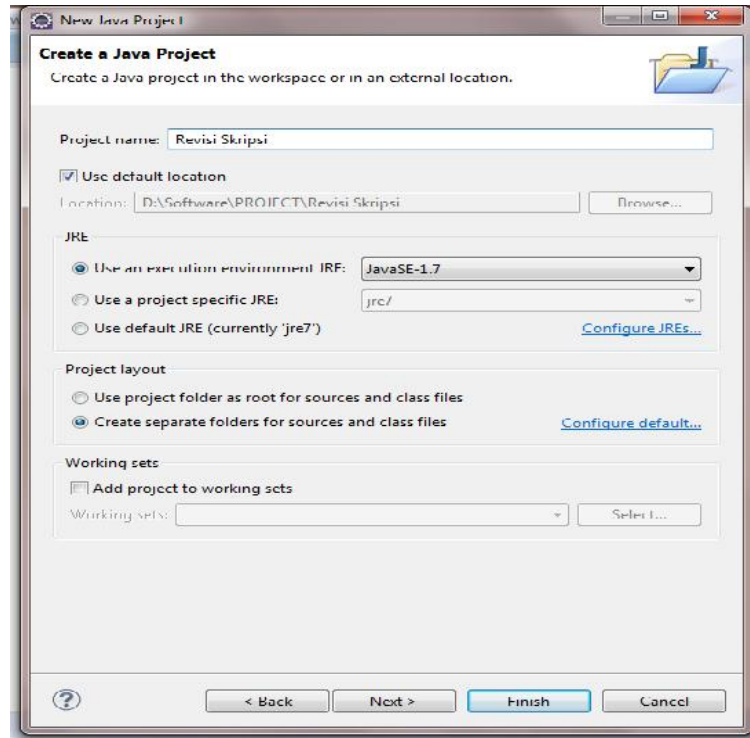
**Gambar 5.1 Langkah awal membuat project dengan JavaFX 2**

2. Maka akan muncul kotak dialog box, pilih JavaFX → JavaFX Project



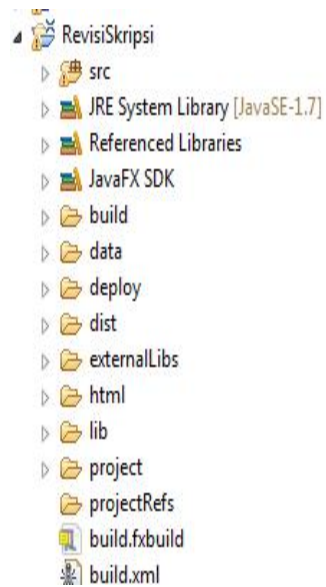
**Gambar 5.2 Kotak dialog box**

3. Ketik nama project, setelah selesai, klik finish.



**Gambar 5.3 New Java Project**

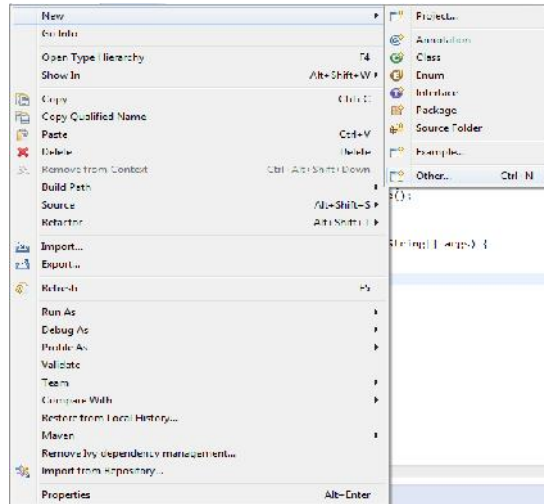
Maka project anda akan muncul gambar berikut :



**Gambar 5.4 folder awal project**

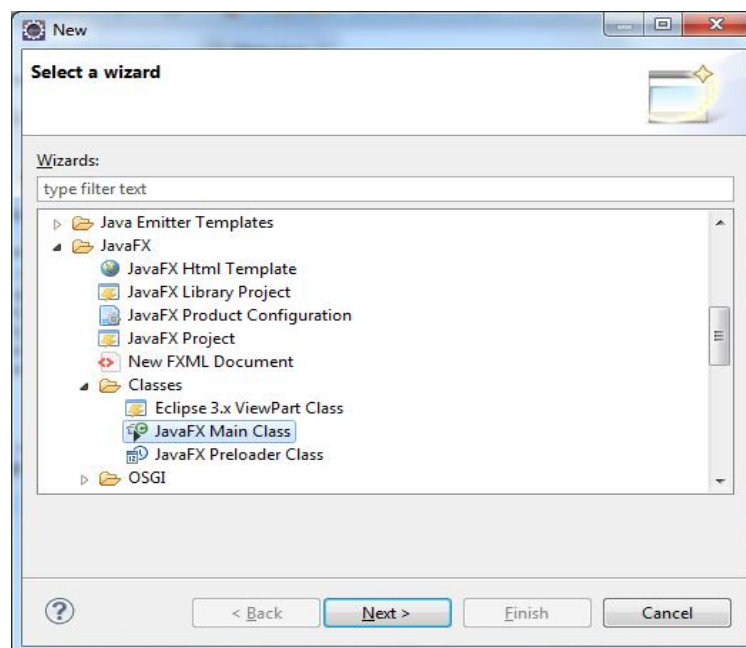
### 5.1.5.2 Membuat aplikasi dengan JavaFX 2

1. Arahkan kursor anda ke folder src → klik kanan dan anda akan mendapatkan tampilan seperti berikut :



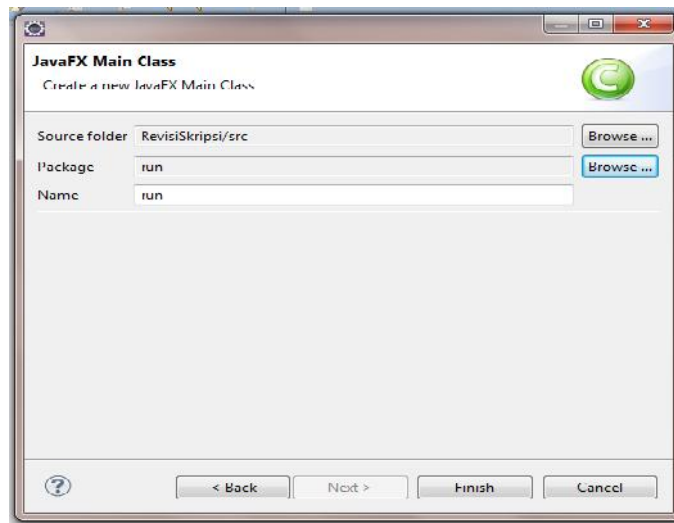
**Gambar 5.5 Langkah awal membuat aplikasi dengan Java FX 2**

2. Pilih other. Setelah itu, anda akan disuguhkan dengan form seperti pada saat kita membuat project untuk pertama kali.



**Gambar 5.6 New Java Application**

3. Carilah folder JavaFX → Classes, kemudian pilih JavaFX Main Class, klik next.



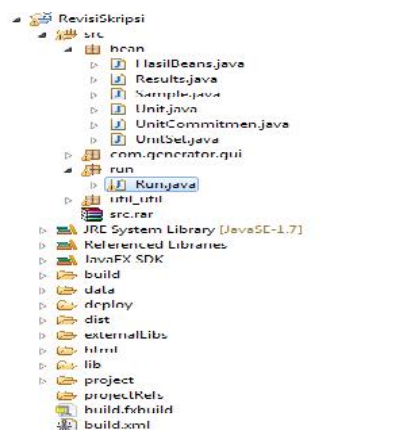
**Gambar 5.7 JavaFX Main Class**

Dalam pembuatan program simulasi perhitungan pembebanan ekonomis ini terdapat 4 package dan masing-masing package terdiri dari beberapa class, yaitu sebagai berikut :

1. Package “bean” : Class yang ada didalam package ini berfungsi sebagai kerangka objek. Lebih tepatnya digunakan untuk mendefinisikan struktur array yang akan digunakan dalam proses pembuatan aplikasi ini.
2. Package “com.generator.gui” : Di package ini berisi class yang berkaitan dengan pembuatan gui, logic gui, file-file gambar yang dibutuhkan gui.
3. Package “run” : package untuk class *main* proses pengolahan data. Saat tombol “Execute” di eksekusi, class yang ada didalam package inilah yang

digunakan untuk memproses data untuk diperoleh hasil yang nantinya ditampilkan dalam GUI (*Graphical User Interface* / Antar Muka).

4. Package “util\_util” : Package ini berisi class yang digunakan untuk bantuan dalam proses, tapi bukan proses untuk mengolah datanya, melainkan fungsi fungsi tambahan seperti : load data dari file, menyimpan data ke file, membentuk data json, dan lain lain.
4. Misalkan kita akan akan membuat Class dengan nama run pada package run, maka rubahlah package dengan cara mengklik tombol browse dan isi nama class dengan nama Run. Setelah selesai, klik finish.
5. Setelah selesai, maka project anda akan berubah seperti pada gambar berikut :



**Gambar 5.8 folder akhir project**

6. Menyusun program yang akan dirancang.



#### **A. Program untuk *Unit Commitment* dengan menerapkan metode *Priority List***

Pemodelan *Unit Commitment* bertujuan untuk mendapatkan status *on-off* dari setiap unit generator. Model ini menggunakan metode *Priority List*. Selanjutnya keluaran dari model ini langsung menjadi input untuk *Economic Dispatch*. *Flow chart* untuk model *Unit Commitment* dapat dilihat lebih jelas pada gambar 3.3, sedangkan listing programnya dapat dilihat dalam lampiran 1.

#### **B. Program untuk *Economic Dispatch* dengan menerapkan metode *Lagrange***

Pemodelan selanjutnya adalah pemodelan *Economic Dispatch* menggunakan metode *Lagrange*, seperti yang diuraikan dalam bab sebelumnya. Tujuan dari model ini adalah mengoptimumkan power yang di *dispatch* dari setiap blok pembangkitan. Dengan mensimulasi besaran  $P_{load}$  maka power yang *dispatch* dapat berubah-ubah untuk masing-masing generator. *Flow chart* dari *software* untuk model *Economic Dispatch* dapat dilihat dalam gambar 3.4. Untuk potongan listing program dapat dilihat dalam lampiran 2.


### **C. Program untuk *Unit Commitment* dengan menerapkan metode *Dynamic Programming***

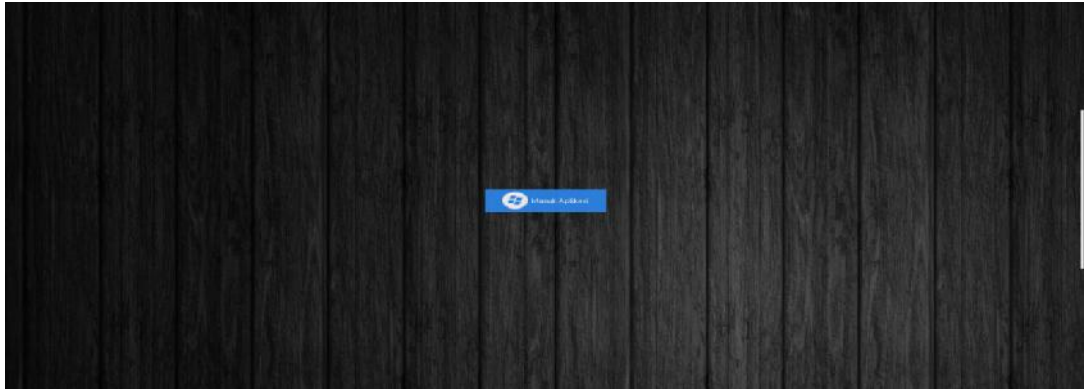
Pemodelan yang terakhir adalah pemodelan *Unit Commitment* dengan menerapkan metode *Dynamic Programming*. Tujuan pemodelan ini dilakukan untuk mendapatkan nilai beban yang ditanggung oleh masing-masing unit generator tiap blok dan biaya pembangkitan untuk melayani sejumlah beban dari semua bentuk konfigurasi unit pembangkit gas dan unit pembangkit uap didalam blok pembangkit. *Flow chart* dari *software* untuk model *Unit Commitment* dengan menerapkan metode *Dynamic Programming* dapat dilihat dalam gambar 3.5. Untuk potongan listing program dapat dilihat dalam lampiran 3.

### **D. Program untuk menampilkan hasil optimasi kedalam *User Interface***

Program ini berfungsi untuk menampilkan nilai dari parameter hasil program *Unit Commitment* dan *Economic Dispatch*. Listing untuk program ini dapat dilihat dalam lampiran 4. Setelah semua program telah dirancang, maka aplikasi siap untuk dioperasikan.

Adapun langkah-langkah mengoperasikan aplikasi yang telah dibuat sebagai berikut :

- a. Klik tombol MainRunner  pada JavaFX
- b. Maka akan muncul halaman awal aplikasi seperti ditunjukkan dalam gambar 5.9.



**Gambar 5.9 Tampilan halaman awal aplikasi**

- c. Klik tombol masuk aplikasi, maka akan muncul tampilan awal untuk input beberapa data yang diperlukan, seperti persamaan *fuel rate function* tiap blok pembangkit yang berbentuk polynomial ordo 2 yaitu  $F_i P_i = a + bx + cx^2$ , nilai kapasitas maksimum dan minimum dari masing-masing blok pembangkit, *fuel cost*, dan total *load demand* yang diinginkan.

**Gambar 5.10 Tampilan untuk input data**

Logic yang dieksekusi pertama kali dalam program ini adalah me-load data sample dengan format json dari file \*.txt, jika dalam file tersebut terdapat data sample maka secara otomatis program ini menampilkannya dalam tabel.

Berikut logiknya :

```
public void initialize(URL location, ResourceBundle resources) {
    /*
```

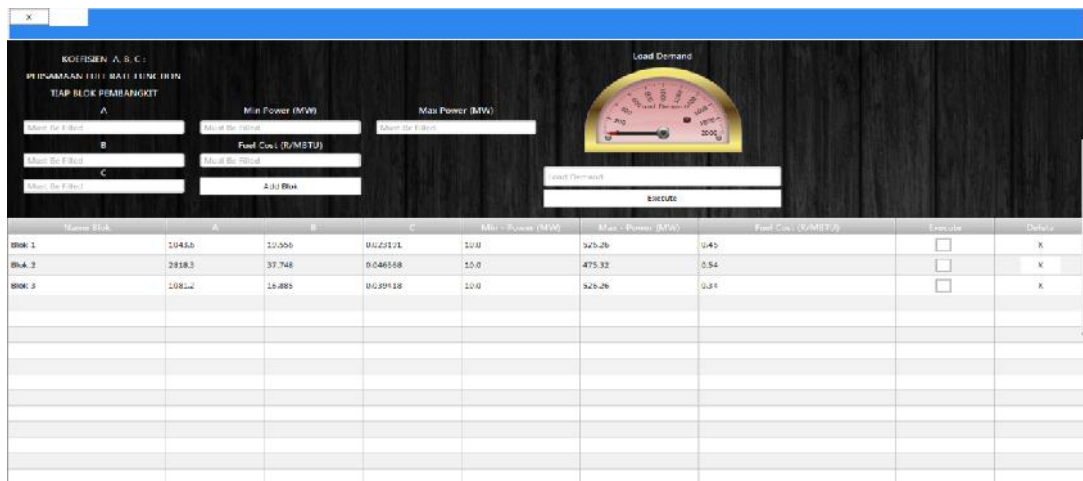
```

*****
*/
Utility.readJson();
List_Unit = Run.DATA_SAMPLES;
Table_Generator.setItems(List_Unit);
}

```

❖ Method `Utility.readJson()` adalah method yang akan meload data Json dari file txt, lalu di tampilkan ke dalam Table View.

d. Inputkan data persamaan *fuel rate function*, kapasitas maksimum dan minimum blok pembangkit dan *fuel cost*. Lalu tekan tombol Add Blok. Maka data akan muncul dalam kolom yang telah tersedia, seperti ditunjukkan dalam gambar 5.11.



**Gambar 5.11 tampilan data inputan**

Tombol “Add Blok” digunakan untuk menambahkan data sample. Berikut logiknya :

```

public void addBlok() {
    if (cekInput()) {
        try {

            Unit unit = new Unit();
            unit.name = "Generator " + (List_Unit.size() + 1);

```

```

        unit.a = new
Double(Double.valueOf(I_ValueA.getText()));
        unit.b = new
Double(Double.valueOf(I_ValueB.getText()));
        unit.c = new
Double(Double.valueOf(I_ValueC.getText()));
        unit.min = new
Double(Double.valueOf(I_MinPower.getText()));
        unit.max = new
Double(Double.valueOf(I_MaxPower.getText()));
        unit.fC = new
Double(Double.valueOf(I_ProdCost.getText()));
        Run.Data.add(unit);
        List_Unit.add(unit);
        clearBox();

    } catch (Exception e) {
        Msg.setText("Field Harus berisi Angka !");
    }
}
}

```

Tombol “X” pada masing – masing sample : berfungsi untuk menghapus data sample. Berikut logiknya :

```

Callback<TableColumn<Unit, Boolean>, TableCell<Unit, Boolean>> delCell;
delCell = new Callback<TableColumn<Unit, Boolean>,
TableCell<Unit, Boolean>>() {
    @Override
    public TableCell<Unit, Boolean>
call(TableColumn<Unit, Boolean> arg0) {
        return new ButtonCell<>();
    }
};

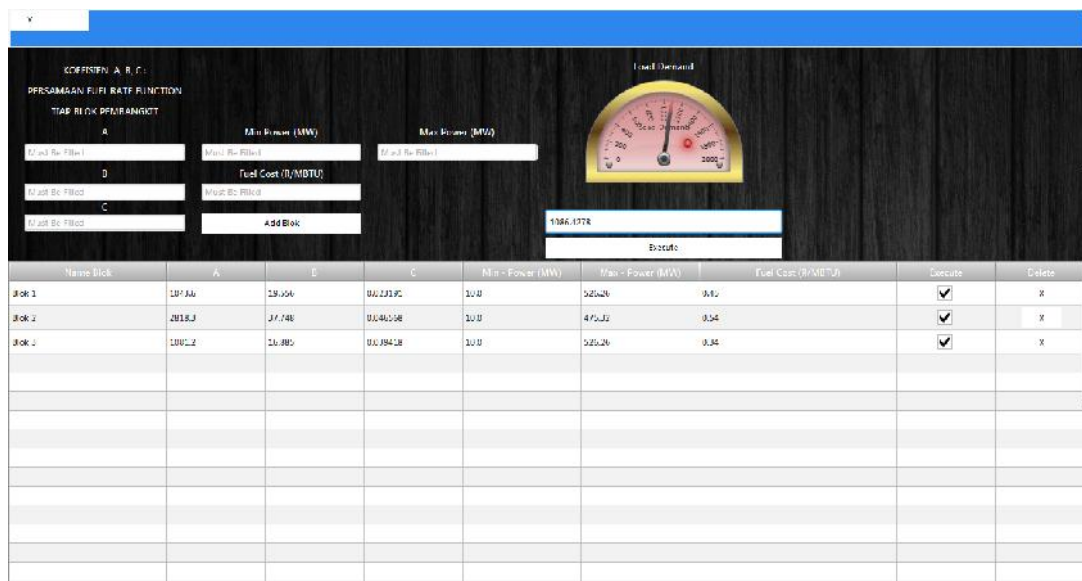
```

```

public ButtonCell() {
    this.button = new Button("X");
    this.button.setAlignment(Pos.CENTER);
    this.button.setOnAction(new EventHandler<ActionEvent>()
{
    @Override
    public void handle(ActionEvent arg0) {
        Run.DATA_SAMPLES.remove(getIndex());
    }
}
);
}

```

e. Input *load demand* yang diinginkan dan beri tanda centang pada kolom *excute* untuk data yang akan dieksekusi. Misal *load demand* sebesar 1086,4278 MW, maka tampilan akan berubah menjadi seperti diperlihatkan dalam gambar 5.12.



**Gambar 5.12 Tampilan load demand**

f. Klik tombol *excute*, maka program akan mengeksekusi data yang telah diinputkan.

Tombol *Execute* : Tombol ini mengesekusi logic pemrosesan data. Berikut adalah main logic dari program ini.

```

public void execute() {
    Utility.createDataJson(Run.DATA_SAMPLES);
    min = new Double(0);
    max = new Double(0);
    Run.refreshData();
    addDataExkecut();
    if (totalExec != 3) {
        Dialogs.showErrorDialog(new Stage(),
            "Check 3 Blok ");
        totalExec = 0;
    } else {
        if (I_LoadDemand.getText().isEmpty()) {
            Dialogs.showErrorDialog(new Stage(),
                "Load Demand Tidak boleh Kosong!");
        } else if
        (Double.parseDouble(I_LoadDemand.getText()) < min) {
            Dialogs.showErrorDialog(new Stage(),
                "Load Demand Tidak boleh kurang dari " + min + "!");
        } else if
        (Double.parseDouble(I_LoadDemand.getText()) > max) {
            Dialogs.showErrorDialog(new Stage(),
                "Angka Maksimal Telah terlampaui batas maksimal " + max + "!");
        }
        else {
            notifikasi = Notifier.INSTANCE;
            notifikasi.notifyInfo("Loading", "Mohon Tunggu...");
            notifikasi.setPopupLifetime(Duration.millis(800));
            Run.Go();
            notifikasi.getTimeline().setOnFinished(
                new EventHandler<ActionEvent>()
            {
                @Override
                public void handle(ActionEvent arg0) {
                    notifikasi.getStage().close();
                    try {
                        Frame.setSceneContent("HasilGui.fxml");
                        //
                        Frame.setSceneContent("Hasil.fxml");
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                }
            });
        }
    }
}

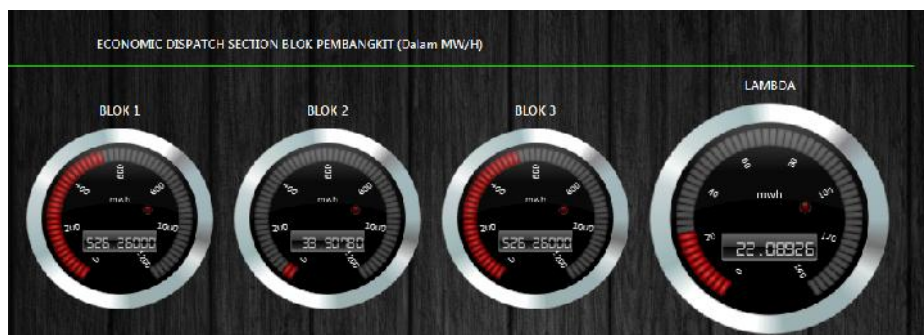
```

Berikut adalah hasil eksekusi program studi perhitungan pembebanan ekonomis dengan *load* sebesar 1086,4278 MW.



**Gambar 5.13 Unit Commitment section blok pembangkit**

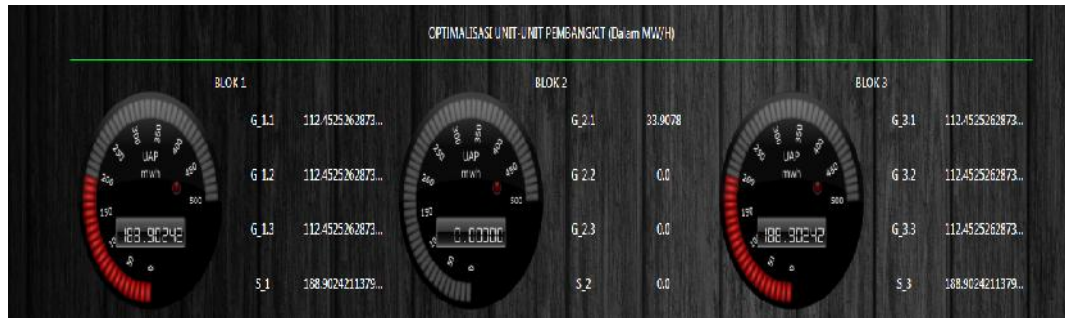
Gambar 5.13 diatas memperlihatkan hasil optimasi *unit commitment* dengan *load* sebesar 1086,4278 MW. Dengan total *load demand* sebesar 1086,4278 MW , hasil simulasi menunjukkan *unit commitment* dari ketiga blok tersebut dalam keadaan operasional (ON), dengan ditunjukkannya menyalanya warna hijau dari masing-masing blok pembangkit. Dan apabila blok pembangkit tersebut dalam keadaan *stand-by* maka blok tersebut tidak memberikan indikasi warna hijau.



**Gambar 5.14 Economic Dispatch Section Blok Pembangkit**

Gambar 5.14 diatas memperlihatkan hasil optimasi *economic dispatch* tiap blok pembangkit. Dengan beban sebesar 1086.4278 MW, *economic dispatch* memutuskan blok 1 dan blok 3 beroperasi dengan kapasitas maksimumnya yaitu sebesar 526,26 MW, sedangkan blok 2 beroperasi dengan daya yang dibangkitkan sebesar 33,9078 MW.





**Gambar 5.15 Optimalisasi Unit-unit pembangkit didalam blok**

Gambar 5.15 diatas memperlihatkan optimalisasi unit-unit pembangkit didalam blok. Dari hasil *economic dispatch* blok 1 dan blok 3 beroperasi dalam kapasitas maksimumnya, sehingga blok 1 dan blok 3 dalam pengoptimalannya dengan *unit commitment* dengan metode *dynamic programming* masuk dalam *state* 4, dimana blok pembangkitan tersebut bekerja dengan konfigurasi 3.3.1 (3 unit pembangkit gas, 3 HRSG, 1 pembangkit uap) dalam keadaan operasional (ON). Sedangkan pada blok 2, beban yang ditanggung hanya 33,9078 MW. Dengan beban yang dibangkitkan sangat kecil optimalisasi unit-unit pembangkit masuk dalam *state* 1 (hanya ada 1 generator yang operasi/ON), karena hanya 1 unit generator yang ON dan daya yang dibangkitka juga masih dibawah 50.89 MW (batas minimum *state* 1) sehingga gas buang yang dihasilkan oleh satu generator tersebut tidak mampu menghasilkan uap, sehingga pembangkit uap dalam keadaan *stand-by* (OFF).



**Gambar 5.16** *Production cost* dari masing-masing unit generator

Gambar 5.16 diatas memperlihatkan *production cost* dari masing-masing unit-unit pembangkit didalam blok. Dimana *production cost* dari masing-masing unit pembangkit tersebut dijadikan sebagai pembandingan keekonomisan dari segi pemakaian bahan bakar.

- g. Jika ingin menginputkan *load demand* atau persamaan yang lain dengan berbagai variasi, maka tekan tombol kembali disebelah kiri pojok atas aplikasi. Dimana tombol “Kembali” berfungsi untuk kembali ke *screen* awal.



**Gambar 5.17** Tampilan kembali ke *screen* awal

Berikut logiknya :

```

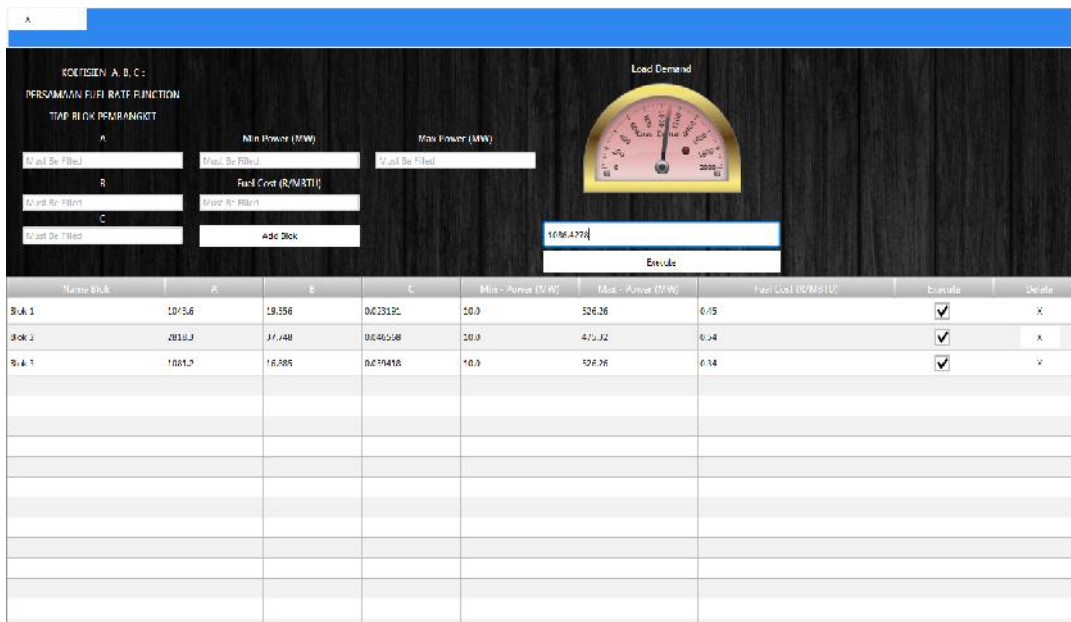
public void backAct() {
    try {
        new InputControl();

        Frame.setSceneContent("InputSection.fxml");
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

- h. Ulangi semua prosedur mulai dari prosedur (d).
- i. Selesai.

Untuk mendapatkan ilustrasi mengenai tampilan *software* JavaFX 2 secara keseluruhan dapat dilihat pada gambar 5.18 samapi dengan gambar 5.20.



**Gambar 5.18 Tampilan *load demand* dengan beban 1086,4278 MW dengan *software* JavaFX 2**

Gambar 5.18 diatas memperlihatkan tampilan *load demand* dengan beban sebesar 1086,4278 MW dengan *software* JavaFX 2.



**Gambar 5.19** Tampilan *Unit Commitment*, hasil perhitungan pembebanan ekonomis blok pembangkitan dan pembagian beban tiap unit generator dengan *software JavaFX 2*

Gambar 5.19 diatas menampilkan *Unit Commitment* blok pembangkitan, hasil perhitungan pembebanan ekonomis tiap blok pembangkitan serta pembagian beban tiap unit generator yang telah dirancang dengan *software JavaFX 2*.



**Gambar 5.20** Tampilan hasil perhitungan *cost production* dengan *software JavaFX 2*

Gambar 5.20 diatas menunjukkan hasil perhitungan *cost production* dengan *software* JavaFX 2 dari masing-masing unit generator yang ada dalam tiap-tiap blok pembangkitan.

### 5.1 Pengujian *software*

Tujuan dari pengujian ini adalah untuk mengetahui apakah hasil optimasi dengan model *Unit Commitment* dengan *Economic Dispatch* yang dikembangkan dengan *software* JAVA FX 2 akan didapatkan hasil yang diharapkan. Untuk keperluan ini, maka program yang telah dibuat divalidasi dengan menggunakan POWERGEN dan Ms. Excel. Untuk perbandingan tampilan kedua program, dapat dilihat dalam tabel 5.1 sampai dengan tabel 5.7.

**Tabel 5.1 Hasil Validasi JAVA FX dengan POWERGEN Untuk Pembagian Beban Blok**

LOAD	JAVA FX				POWERGEN				Error
	P Blok 1	P Blok 2	P Blok 3	LAMBDA	P Blok 1	P Blok 2	P Blok 3	LAMBDA	
1086,427789	526,26	33,91	526,26	22,08926	526,3	33,9	526,3	22,0896	0,001 %
1034,674992	508,46	0	526,15	19,84482	508,5	0	526,2	19,8447	0,001 %
1035,612709	509,35	0	526,26	19,86317	509,3	0	526,3	19,863	0,001 %
998,948917	448,6	0	510,35	19,4204	488,6	0	510,4	19,4205	0,001 %
1033,405701	507,79	0	525,62	19,82974	507,8	0	525,6	19,8298	0,001 %
1041,9024	515,64	0	526,26	19,99736	515,6	0	526,3	19,9974	0,001 %
1035,718201	509,46	0	526,26	19,86542	509,5	0	526,3	19,8656	0,001 %
1039,848012	513,59	0	526,26	19,95353	513,6	0	526,3	19,9536	0,001 %
1060,818016	524,56	10	526,26	20,18758	524,6	10	526,3	20,1877	0,001 %
1051,874596	525,61	0	526,26	20,21	525,6	0	526,3	20,21	0,001 %
1015,682701	497,92	0	517,76	19,61919	497,9	0	517,8	19,6192	0,001 %
1066,924785	526,26	14,4	526,26	21,10839	526,3	14,4	526,3	21,1083	0,001 %
1037,674286	511,41	0	526,26	19,90715	511,4	0	526,3	19,9072	0,001 %
1101,8161	526,26	49,3	526,26	22,86319	526,3	49,3	526,3	22,863	0,001 %
1112,989498	526,26	60,47	526,26	23,42514	526,3	60,5	526,3	23,4249	0,001 %
1145,055614	526,26	92,54	526,26	25,03785	526,3	92,5	526,3	25,0382	0,001 %

1120,44799	526,26	67,93	526,26	23,80025	526,3	67,9	526,3	23,7999	0,001 %
1250,142494	526,26	197,62	526,26	30,32304	526,3	197,6	526,3	30,3232	0,001 %
1368,893608	526,26	316,37	526,26	36,29544	526,3	316,4	526,3	36,2952	0,001 %
1374,24971	526,26	321,73	526,26	36,56481	526,3	321,7	526,3	36,5651	0,001 %
1385,284211	526,26	332,76	526,26	37,11978	526,3	332,8	526,3	37,12	0,001 %
1290,392502	526,26	237,87	526,26	33,34735	526,3	237,9	526,3	32,3478	0,001 %
1191,838505	526,26	139,32	526,26	27,39073	526,3	139,3	526,3	27,391	0,001 %
1117,33268	526,26	64,81	526,26	23,64357	526,3	64,8	526,3	23,6433	0,001 %

Tabel 5.1 diatas memperlihatkan hasil validasi JAVA FX 2 dengan POWERGEN untuk pembagian beban tiap blok dengan berbagai variasi beban tiap jam. Dimana dari hasil validasi tersebut diperoleh bahwa *error* kurang dari 0.001%. Hal ini menunjukkan *software* yang telah di rancang telah berjalan baik dan sesuai harapan.

Untuk menjelaskan tampilan saat validasi, dapat diambil contoh pada pengujian beban pada pukul 01.00 sebesar 1086,4278 MW pada gambar 5.1



```

E:\KU9C35-1\AST1-1\POWERGENDC.EXE
lambda = 22.7566 totalgen = 1099.7
lambda = 21.5604 totalgen = 1075.9
lambda = 22.1585 totalgen = 1087.8
lambda = 21.8594 totalgen = 1081.9
lambda = 22.0090 totalgen = 1084.8
lambda = 22.0837 totalgen = 1086.3
lambda = 22.1211 totalgen = 1087.1
lambda = 22.1024 totalgen = 1086.7
lambda = 22.0931 totalgen = 1086.5
lambda = 22.0884 totalgen = 1086.4
lambda = 22.0908 totalgen = 1086.5
lambda = 22.0896 totalgen = 1086.4

generator  output  limit  inc cost  penalty fact  operating cost
          mw          $/mwhr          $/hr
-----
UNIT1    526.3    max    20.2239    1.0000    8168.62
UNIT2     33.9    max    22.0896    1.0000    2242.11
UNIT3    526.3    max    19.8469    1.0000    7100.53
-----
totals    1086.4
lambda = 22.0896                                17511.26

do you want to run edc again? (Y or N) :

```

**Gambar 5.21 Validasi program JavaFX 2 dengan POWERGEN untuk pembagian beban tiap blok dengan beban 1086,4278 MW**

**Tabel 5.2 Hasil Validasi JavaFX 2 dengan Ms. Excel Untuk Pembagian Beban Tiap Unit-unit Generator Blok I**

Jam	JavaFX 2					Ms. Excel				
	Beban (MW)					Beban (MW)				
	G11	G12	G13	S1	Total	G11	G12	G13	S1	Total
01.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
02.00	108.45	108.45	108.45	183.106	508.456	108.45	108.45	108.45	183.106	508.456
03.00	108.651	108.651	108.651	183.397	509.35	108.651	108.651	108.651	183.397	509.35
04.00	103.986	103.986	103.986	176.64	488.598	103.986	103.986	103.986	176.64	488.598
05.00	108.299	108.299	108.299	182.88	507.777	108.299	108.299	108.299	182.88	507.777
06.00	110.065	110.065	110.065	185.445	515.64	110.065	110.065	110.065	185.445	515.64
07.00	108.675	108.675	108.675	183.43	509.455	108.675	108.675	108.675	183.43	509.455
08.00	109.603	109.603	109.603	184.776	513.585	109.603	109.603	109.603	184.776	513.585
09.00	112.069	112.069	112.069	188.348	524.555	112.069	112.069	112.069	188.348	524.555
10.00	112.307	112.307	112.307	188.69	525.611	112.307	112.307	112.307	188.69	525.611
11.00	106.081	106.081	106.081	179.674	497.917	106.081	106.081	106.081	179.674	497.917
12.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
13.00	109.115	109.115	109.115	184.068	511.413	109.115	109.115	109.115	184.068	511.413
14.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26

15.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
16.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
17.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
18.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
19.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
20.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
21.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
22.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
23.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
24.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26

**Tabel 5.3 Hasil Validasi JavaFX 2 dengan Ms. Excel Untuk Pembagian Beban Tiap Unit-unit Generator Blok II**

Jam	JavaFX 2					Ms. Excel				
	Beban (MW)					Beban (MW)				
	G21	G22	G23	S1	Total	G21	G22	G23	S1	Total
01.00	33.9078	0	0	0	33.9078	33.9078	0	0	0	33.9078
02.00	0	0	0	0	0	0	0	0	0	0
03.00	0	0	0	0	0	0	0	0	0	0
04.00	0	0	0	0	0	0	0	0	0	0
05.00	0	0	0	0	0	0	0	0	0	0
06.00	0	0	0	0	0	0	0	0	0	0
07.00	0	0	0	0	0	0	0	0	0	0
08.00	0	0	0	0	0	0	0	0	0	0
09.00	10	0	0	0	10	10	0	0	0	10
10.00	0	0	0	0	0	0	0	0	0	0
11.00	0	0	0	0	0	0	0	0	0	0
12.00	14.4	0	0	0	14.4	14.4	0	0	0	14.4
13.00	0	0	0	0	0	0	0	0	0	0
14.00	49.296	0	0	0	49.296	49.296	0	0	0	49.296
15.00	36.713	0	0	23.756	60.469	36.713	0	0	23.756	60.469
16.00	59.174	0	0	33.36	92.534	59.174	0	0	33.36	92.534
17.00	41.937	0	0	25.99	67.927	41.937	0	0	25.99	67.927
18.00	60.967	60.967	0	75.688	197.622	60.967	60.967	0	75.688	197.622
19.00	65.269	65.269	65.269	120.564	316.371	65.269	65.269	65.269	120.564	316.371
20.00	66.471	66.471	66.471	122.305	321.718	66.471	66.471	66.471	122.305	321.718
21.00	68.954	68.954	68.954	125.901	332.763	68.954	68.954	68.954	125.901	332.763



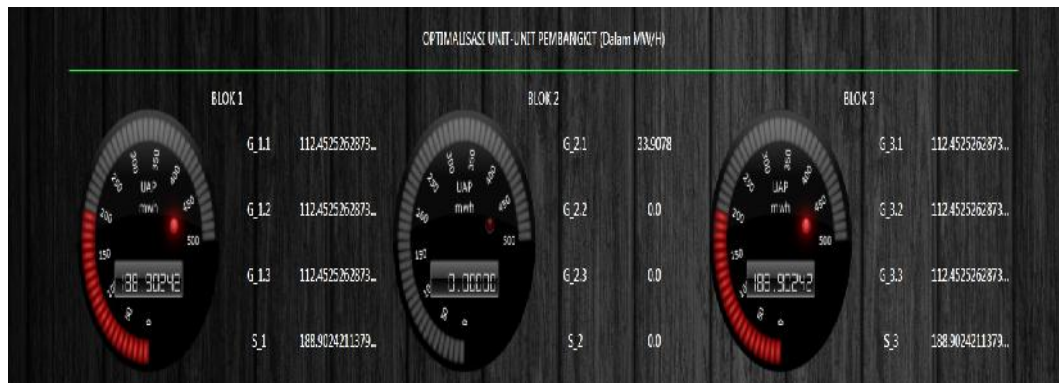
22.00	74.69	74.69	0		149.38	74.69	74.69	0		149.38
23.00	91.944	0	0	47.373	139.317	91.944	0	0	47.373	139.317
24.00	39.755	0	0	25.057	64.812	39.755	0	0	25.057	64.812

**Tabel 5.4 Hasil Validasi JavaFX 2 dengan Ms. Excel Untuk Pembagian Beban Tiap Unit-unit Generator Blok III**

Jam	JavaFX 2					Ms. Excel				
	Beban (MW)					Beban (MW)				
	G31	G32	G33	S3	Total	G31	G32	G33	S1	Total
01.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
02.00	112.428	112.428	112.428	188.86	526.144	112.428	112.428	112.428	188.86	526.144
03.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
04.00	108.875	108.875	108.875	183.721	510.346	108.875	108.875	108.875	183.721	510.346
05.00	112.308	112.308	112.308	188.693	525.617	112.308	112.308	112.308	188.693	525.617
06.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
07.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
08.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
09.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
10.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
11.00	110.54	110.54	110.54	186.136	517.756	110.54	110.54	110.54	186.136	517.756
12.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
13.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
14.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
15.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
16.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
17.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
18.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
19.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
20.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
21.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
22.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
23.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26
24.00	112.45	112.45	112.45	188.91	526.26	112.45	112.45	112.45	188.91	526.26

Tabel 5.2, tabel 5.3 dan tabel 5.4 diatas memperlihatkan hasil validasi JavaFX 2 dengan Ms. Excel untuk pembagian beban tiap unit-unit generator per blok. Dimana dari hasil validasi tersebut antara hasil perhitungan dengan JavaFX 2 dan Ms. Excel tidak terdapat *error* dalam perhitungannya.

Untuk menjelaskan tampilan pembagian beban tiap-tiap unit generator saat validasi, dapat diambil contoh pada pengujian beban pada pukul 01.00 sebesar 1086,4278 MW pada gambar 5.2.



Blok	Konfigurasi	Load Demand	P <sub>in</sub> = P <sub>out</sub>	P G1.1	P G1.2	P G1.3	S1
I	Non Konfigurasi	0	0	0	0	0	0
	1.1.1	0	8.037522605	1427806192	1000000000	0	0
	2.2.1	0	18.8061201	1466486388	1000000000	0	0
	3.3.1	0	25.03000365	1482788672	1000000000	112.4525263	112.4525263
II	Non Konfigurasi	33.91	0	0	0	0	0
	1.1.1	0	8.037522605	1427806192	1000000000	0	0
	2.2.1	0	18.8061201	1466486388	1000000000	0	0
	3.3.1	0	25.03000365	1482788672	1000000000	0	0
III	Non Konfigurasi	0	0	0	0	0	0
	1.1.1	0	8.037522605	1427806192	1000000000	0	0
	2.2.1	0	18.8061201	1466486388	1000000000	0	0
	3.3.1	526.26	25.03000365	1482788672	1000000000	112.4525263	112.4525263

**Gambar 5.22 Validasi program JavaFX 2 dengan Ms. Excel untuk pembagian beban tiap-tiap unit generator dengan beban 1086,4278 MW**

**Tabel 5.5 Validasi JavaFX 2 dengan Ms. Excel Untuk Perhitungan  
Production Cost tiap-tiap unit generator Blok I**

Jam	JavaFX 2				Ms. Excel			
	Biaya (\$/h)				Biaya (\$/h)			
	G11	G12	G13	Total	G11	G12	G13	Total
01.00	3535.99	3535.99	3535.99	10608	3535.99	3535.99	3535.985	10608
02.00	3437.41	3437.41	3437.41	10312.2	3437.41	3437.41	3437.414	10312.2
03.00	3442.17	3442.17	3442.17	10326.5	3442.17	3442.17	3442.168	10326.5
04.00	3327.93	3327.93	3327.93	9983.79	3327.93	3327.93	3327.93	9983.79
05.00	3433.51	3433.51	3433.51	10300.5	3433.51	3433.51	3433.508	10300.5
06.00	3476.99	3476.99	3476.99	10431	3476.99	3476.99	3476.99	10431
07.00	3422.75	3422.75	3422.75	10268.3	3422.75	3422.75	3422.75	10268.3
08.00	3465.61	3465.61	3465.61	10396.8	3465.61	3465.61	3465.606	10396.8
09.00	3526.51	3526.51	3526.51	10579.5	3526.51	3526.51	3526.509	10579.5
10.00	3532.39	3532.39	3532.39	10597.2	3532.39	3532.39	3532.392	10597.2
11.00	3379.1	3379.1	3379.1	10137.3	3379.1	3379.1	3379.095	10137.3
12.00	3535.99	3535.99	3535.99	10608	3535.99	3535.99	3535.985	10608
13.00	3453.57	3453.57	3453.57	10360.7	3453.57	3453.57	3453.57	10360.7
14.00	3535.99	3535.99	3535.99	10608	3535.99	3535.99	3535.985	10608
15.00	3535.99	3535.99	3535.99	10608	3535.99	3535.99	3535.985	10608
16.00	3535.99	3535.99	3535.99	10608	3535.99	3535.99	3535.985	10608
17.00	3535.99	3535.99	3535.99	10608	3535.99	3535.99	3535.985	10608
18.00	3535.99	3535.99	3535.99	10608	3535.99	3535.99	3535.985	10608
19.00	3535.99	3535.99	3535.99	10608	3535.99	3535.99	3535.985	10608
20.00	3535.99	3535.99	3535.99	10608	3535.99	3535.99	3535.985	10608
21.00	3535.99	3535.99	3535.99	10608	3535.99	3535.99	3535.985	10608
22.00	3535.99	3535.99	3535.99	10608	3535.99	3535.99	3535.985	10608
23.00	3535.99	3535.99	3535.99	10608	3535.99	3535.99	3535.985	10608
24.00	3535.99	3535.99	3535.99	10608	3535.99	3535.99	3535.985	10608

**Tabel 5.6 Validasi JavaFX 2 dengan Ms. Excel Untuk Perhitungan  
Production Cost tiap-tiap unit generator Blok II**

Jam	JavaFX 2				Ms. Excel			
	Biaya (\$/h)				Biaya (\$/h)			
	G21	G22	G23	Total	G21	G22	G23	Total
01.00	4151.79	0	0	4151.79	4151.79	0	0	4151.79
02.00	0	0	0	0	0	0	0	0
03.00	0	0	0	0	0	0	0	0

04.00	0	0	0	0	0	0	0	0
05.00	0	0	0	0	0	0	0	0
06.00	0	0	0	0	0	0	0	0
07.00	0	0	0	0	0	0	0	0
08.00	0	0	0	0	0	0	0	0
09.00	3200.44	0	0	3200.44	3200.44	0	0	3200.44
10.00	0	0	0	0	0	0	0	0
11.00	0	0	0	0	0	0	0	0
12.00	3371.72	0	0	3371.72	3371.72	0	0	3371.72
13.00	0	0	0	0	0	0	0	0
14.00	4792.29	0	0		4792.29	0	0	
15.00	4266.92	0	0	4266.92	4266.92	0	0	4266.92
16.00	5215.09	0	0	5215.09	5215.09	0	0	5215.09
17.00	4483.26	0	0	4483.26	4483.26	0	0	4483.26
18.00	5292.78	5292.78	0	10585.6	5292.78	5292.78	0	10585.6
19.00	5480.49	5480.49	5480.49	16441.5	5480.49	5480.49	5480.488	16441.5
20.00	5533.23	5533.23	5533.23	16599.7	5533.23	5533.23	5533.23	16599.7
21.00	5642.61	5642.61	5642.61	16927.8	5642.61	5642.61	5642.606	16927.8
22.00	5897.5	5897.5	0	11795	5897.5	5897.5	0	11795
23.00	6682.71	0	0	6682.71	6682.71	0	0	6682.71
24.00	4392.59	0	0	4392.59	4392.59	0	0	4392.59

**Tabel 5.7 Validasi JavaFX 2 dengan Ms. Excel Untuk Perhitungan  
Production Cost tiap-tiap unit generator Blok III**

Jam	JavaFX 2				Ms. Excel			
	Biaya (\$/h)				Biaya (\$/h)			
	G31	G32	G33	Total	G31	G32	G33	Total
01.00	3478.52	3478.52	3478.52	10435.6	3478.52	3478.52	3478.524	10435.6
02.00	3478.07	3478.07	3478.07	10434.2	3478.07	3478.07	3478.074	10434.2
03.00	3478.52	3478.52	3478.52	10435.6	3478.52	3478.52	3478.524	10435.6
04.00	3386.82	3386.82	3386.82	10160.5	3386.82	3386.82	3386.82	10160.5
05.00	3474.82	3474.82	3474.82	10424.5	3474.82	3474.82	3474.819	10424.5
06.00	3478.52	3478.52	3478.52	10435.6	3478.52	3478.52	3478.524	10435.6
07.00	3478.52	3478.52	3478.52	10435.6	3478.52	3478.52	3478.524	10435.6
08.00	3478.52	3478.52	3478.52	10435.6	3478.52	3478.52	3478.524	10435.6
09.00	3478.52	3478.52	3478.52	10435.6	3478.52	3478.52	3478.524	10435.6
10.00	3478.52	3478.52	3478.52	10435.6	3478.52	3478.52	3478.524	10435.6
11.00	3429.49	3429.49	3429.49	10288.5	3429.49	3429.49	3429.49	10288.5

12.00	3478.52	3478.52	3478.52	10435.6	3478.52	3478.52	3478.524	10435.6
13.00	3478.52	3478.52	3478.52	10435.6	3478.52	3478.52	3478.524	10435.6
14.00	3478.52	3478.52	3478.52	10435.6	3478.52	3478.52	3478.524	10435.6
15.00	3478.52	3478.52	3478.52	10435.6	3478.52	3478.52	3478.524	10435.6
16.00	3478.52	3478.52	3478.52	10435.6	3478.52	3478.52	3478.524	10435.6
17.00	3478.52	3478.52	3478.52	10435.6	3478.52	3478.52	3478.524	10435.6
18.00	3478.52	3478.52	3478.52	10435.6	3478.52	3478.52	3478.524	10435.6
19.00	3478.52	3478.52	3478.52	10435.6	3478.52	3478.52	3478.524	10435.6
20.00	3478.52	3478.52	3478.52	10435.6	3478.52	3478.52	3478.524	10435.6
21.00	3478.52	3478.52	3478.52	10435.6	3478.52	3478.52	3478.524	10435.6
22.00	3478.52	3478.52	3478.52	10435.6	3478.52	3478.52	3478.524	10435.6
23.00	3478.52	3478.52	3478.52	10435.6	3478.52	3478.52	3478.524	10435.6
24.00	3478.52	3478.52	3478.52	10435.6	3478.52	3478.52	3478.524	10435.6

Tabel 5.5, tabel 5.6 dan tabel 5.7 diatas memperlihatkan hasil validasi JavaFX 2 dengan Ms. Excel untuk perhitungan *Cost Production* dengan load 1086,43 MW. Dimana dari hasil validasi tersebut antara hasil perhitungan dengan JavaFX 2 dan Ms. Excel tidak terdapat nilai *error*.

Untuk menjelaskan tampilan *cost production* saat validasi, dapat diambil contoh pada pengujian beban pada pukul 01.00 sebesar 1086,4278 MW pada gambar 5.3.



Blok	Konfigurasi	A	B	C	Load Demand	P G1.1	P G1.2	P G1.3	Hasil Perhitungan	FG1.1	FG1.2	FG1.3	Total Cost	
I	Non Konfigurasi	1010,8	19,558	0,023191	0	0	0	0	0	0	0	0	0	
	1.1.1	1010,8	19,558	0,023191	0	0	0	0	0	0	0	0	0	
	2.2.1	2087,2	19,558	0,023191	0	0	0	0	0	0	0	0	0	
	3.2.1	3129,8	19,558	0,023191	526,26	112,45	112,45	112,45	2130,9	6597,217	878,7508	3535,922	3535,922	10807,7674
II	Non Konfigurasi	2818,3	37,748	0,046358	23,8078	33,9078	0	0	2818,3	1279,832	32,34104	4111,753	0	4151,79268
	1.1.1	2818,3	37,748	0,046358	0	0	0	0	0	0	0	0	0	
	2.2.1	3626,6	37,748	0,046358	0	0	0	0	0	0	0	0	0	
	3.2.1	8424,0	37,748	0,046358	0	0	0	0	0	0	0	0	0	
III	Non Kombinasi	1010,7	19,885	0,019418	0	0	0	0	0	0	0	0	0	
	1.1.1	1010,7	19,885	0,019418	0	0	0	0	0	0	0	0	0	
	2.2.1	2087,4	19,885	0,019418	0	0	0	0	0	0	0	0	0	
	3.2.1	3113,6	19,885	0,019418	536,36	112,45	112,45	112,45	4214,5	6696,155	1485,232	4118,458	4118,458	10845,0668

**Gambar 5.23 Validasi program JavaFX 2 dengan Ms. Excel untuk *cost production* tiap-tiap unit generator dengan beban 1086,4278 MW**

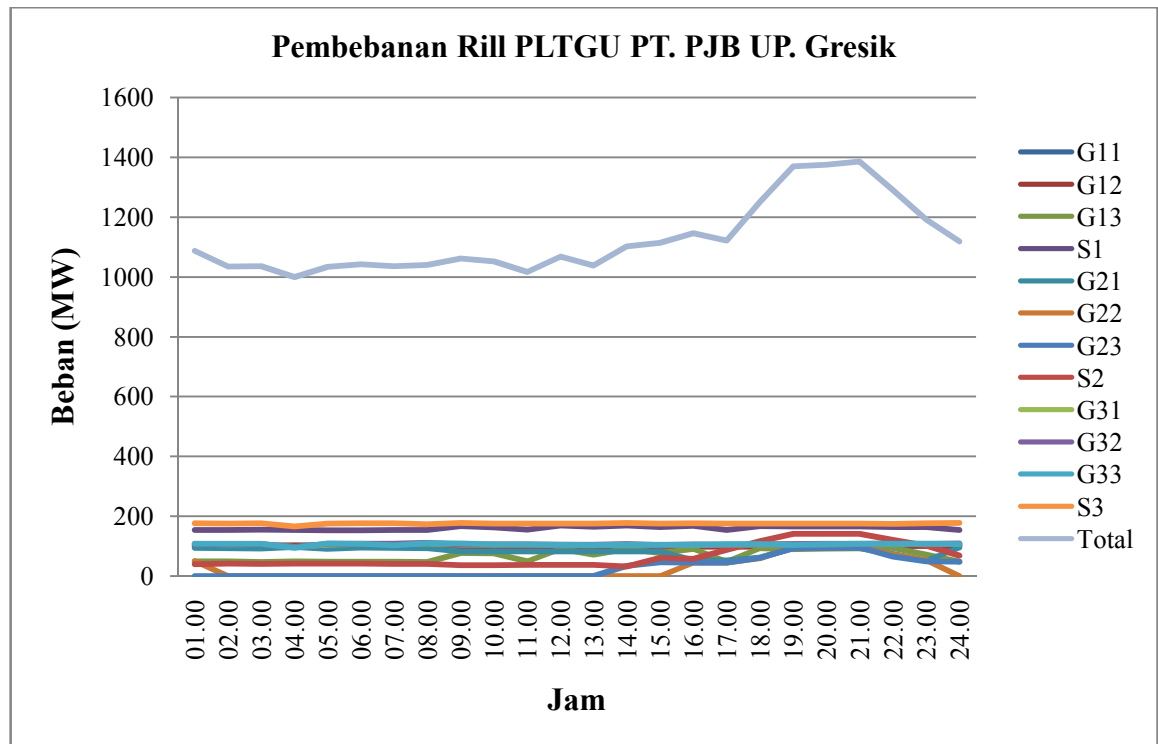
Untuk mengetahui efektifitas dan efisiensi hasil optimasi dengan model *Unit Commitment* yang menggabungkan metode *Priority List* dengan *Dynamic Programming* dan model *Economic Dispatch* metode *Lagrange*, biaya pembangkitan hasil perhitungan program JavaFX 2 akan dibandingkan dengan biaya pembangkitan riil di PLTGU PT. PJB UP. Gresik. Data beban yang dipakai sebagai perbandingan antara perhitungan model *Unit Commitment* serta model *Economic Dispatch* dan pembangkitan riil merupakan daya beban nyata pada periode yang sama yaitu pada tanggal 31 Maret 2013.

## 5.2 Simulasi dan Analisa

### 5.3.1 Pola Pembebanan Riil PLTGU PT. PJB UP. Gresik

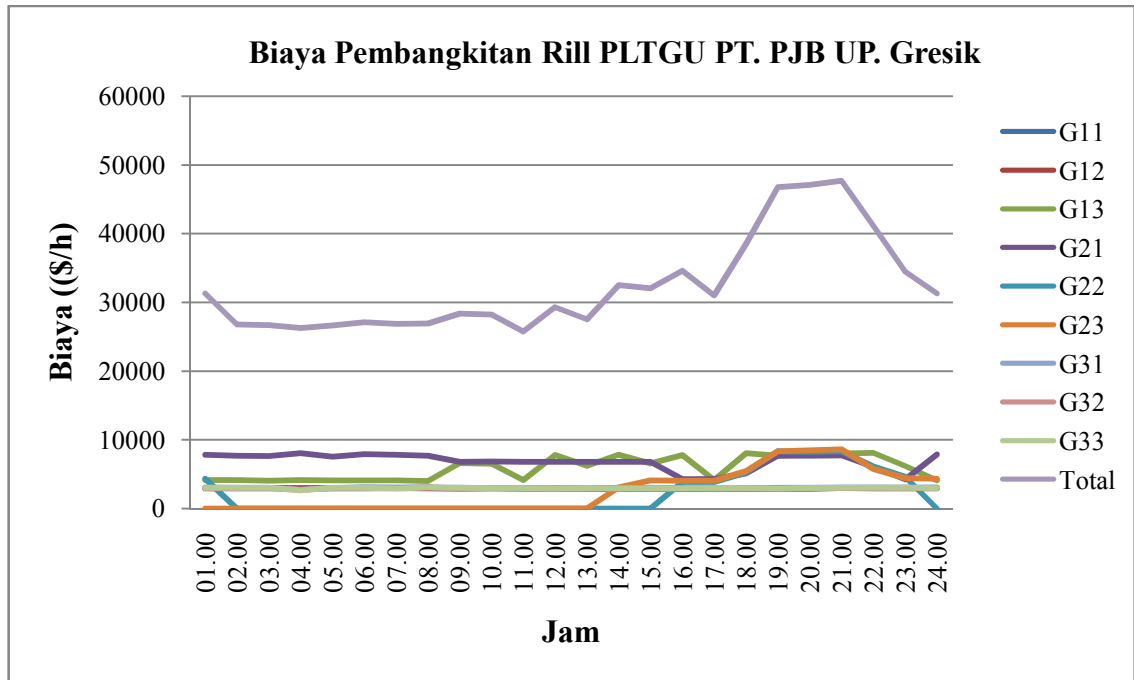
Beban harian dengan interval waktu 1 jam pada tanggal 31 Maret 2013 mengikutsertakan semua blok pembangkitnya dalam proses pembangkitan sejak awal dan ditunjukkan pada gambar 5.7. beban terendah bernilai 998,9489 MW terjadi pada pukul 04.00 dini hari dan beban tertinggi (*peak load*) terjadi pada

pukul 21.00 dengan nilai sebesar 1385,2842 MW (data dapat dilihat dalam lampiran 6).



**Gambar 5.24 Pembebanan Rill Unit-unit PLTGU PT. PJB UP. Gresik**

Unit-unit pembangkit pada pembangkitan riil di PLTGU PT. PJB UP. Gresik tidak ada yang dioperasikan pada kapasitas maksimum. Blok I (G11, G12, G13, dan S1) dan blok III (G31, G32, G33 dan S3) beroperasi dalam konfigurasi *state 4* dengan besar daya keluaran unit-unit pembangkit gas yang hamper sama. Pada blok 2 yang dikenal sebagai blok termahal karena menggunakan bahan bakar HSD, G21 turut dioperasikan sejak awal periode jam 13.00 dan 15.00. pola pembebanan blok 2 sedikit divariasikan dalam berbagai konfigurasi dengan *state 2* (G21 dan S2) sebagai konfigurasi paling sederhana.



**Gambar 5.25 Biaya Pembangkitan Rill PLTGU PT. PJB UP. Gresik**

Biaya pembangkitan rill yang dihasilkan pembebanan rill di PLTGU PT. PJB UP. Gresik ditunjukkan pada gambar 5.8. Biaya pembangkitan yang hampir konstan dihasilkan oleh unit-unit pembangkit pada blok III (G31, G32, dan G33), sedangkan unit-unit pembangkit pada blok I (G11, G12, dan G13) dan blok II (G21, G22, dan G23) menghasilkan biaya pembangkitan yang cukup variatif.

### 5.3.2 Pola Pembebanan Dengan Model *Unit Commitment* Dan *Economic Dispatch* per Blok

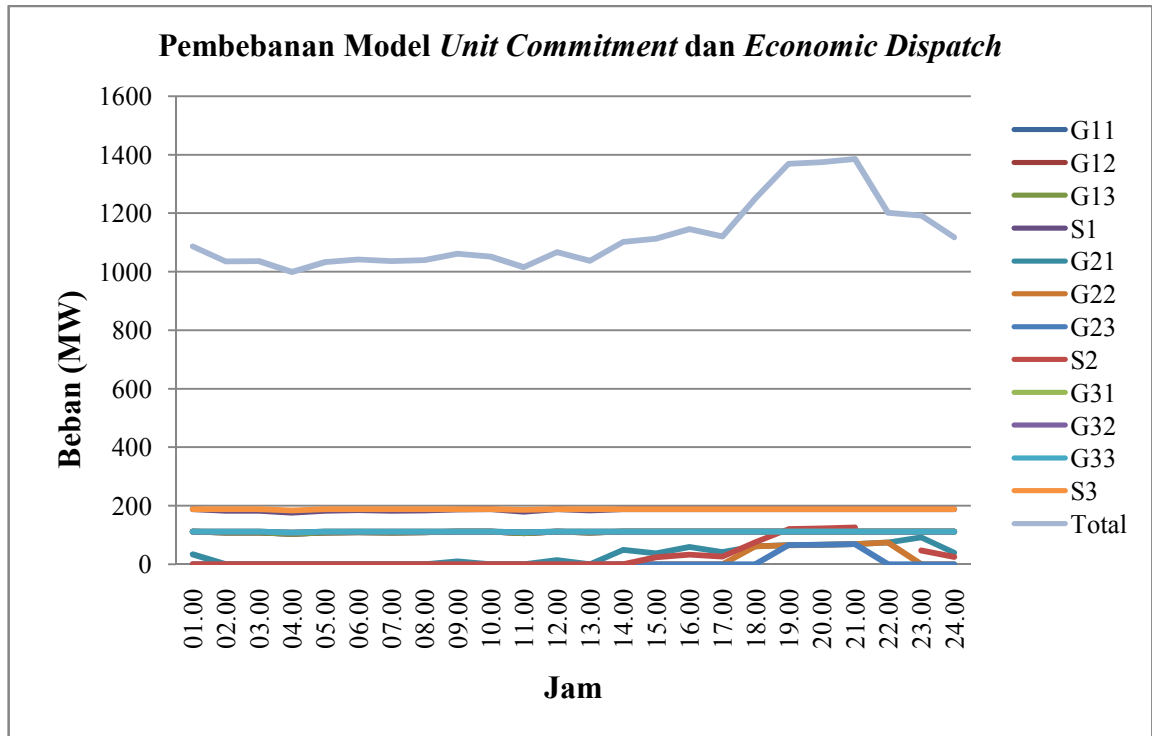
Data beban tanggal 31 Maret 2013 menunjukkan nilai yang cukup variatif dengan variasi nilai mendekati ambang batas jumlah kapasitas maksimal blok 3 dan blok 1 sebesar 1052,52 MW.

Untuk melayani beban-beban ini, metode *Unit Commitment* memprioritaskan memaksimalkan kapasitas pembangkitan blok 3 terlebih dahulu.



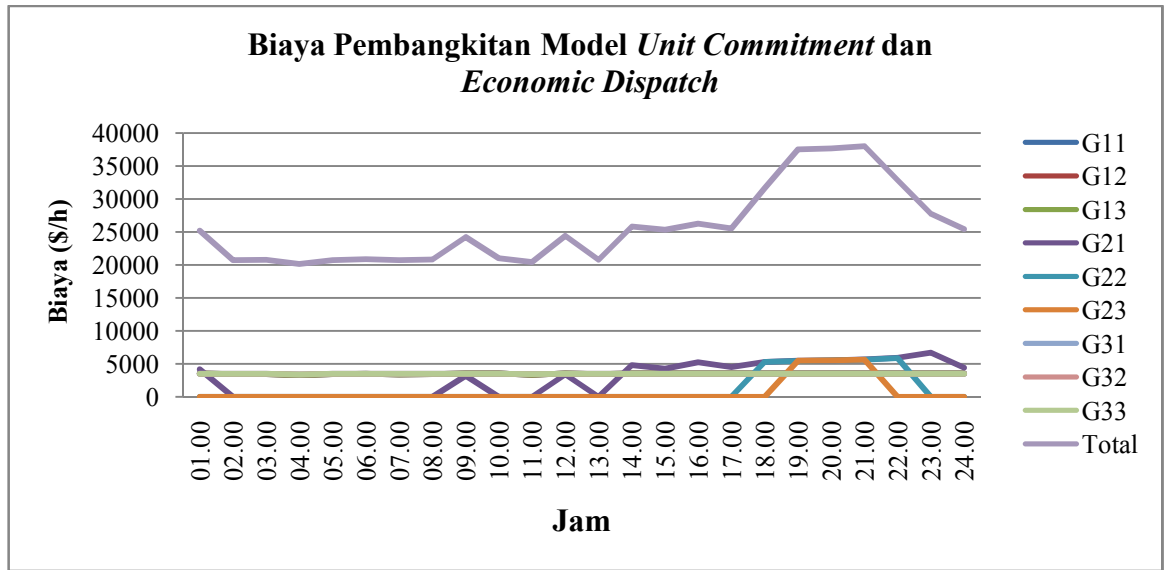
Sisa beban yang belum terlayani akan diberikan kepada blok 1. Jika pembangkitan pada blok 1 sudah mencapai kapasitas maksimal dan masih ada beban yang belum terlayani, sisa beban akan diberikan kepada blok 2. Hasil pembebanan unit-unit pembangkit dapat dilihat dalam gambar 5.9.

Proses pembangkitan blok III (G31, G32, G33, dan S3) dan blok I (G11, G12, G13, dan S1) menggunakan konfigurasi *state* 4 (semua unit-unit pembangkit dioperasikan) untuk melayani semua data terkait besarnya nilai beban. Sedangkan pada blok II, proses pembangkitan menggunakan berbagai konfigurasi sesuai nilai beban dan pilihan konfigurasi yang mampu menghasilkan biaya termurah selama periode 14 jam yang datanya dapat dilihat dalam lampiran 5. Penggunaan konfigurasi *state* 1 (G21) pada blok 2 terjadi pada pukul 01.00, 09.00, 12.00 dan 14.00, *state* 2 (G21 dan S2) terjadi pada jam 15.00, 16.00, 17.00, 23.00 dan 24.00, *state* 3 (G21, G22, dan S2) terjadi sebanyak 2 kali pada jam 18.00 dan 22.00, sedangkan penggunaan *state* 4 dijadwalkan terjadi pada jam 19.00, 20.00 dan 21.00. Selain jam-jam tersebut, blok 2 tidak dioperasikan karena beban sudah terlayani oleh pembangkitan pada blok 1 dan blok 3.



**Gambar 5.26** Pembebanan Unit-unit Pembangkit dengan Model *Unit Commitment* dan *Economic Dispatch*

Pola biaya pembangkitan hasil perhitungan pembebanan dengan pemodelan *Unit Commitment* dan *Economic Dispatch* mengikuti pola pembebanan yang ditunjukkan pada gambar 5.10. Biaya tertinggi dihasilkan oleh unit-unit pembangkit gas blok 3 (G31, G32, dan G33) yang mengalami pembebanan penuh. Biaya pembangkitan variatif dihasilkan oleh pembangkitan unit-unit pembangkit gas blok I (G11, G12, dan G13) dan blok I (G11, G12, dan G13) sesuai beban yang harus dilayani.

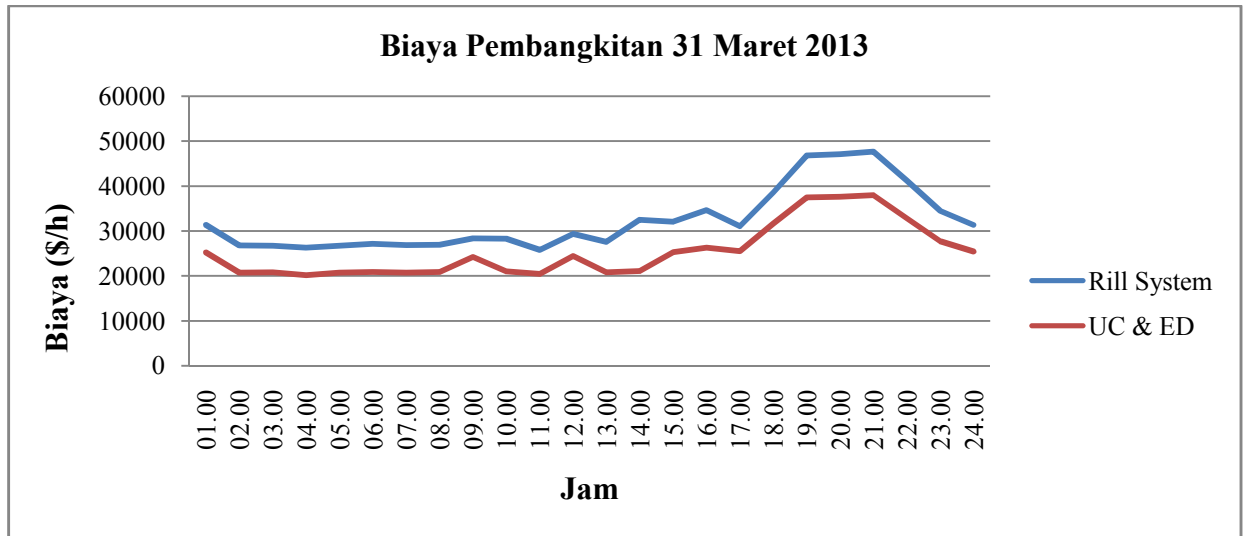


**Gambar 5.27 Biaya Pembangkitan Hasil Perhitungan dengan Pemodelan *Unit Commitment* dan *Economic Dispatch***

Hal lain yang dapat dilihat pada biaya pembangkitan unit-unit pembangkit gas di blok 2 (G21, G22 dan G23) adalah besarnya biaya pembangkitan ketiga unit tersebut sempat melebihi biaya pembangkitan maksimal unit-unit pembangkit gas di blok 1 (G11, G12 dan G13) dan blok 3 (G31, G32 dan G33), padahal daya keluarannya tidak sebesar daya keluaran unit-unit pembangkit gas kedua blok. Kondisi ini membuktikan bahwa penggunaan bahan bakar HSD sebagai bahan bakar pembangkitan sangat tidak ekonomis dan tidak efisien.

### 5.3.3 Perbandingan Biaya Pembangkitan

Gambar 5.28 memperlihatkan bahwa pola pengoperasian riil di PLTGU PT. PJB UP. Gresik menghasilkan biaya pembangkitan lebih mahal daripada biaya pembangkitan hasil simulasi model *Unit Commitment* dan *Economic Dispatch* (data dapat dilihat dalam lampiran 5.1 dan 6.1).



**Gambar 5.28 Perbandingan Biaya Pembangkitan Hasil Pembebanan ED dan UC dan Pembangkitan Riil PLTGU PT. PJB UP. Gresik**

Pada *real system* total biaya produksi yang dihasilkan selama satu hari sebesar US\$ 774.910,3471. Sedangkan untuk memenuhi permintaan beban tanggal 31 Maret 2013, pola pembangkitan dengan pemodelan *Unit Commitment* dan *Economic Dispatch* sebesar US\$ 609.730,312. Sehingga pola pembangkitan dengan pemodelan *Unit Commitment* dan *Economic Dispatch* akan mampu menghemat biaya pembangkitan sebesar US\$ 165189,8361.

Dari hasil simulasi dapat dibuktikan bahwa optimasi pembebanan ekonomis dengan model *Unit Commitment* yang mengembangkan metode *Priority List* dan *Dynamic Programming* dan model *Economic Dispatch* dengan metode *Lagrange* mampu menghasilkan solusi yang lebih optimal dibandingkan dengan pembangkitan riil PLTGU PT. PJB UP. Gresik.