

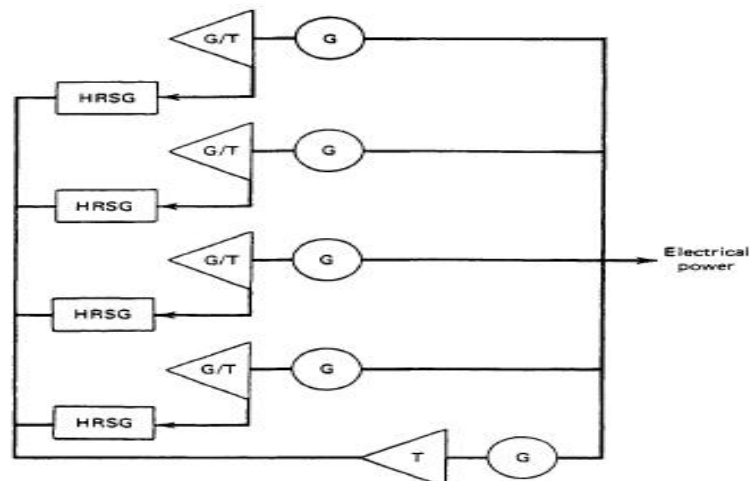
## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Pembangkit Listrik Tenaga Gas Uap

Pembangkit listrik tenaga gas uap (PLTGU) adalah pembangkit daur ganda (*combined cycle*) yang menggunakan gas buang bertemperatur tinggi dari satu turbin gas atau lebih untuk menghasilkan uap dan digunakan untuk menjalankan generator turbin uap. Keuntungan dari penggunaan pembangkit listrik daur ganda adalah tingginya efisiensi pembangkitan (Wood and Wollenberg, 1996:16)

Bagian-bagian utama PLTGU terdiri dari turbin gas beserta generatornya, ketel / HRSG (*Heat Recovery Steam Generator*), turbin uap beserta generatornya dan alat pendukung lainnya.



Sumber : Wood and Wollenberg (1996)

**Gambar 2.1 Sebuah Pembangkit Listrik Daun Ganda (*Combined Cycle*) dengan 4 Turbin Gas dan 1 Turbin Uap**

Gambar 2.1 menggambarkan konfigurasi sebuah pembangkit listrik tenaga gas dan uap dengan empat turbin gas beserta generatornya, empat HRSG yang masing-masing menghubungkan sebuah turbin gas dengan turbin uap, serta satu turbin uap dan generatornya. Karakteristik efisiensi pembangkitan bergantung kepada jumlah turbin gas yang dioperasikan. Semakin banyak turbin gas yang dioperasikan, gas buang yang dapat digunakan untuk memanaskan air di dalam HRSG juga semakin banyak sehingga keluaran turbin uap menjadi semakin besar.

### **2.1.1 Proses Pembangkitan Listrik di PLTGU**

Proses pembangkitan listrik di PLTGU dimulai dari turbin gas. Turbin gas yang berfungsi sebagai pembangkit listrik tahap pertama mempunyai beberapa peralatan utama, yaitu :

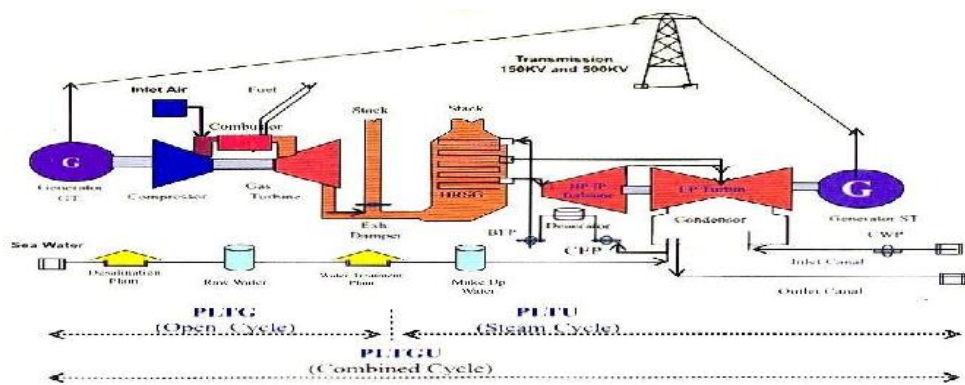
- a. Ruang pemampatan udara (*compressor*)
- b. Ruang pembakaran (*combustor*)
- c. Turbin
- d. Generator
- e. Penggerak mula (*starting device*)

Proses pembangkitan diawali dengan menjalankan motor starter (penggerak mula) yang akan memutar *compressor* untuk memampatkan udara pembakaran. Bersamaan dengan proses pemampatan udara pada ruang bakar, bahan bakar gas bumi atau solar / HSD diinjeksikan dan kemudian dinyalakan dengan busi (untuk awal pembakaran) sehingga terjadilah pembakaran didalam ruang bakar. Setelah gas hasil pembakaran mampu memutar turbin, *compressor* dan generator, motor *starter* akan mati secara otomatis pada saat kecepatan

putarnya mencapai nilai tertentu, sedangkan putaran turbin *compressor* terus naik hingga kecepatannya mencapai nilai yang telah ditentukan (*full speed no load*). Selanjutnya, generator menghasilkan energi listrik yang siap disalurkan kepada konsumen.

Gas sisa pembakaran dari turbin gas dilewatkan ke dalam HRSG untuk menghasilkan uap tekanan tinggi dan tekanan rendah. Ketel ini tanpa pembakaran jadi murni menggunakan gas sisa pembakaran dari turbin gas.

Uap hasil produksi ketel / HRSG digunakan untuk menggerakkan turbin uap. Turbin uap adalah jenis *compound tandem* yang terdiri dari turbin tekanan tinggi dan turbin tekanan rendah. Uap dari saluran tekanan tinggi yang masuk ke turbin tekanan tinggi dan uap dari saluran tekanan rendah yang masuk kedalam turbin tekanan rendah secara bersamaan dikondensasikan dengan kondensor. Air kondensasi dipanaskan kembali ke ketel / HRSG sehingga kembali terbentuk uap untuk memutar turbin. Energi mekanik turbin digunakan memutar generator untuk menghasilkan energi listrik yang kemudian disalurkan kepada konsumen.



Sumber : Profil Unit Pembangkitan Gresik, 2007.

**Gambar 2.2 Skema Pembangkitan PLTGU**

## 2.2 Karakteristik Pembangkit Termal

Karakteristik pembangkit termal terdiri dari beberapa macam, diantaranya adalah karakteristik *input-output*, karakteristik *heat rate*, karakteristik *incremental heat rate* dan *incremental fuel cost* (Wood and Wollenberg, 1996:8).

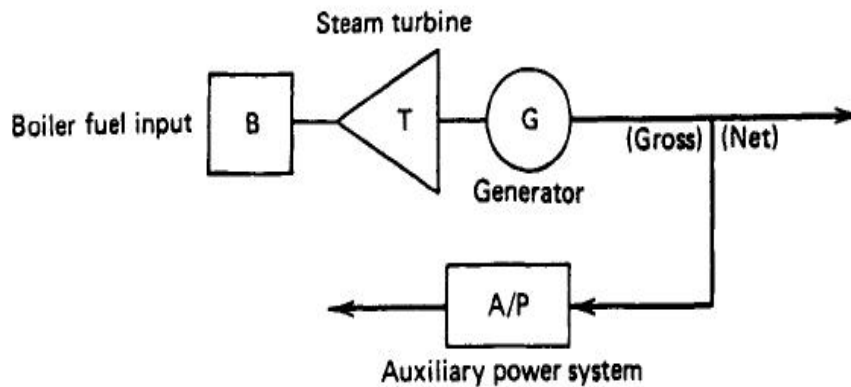
### 2.2.1 Karakteristik *Input-Output*

Karakteristik *input-output* menggambarkan hubungan antara masukan pembangkit sebagai fungsi dari keluaran pembangkit. Pada pembangkit termis, masukannya adalah bahan bakar yang dinyatakan dalam nilai uang sebagai gambaran biaya yang diperlukan untuk bahan bakar yang dinotasikan  $F$  (\$/jam) dengan keluaran daya yang dibangkitkan dinyatakan dalam Megawatt (MW). Masukan juga dinyatakan dalam satuan panas BTU/jam dari bahan bakar yang diberikan pada boiler untuk menghasilkan keluaran pembangkit dan dinotasikan  $H$  (MBTU/h) dengan  $H$  adalah *heat* (panas), BTU adalah *British Thermal Unit* (satuan panas sistem inggris), dan  $h$  adalah *hour* (jam). Hubungan antara  $H$  dan  $F$  dapat dinyatakan dalam persamaan

$$F = H \times \frac{\$}{MBTU} \quad (2.1)$$

dimana  $\frac{\$}{MBTU}$  adalah biaya yang diperlukan per satuan panas dari bahan bakar.

Ciri dari unit Boiler-Turbine-Generator dapat digambarkan dalam gambar berikut ini :



Sumber : Wood and Wollenberg (1996)

**Gambar 2.3 Unit Boiler – Turbin – Generator**

Gambar 2.3 menggambarkan sebuah boiler yang menghasilkan uap untuk menjalankan turbin yang dikopel dengan rotor dari generator. Keluaran pembangkit tidak hanya dihubungkan dengan sistem, tetapi juga dengan sistem peralatan bantu. Keluaran pembangkit didefinisikan sebagai daya yang dikeluarkan oleh generator untuk beban, selain untuk keperluan pembangkit sendiri. Jadi, daya keluaran dalam karakteristik *input-output* adalah daya netto dari pembangkit dan dinotasikan P (MW).

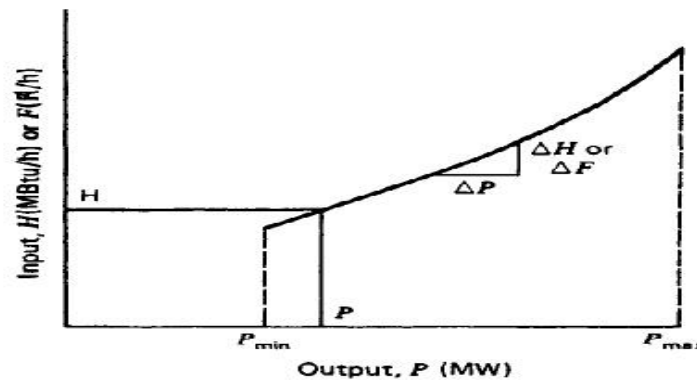
Persamaan karakteristik *input-output* adalah sebagai berikut :

$$H = f. (P) \tag{2.2}$$

atau

$$F = f (P) \tag{2.3}$$

Sedangkan kurvanya dapat digambarkan seperti pada gambar di bawah ini



Sumber : Wood and Wollenberg (1996)

**Gambar 2.4 Kurva Karakteristik Input-Ouput**

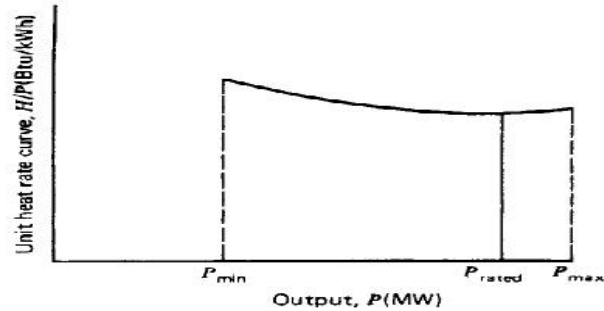
Karakteristik *input-output* berguna untuk mengetahui efisiensi maksimal setiap unit pembangkit termis. Unit pembangkit mempunyai efisiensi tinggi apabila mempunyai kurva dengan lengkungan yang landai. Hal ini berarti bahwa dengan masukan bahan bakar tetap dapat melayani kebutuhan daya yang meningkat.

### 2.2.2 Karakteristik *Heat Rate*

Karakteristik *heat rate* merupakan karakteristik yang menunjukkan efisiensi mesin dan digunakan dalam pengaturan pembebanan beberapa unit pembangkit. Kurva *heat rate* sebuah unit pembangkit menunjukkan masukan kalor dalam BTU yang diberikan untuk menghasilkan keluaran energi KWh pada MW keluaran dari suatu unit.

$$HR = \frac{I}{L \times h} \text{ (BTU / KWh)} \quad (2.4)$$

Dimana =  $I$  : masukan ;  $L$  : beban ;  $h$  : waktu operasi

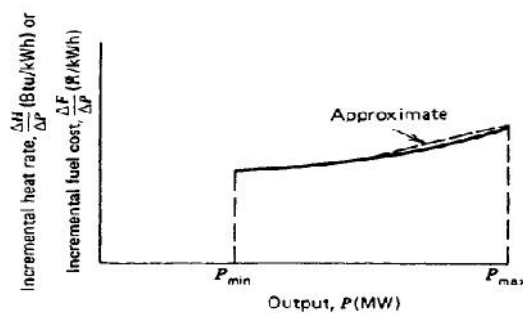


Sumber : Wood and Wollenberg (1996)

**Gambar 2.5 Kurva Karakteristik Heat Rate**

### 2.2.3 Karakteristik *Incremental Heat Rate* dan *Incremental Fuel Cost*

Karakteristik *incremental heat rate* dan *incremental fuel cost* menunjukkan besarnya perubahan masukan tambahan tingkat panas dalam BTU atau biaya bahan bakar dalam *dollar* (\$) bila terjadi perubahan keluaran daya pembangkit dalam MW.



Sumber : Wood and Wollenberg (1996)

**Gambar 2.6 Kurva Karakteristik *Incremental Heat Rate* dan *Incremental Fuel Cost***

Hubungan-hubungan karakteristik penambahan *incremental heat rate* (IHR) dan *incremental fuel cost* (IFC) dapat ditunjukkan dalam persamaan

$$\text{IHR} = \frac{\Delta H}{\Delta P} \left( \frac{\text{MBTU}}{\text{KWh}} \right) \quad (2.5)$$

$$\text{IFC} = \frac{\Delta F}{\Delta P} \left( \frac{\$}{\text{KWh}} \right) \quad (2.6)$$

Bila harga  $\Delta$  sangat kecil, maka dapat dinyatakan dengan persamaan

$$\text{IHR} = \frac{dH}{dP} \left( \frac{\text{MBTU}}{\text{KWh}} \right) \quad (2.7)$$

$$\text{IFC} = \frac{dF}{dP} \left( \frac{\$}{\text{KWh}} \right) \quad (2.8)$$

## 2.3 Beban Listrik

### 2.3.1 Jenis dan Pola Beban

Beban listrik adalah besarnya konsumsi pemakaian tenaga listrik oleh pelanggan (konsumen) untuk melakukan kerja atau memenuhi kebutuhannya sehingga beban listrik bersifat dinamis. Sifat beban yang dinamis menjadi permasalahan vital sehingga senantiasa diupayakan menjaga keseimbangan daya listrik yang dibangkitkan sama dengan besar kebutuhan beban listrik konsumen.

Dalam sistem operasi pembangkit tenaga listrik dibedakan beberapa istilah beban listrik, yaitu :

- a. Beban puncak (*peak load*), yaitu harga tertinggi dari kebutuhan beban listrik pada periode waktu tertentu.



- b. Beban rata-rata (*average load*), yaitu produksi tenaga listrik yang dibangkitkan selama periode waktu tertentu dibagi dengan periode waktu pembangkitannya.
- c. Beban dasar (*base load*), yaitu beban total sistem dikurangi dengan fluktuasi beban listrik selama masa pembangkitan.

Kurva yang menggambarkan penunjukan beban listrik pada setiap satuan waktu disebut kurva beban listrik (*load curve*), misalnya kurva beban harian dalam selang periode waktu 24 jam. Karakteristik beban, yang lazim disebut sebagai kurva beban, terbagi menjadi beberapa pola, yaitu :

- a. Beban jam-an, yaitu gambaran besarnya beban yang terjadi pada setiap jam yang harus dipenuhi oleh sistem pembangkitan. Perubahan besar beban menyebabkan operasi pembangkit juga berubah agar faktor ekonomi tetap tercapai.
- b. Beban harian, yaitu gambaran besarnya beban yang harus dilayani oleh pembangkit setiap harinya. Beban ini digunakan dalam mencari jadwal pengoperasian unit-unit pembangkit agar diperoleh biaya operasi yang ekonomis.
- c. Beban mingguan, yaitu gambaran besarnya beban yang harus dilayani oleh pembangkit kepada beban selama tujuh hari.
- d. Beban tahunan, yaitu gambaran besarnya beban yang harus dilayani oleh pembangkit kepada beban dalam kurun waktu satu tahun.

### **2.3.2 Pengaruh Pembebanan terhadap Biaya Bahan Bakar**

Pembebanan menentukan keandalan dan kestabilan sistem tenaga listrik sehingga besar daya siap dibangkitkan setiap saat diupayakan sama dengan kebutuhan beban. Penggunaan bahan bakar fosil pada pembangkit termis akan mendorong pertambahan jumlah bahan bakar sehingga meningkatkan biaya pembangkitan saat melayani beban. Dalam batas tertentu, fluktuasi beban akan menimbulkan fluktuasi biaya bahan bakar serupa sehingga didapat pola korelasi antara beban dan biaya.

Upaya mencapai kondisi optimal pada keluaran daya unit pembangkit dan biaya bahan bakar dapat dilakukan dengan mengatur besar biaya sebagai masukan terhadap keluaran daya pembangkit berdasarkan asumsi bahwa penentuan besar masukan secara optimal akan menghasilkan keluaran yang optimal pula. Biaya setiap keluaran dipengaruhi oleh harga bahan bakar dan efisiensi.

## **2.4 Optimasi dalam bidang tenaga listrik**

Optimasi adalah suatu usaha untuk mendapatkan hasil terbaik dalam suatu kondisi tertentu. Dalam desain, konstruksi, dan perawatan dari segala sistem teknik, dalam setiap tahap pengambilan keputusan para sarjana teknik harus mempertimbangkan aspek teknologi dan managerial. Tujuan utama dari keseluruhan pengambilan keputusan adalah meminimalkan usaha yang dibutuhkan atau memaksimalkan keuntungan yang diinginkan. Sebab usaha yang dibutuhkan atau keuntungan yang diinginkan dalam dunia nyata dapat diekspresikan dalam suatu fungsi variabel-variabel keputusan tertentu, sehingga

optimasi dapat didefinisikan sebagai suatu proses untuk menemukan suatu kondisi yang memberikan harga maksimum atau minimum dari suatu fungsi tersebut (Rao, 1996:1).

Model matematis untuk optimasi adalah (Rao, 1996:6) :

$$\text{Cari } X = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} \text{ yang meminimalkan/memaksimalkan } f(X) \quad (2.9)$$

$$\text{s.t. : } g_j(X) \leq 0 \quad j = 1, 2, \dots, m$$

$$l_j(X) = 0 \quad j = 1, 2, \dots, p$$

dimana :  $X$  = vektor n-dimensi

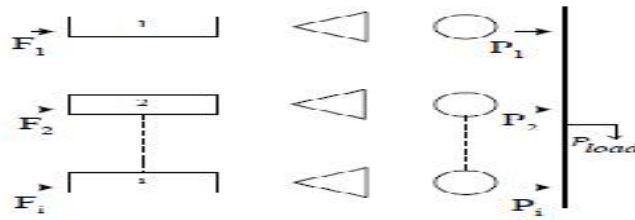
$f(X)$  = *objective function*

$g_j(X)$  = pertidaksamaan *constraint*

$l_j(X)$  = persamaan *constraint*

#### 2.4.1 *Economic Load Dispatch* (Pembagian Beban Secara Ekonomis)

*Power generation dispatch* yang biasanya dikenal dengan *economic dispatch* (ED) adalah suatu permasalahan yang mengasumsikan bahwa jika ada N generator unit yang sudah terhubung ke sebuah single bus-bar melayani beban sebesar  $P_{load}$  seperti diilustrasikan di gambar 3.7. Maka model *economic dispatch* akan menentukan kebijakan operasional yang optimum untuk setiap N unit tersebut (Wood and Wollenberg, 1996:29).



Sumber : Wood and Wollenberg (1996)

**Gambar 2.7 Interkoneksi N generator unit yang melayani beban  $P_{load}$**

Masukan (*input*) dari setiap unit, seperti yang terlihat diatas adalah  $F_i$ , merepresentasikan biaya (*cost rate*) dari setiap unit. Keluaran (*output*) dari setiap unit adalah  $P_i$ , yaitu tenaga listrik yang dibangkitkan oleh setiap unit.

Jadi permasalahan *economic dispatch* menentukan optimum operating policy bagi setiap generator, yaitu seberapa besar dia kontribusi supply daya listrik sedemikian rupa sehingga biaya operasi terutama bahan bakar menjadi minimum. Tentunya dengan asumsi bahwa dalam suatu sistem tenaga listrik ada N generator unit yang mensupply sistem yang telah dieksekusi *Unit Commitment*.

#### 2.4.1.1 Metode *Lagrange*

Aplikasi dari teknik optimalisasi ganda (*dual optimization*) disebut *Lagrange*. Disebut optimalisasi ganda karena  $F_t$  yang diminimalkan dibatasi oleh pembatas lainnya.

$F_t$  adalah *objective function* yang besarnya adalah total biaya untuk mensupply beban yang ada. Sehingga permasalahan optimasi disini adalah meminimalkan  $F_t$ . Batasan terpenting dalam pengoperasian sistem adalah jumlah daya keluaran harus sama dengan besarnya permintaan, yang dirumuskan

$$F_t = F_1 + F_2 + \dots + F_N = \sum_{i=1}^N F_i(P_i) \quad (2.10)$$

Dengan  $P_i$  : pembangkit unit  $i$  (MW)  
 $F_i$  : biaya unit  $i$  (\$)  
 $F$  : biaya total  $n$  unit (\$)  
 $N$  : jumlah unit

Fungsi biaya suatu unit pembangkit diperoleh dari karakteristik unit pembangkit dalam satuan \$/h yang berupa persamaan kuadrat.

$$F_i(P_i) = a + b(P_i) + c(P_i)^2 \quad (2.11)$$

*Constraint* dari permasalahan ini adalah total dari keluaran power sama dengan besarnya kebutuhan beban.

$$\begin{aligned} \emptyset = 0 &= P_{load} - \sum_{i=1}^N P_i \\ \sum_{i=1}^{n_g} P_i &= P_{load} \end{aligned} \quad (2.12)$$

*Constraint* kedua adalah keluaran power dari masing-masing unit harus lebih besar atau sama dengan power minimum yang diijinkan dan juga harus lebih kecil atau sama dengan power maksimum yang diijinkan.

$$P_{i,min} \leq P_i \leq P_{i,max} \quad (2.13)$$

Untuk membentuk *necessary condition* untuk harga puncak dari *objective function*, tambah fungsi batasan ke *objective function* setelah sebelumnya dikalikan terlebih dahulu dengan suatu *undetermined multiplier*  $\lambda$ . Hal ini dikenal dengan fungsi *Lagrange*.

$$L = F_T + \lambda \emptyset \quad (2.14)$$

Dalam pembentukan kondisi penting pada nilai-nilai ekstrim fungsi objektif, fungsi batasan ditambahkan ke dalam fungsi objektif setelah fungsi batasan dikalikan dengan sebuah faktor pengali yang tidak dapat dihitung yang disebut faktor pengali Lagrange,  $\lambda$ . Kondisi penting untuk nilai ekstrim fungsi

objektif diperoleh dari turunan pertama fungsi Lagrange dengan menetapkan turunan pertamanya sama dengan nol. Penurunan fungsi Lagrange memberikan fungsi batasan. Persamaan  $N$  yang diperoleh dari penurunan sebagian fungsi Lagrange dengan memperhatikan nilai daya keluaran pada suatu waktu ditunjukkan dengan persamaan

$$\frac{\partial L}{\partial P_i} = \frac{dF_i(P_i)}{dP_i} - \lambda = 0 \quad (2.15)$$

atau

$$0 = \frac{dF_i}{dP_i} - \lambda \quad (2.16)$$

Syarat penting keberadaan biaya operasi minimum dalam sistem pembangkitan termal adalah kenaikan tingkat biaya semua unit sama dengan faktor pengali Lagrange. Untuk itu, persamaan batasan harus ditambah dengan fungsi jumlah daya total harus sama dengan daya beban yang diminta. Sebagai tambahan, terdapat dua pertidaksamaan yang harus dipenuhi oleh setiap unit, yaitu daya keluaran setiap unit harus lebih besar atau sama dengan daya minimal yang bisa dibangkitkan dan juga harus lebih kecil atau sama dengan daya maksimal mampu dibangkitkan pada unit yang bersangkutan.

*Necessary condition* ini dan pertidaksamaan *constraint* dapat diringkas dalam satu set persamaan berikut

$$\begin{array}{ll} \frac{dF_i}{dP_i} = \lambda & N \text{ persamaan} \\ P_{i,\min} \leq P_i \leq P_{i,\max} & 2N \text{ pertidaksamaan} \\ \sum_{i=1}^N P_i = P_{load} & 1 \text{ constraint} \end{array} \quad (2.17)$$

Syarat *Kuhn-Tucker* melengkapi syarat Lagrangian untuk mengikuti ketentuan ketidaksamaan. Syarat-syarat untuk pengiriman daya nyata yang optimal dari pembangkit dengan mengabaikan rugi-rugi daya adalah sebagai berikut (Hadi Saadat,1996):

$$\begin{aligned} \frac{dF_i}{dP_i} = \lambda & \quad \text{untuk} \quad P_{i(\min)} \leq P_i \leq P_{i(\max)} \\ \frac{dF_i}{dP_i} \leq \lambda & \quad \text{untuk} \quad P_i = P_{i(\max)} \\ \frac{dF_i}{dP_i} \geq \lambda & \quad \text{untuk} \quad P_i = P_{i(\min)} \end{aligned} \quad (2.18)$$

Setelah power yang *dispatch* ditemukan, total cost dihitung berdasarkan rumus berikut :

$$C_t = \sum_{i=1}^{n_g} C_i = \sum_{i=1}^n (A_i + B_i x P_i + C_i x P_i^2) \quad (2.19)$$

$C_t$  : total biaya produksi

$C_i$  : biaya produksi dari pembangkit ke  $i$

A,B,C : konstanta

$P_i$  : power yang *dispatch* untuk unit  $i$

$i$  : indeks untuk unit-unit generator = 1,2, ....., N

**Contoh perhitungan optimasi pembangkit termal dengan model *economic dispatch***

Terdapat tiga unit generator dengan spesifikasi sebagai berikut :

Unit	Kurva I/O H	P min (MW)	P max (MW)	Fuel Cost (R/Mbtu)
1	$510 + 7,2 P_1 + 0,00142 P_1^2$	150	600	1,1
2	$310 + 7,85 P_2 + 0,00194 P_2^2$	100	400	1,0
3	$78 + 7,97 P_3 + 0,00482 P_3^2$	50	200	1,0

**Ditanya :** Pembagian pembebanan pada masing-masing pembangkit pada sistem 850 MW dengan metode *Lagrange* dan metode *Iterasi Lambda*.

**Penyelesaian :**

❖ **Perhitungan pembebanan pada masing-masing pembangkit dengan metode Lagrange.**

$$F_1(P_1) = H_1(P_1) * 1,1 = 561 + 7,92 P_1 + 0,001562 P_1^2$$

$$F_2(P_2) = H_2(P_2) * 1,0 = 310 + 7,85 P_2 + 0,00194 P_2^2$$

$$F_3(P_3) = H_3(P_3) * 1,0 = 78 + 7,97 P_3 + 0,00482 P_3^2$$

➤ *Kondisi Optimum Dispatch*

$$\frac{dF_1}{dP_1} = 7,92 + 0,003124 P_1 = \lambda$$

$$P_1 = \frac{\lambda - 7,92}{0,003124}$$

$$\frac{dF_2}{dP_2} = 7,85 + 0,00388 P_2 = \lambda$$

$$P_2 = \frac{\lambda - 7,85}{0,00388} \dots\dots\dots \text{(persamaan 1)}$$

$$\frac{dF_3}{dP_3} = 7,97 + 0,00964 P_3 = \lambda$$

$$P_3 = \frac{\lambda - 7,97}{0,00964}$$

$$P_1 + P_2 + P_3 = 850 \text{ MW} \dots\dots\dots \text{(persamaan 2)}$$

➤ Substitusikan persamaan 1 ke persamaan 2 sehingga didapat nilai sebagai berikut :

$$\begin{aligned} \lambda &= \frac{850 + \frac{7,92}{0,003124} + \frac{7,85}{0,00388} + \frac{7,97}{0,00964}}{\frac{1}{0,003124} + \frac{1}{0,00388} + \frac{1}{0,00964}} \\ &= \frac{850 + 2535,2113 + 2023,196 + 826,763}{320,10243 + 257,731 + 103,734} \\ &= \frac{6235,1703}{681,56743} = 9,148 \text{ R/MWh.} \end{aligned}$$



➤ Substitusi  $\lambda$  ke persamaan 1 :

$$P_1 = \frac{9,148 - 7,92}{0,003124} = \frac{1,228}{0,003124} = 393,2 \text{ MW}$$

$$P_2 = \frac{9,148 - 7,85}{0,00388} = \frac{1,298}{0,00388} = 334,6 \text{ MW}$$

$$P_3 = \frac{9,148 - 7,97}{0,00964} = \frac{1,178}{0,00964} = 122,2 \text{ MW}$$

➤ Check *Necessary condition* :

$$\frac{dF_1}{dP_1} = 7,92 + 0,003124 \cdot 393,2 = 7,92 + 1,228 = 9,148 \text{ R/MWh.}$$

$$\frac{dF_2}{dP_2} = 7,85 + 0,00388 \cdot 334,6 = 7,85 + 1,298 = 9,148 \text{ R/MWh.}$$

$$\frac{dF_3}{dP_3} = 7,97 + 0,00964 \cdot 122,2 = 7,97 + 1,178 = 9,148 \text{ R/MWh.}$$

Karena semua nilai telah memenuhi persamaan  $\frac{dF_i}{dP_i} = \lambda$ , maka nilai P tiap unit generator optimum. Sehingga nilai  $P_1 = 393,2 \text{ MW}$  ;  $P_2 = 334,6 \text{ MW}$  ;  $P_3 = 122,2 \text{ MW}$ .

Sumber : Wood and Wollenberg (1996:32)

## **KASUS II**

Terdapat tiga unit generator dengan spesifikasi sebagai berikut :

Unit	Kurva I/O H	P min (MW)	P max (MW)	Fuel Cost (R/Mbtu)
1	$510 + 7,2 P_1 + 0,00142 P_1^2$	150	600	0,9
2	$310 + 7,85 P_2 + 0,00194 P_2^2$	100	400	1,0
3	$78 + 7,97 P_3 + 0,00482 P_3^2$	50	200	1,0

**Ditanya :** Pembagian pembebanan pada masing-masing unit generator jika  $P_{load} = 850$  MW dengan Metode *Lagrange*.

**Penyelesaian :**

$$F_1(P_1) = H_1(P_1) * 0,9 = 459 + 6,48 P_1 + 0,00128 P_1^2$$

$$F_2(P_2) = H_2(P_2) * 1,0 = 310 + 7,85 P_2 + 0,00194 P_2^2$$

$$F_3(P_3) = H_3(P_3) * 1,0 = 78 + 7,97 P_3 + 0,00482 P_3^2$$

➤ Kondisi *Optimum Dispatch*

$$\frac{dF_1}{dP_1} = 6,48 + 0,00256 P_1 = \lambda \qquad P_1 = \frac{\lambda - 6,48}{0,00256}$$

$$\frac{dF_2}{dP_2} = 7,85 + 0,00388 P_2 = \lambda \qquad P_2 = \frac{\lambda - 7,85}{0,00388} \dots\dots\dots \text{(persamaan 1)}$$

$$\frac{dF_3}{dP_3} = 7,97 + 0,00964 P_3 = \lambda \qquad P_3 = \frac{\lambda - 7,97}{0,00964}$$

$$P_1 + P_2 + P_3 = 850 \text{ MW} \dots\dots\dots \text{(persamaan 2)}$$

➤ Substitusikan persamaan 1 ke persamaan 2 sehingga didapat nilai sebagai berikut :

$$\begin{aligned} \lambda &= \frac{850 + \frac{6,48}{0,00256} + \frac{7,85}{0,00388} + \frac{7,97}{0,00964}}{\frac{1}{0,00256} + \frac{1}{0,00388} + \frac{1}{0,00964}} \\ &= \frac{850 + 2531,25 + 2023,196 + 826,763}{390,625 + 257,731 + 103,734} \\ &= \frac{6231,209}{752,09} = 8,284 \text{ R/MWh.} \end{aligned}$$

➤ Substitusi  $\lambda$  ke persamaan 1 :

$$P_1 = \frac{8,284-6,48}{0,00256} = \frac{1,804}{0,00256} = 704,6 \text{ MW (tidak memenuhi, } > P_{\max}, \text{ set to } P_{\max} =$$

$$600 \text{ MW)} \rightarrow \frac{dF_1}{dP_1} = 6,48 + 0,003124 \cdot 600 = 7,92 + 1,506 = 8,016 \text{ R/ MWh, IC}$$

$< \lambda$ , maka nilai sudah optimum dan dijaga konstan.

$$P_2 = \frac{8,284-7,85}{0,00388} = \frac{0,434}{0,00388} = 111,8 \text{ MW (memenuhi)} \rightarrow \text{set to } (P_{\text{load}} - (P_1 + P_3)) =$$

$$(850 - (600+50)) = 200 \text{ MW} \rightarrow \frac{dF_2}{dP_2} = 7,85 + 0,00388 \cdot 200 = 7,85 + 0,776 =$$

8,626 R/ MWh, IC  $> \lambda$ , maka akan dieksekusi.

$$P_3 = \frac{8,284-7,97}{0,00964} = \frac{0,314}{0,00964} = 32,6 \text{ MW (tidak memenuhi, } < P_{\min}, \text{ set to } P_{\min} = 50$$

$$\text{MW)} \rightarrow \frac{dF_3}{dP_3} = 7,97 + 0,00964 \cdot 50 = 7,97 + 0,482 = 8,452 \text{ R/ MWh, IC } > \lambda,$$

maka akan dieksekusi.

Karena  $P_1$  harus dijaga konstan, maka  $P_2$  dan  $P_3$  dihitung dengan metode

*Lagrange* :

$$P_2 + P_3 = 850 - P_1$$

$$P_2 + P_3 = 850 - 600 = 250 \text{ MW} \dots\dots\dots (\text{persamaan 3})$$

➤ Substitusikan persamaan 1 ke persamaan 3 sehingga didapat :

$$\begin{aligned} \lambda &= \frac{250 + \frac{7,85}{0,00388} + \frac{7,97}{0,00964}}{\frac{1}{0,00388} + \frac{1}{0,00964}} \\ &= \frac{850 + 2023,196 + 826,763}{257,731 + 103,734} \\ &= \frac{3099,959}{361,465} = 8,576 \text{ R/ MWh} \end{aligned}$$

➤ Substitusikan  $\lambda$  ke persamaan 1 sehingga didapat :

$$P_2 = \frac{8,576-7,85}{0,00388} = \frac{0,726}{0,00388} = 187,1 \text{ MW.}$$

$$P_3 = \frac{8,576 - 7,97}{0,00964} = \frac{0,606}{0,00964} = 62,9 \text{ MW.}$$

Check *Necessary condition* :

$$\frac{dF_2}{dP_2} = 7,85 + 0,00388 \cdot 187,1 = 7,85 + 0,726 = 8,576 = \lambda$$

$$\frac{dF_3}{dP_3} = 7,97 + 0,00964 \cdot 62,9 = 7,97 + 0,606 = 8,576 = \lambda$$

Karena semua nilai telah memenuhi persamaan  $\frac{dF_i}{dP_i} = \lambda$ , maka nilai P tiap unit generator optimum. Sehingga nilai  $P_1 = 600 \text{ MW}$  ;  $P_2 = 187,1 \text{ MW}$  ;  $P_3 = 62,9 \text{ MW}$ .

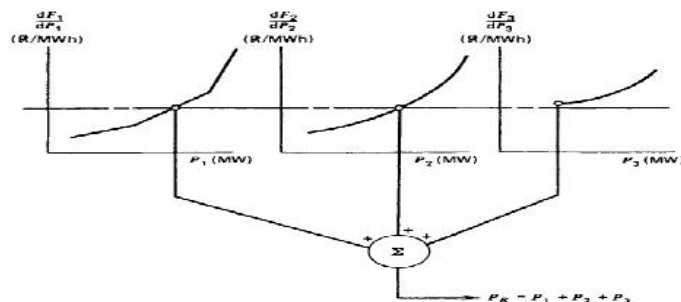
Sumber : Wood and Wollenberg (1996:33)

❖ **Perhitungan pembebanan pada masing-masing pembangkit dengan metode Iterasi Lambda.**

➤ Metode Iterasi Lambda

Pada metode ini  $\lambda$  ditentukan terlebih dahulu, kemudian dengan menggunakan syarat optimum dihitung  $P_i$  (output setiap pembangkit). Dengan menggunakan *constraint*, diperiksa apakah jumlah total dari output sama dengan beban sistem, bila belum harga  $\lambda$  ditentukan kembali.

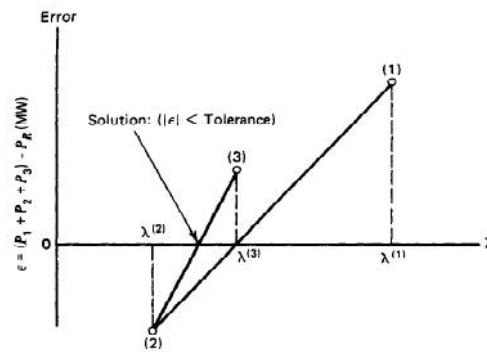
Konsepsi dari metode ini dijelaskan dengan kurva Incremental Fuel Cost dibawah ini :



**Gambar 2.8 Grafik penyelesaian dengan metode Iterasi Lambda**

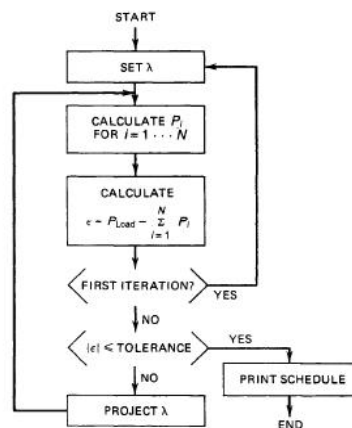
Dari kurva-kurva tersebut dan dengan menetapkan harga  $\lambda$ , maka dapat diperoleh harga  $P_1$ ,  $P_2$  dan  $P_3$ . Untuk harga  $\lambda$  yang pertama tentunya belum merupakan harga  $\lambda$  yang benar.

Bila harga  $P_1 + P_2 + P_3$  lebih kecil dari  $P_R$  maka akan ditentukan kembali harga  $\lambda$  yang lebih besar dari harga  $\lambda$  yang pertama dan sebaliknya, kemudian dihitung penyelesaiannya. Dengan telah diperoleh dua kali perhitungan diatas maka secara *ekstrapolasi* dapat ditentukan harga  $\lambda$  selanjutnya sampai dicapai harga yang dikehendaki (dimana dicapai  $P_1 + P_2 + P_3 = P_R$ ).



**Gambar 2.9 Ekstrapolasi**

Penyelesaian metode ini dengan menggunakan komputer *flow chart* dapat dilihat berikut ini :



**Gambar 2.10 flow chart metode iterasi lambda**

➤ Penyelesaian perhitungan pembebanan ekonomis dengan metode Iterasi Lambda.

Diketahui sistem seperti pada Kasus I.

Ditanya : pembagian pembebanan pada masing-masing pembangkit pada beban sistem 850 MW dengan metode iterasi lambda (error <7,5).

Penyelesaian :

$$\text{Error} = e = P_{\text{load}} - \sum_{i=1}^N P_i$$

$$\sum_{i=1}^N P_i = P_1 + P_2 + P_3$$

Ditentukan  $\lambda$  awal 8 dan 10

❖ Iterasi 1

$$\lambda = \frac{8+10}{2} = 9, \text{ maka}$$

$$\sum_{i=1}^N P_i = \frac{9-7,92}{0,003124} + \frac{9-7,85}{0,00388} + \frac{9-7,97}{0,00964} = 748,95 \text{ MW.}$$

$$e = P_{\text{load}} - \sum_{i=1}^N P_i = 850 - 748,95 = 111,05$$

❖ Iterasi 2

$$\lambda = \frac{9+10}{2} = 9,5$$

$$\sum_{i=1}^N P_i = \frac{9,5-7,92}{0,003124} + \frac{9,5-7,85}{0,00388} + \frac{9,5-7,97}{0,00964} = 1089,7 \text{ MW.}$$

$$e = P_{\text{load}} - \sum_{i=1}^N P_i = 850 - 1089,7 = -239,7$$

❖ Iterasi 3

$$\lambda = \frac{9+9,5}{2} = 9,25$$

$$\sum_{i=1}^N P_i = \frac{9,25-7,92}{0,003124} + \frac{9,25-7,85}{0,00388} + \frac{9,25-7,97}{0,00964} = 919,34 \text{ MW.}$$

$$e = P_{\text{load}} - \sum_{i=1}^N P_i = 850 - 919,34 = -69,34$$

❖ Iterasi 4

$$\lambda = \frac{9+9,25}{2} = 9,125$$

$$\sum_{i=1}^N P_i = \frac{9,125-7,92}{0,003124} + \frac{9,125-7,85}{0,00388} + \frac{9,125-7,97}{0,00964} = 834,14 \text{ MW.}$$

$$e = P_{\text{load}} - \sum_{i=1}^N P_i = 850 - 834,15 = 15,86.$$

❖ Iterasi 5

$$\lambda = \frac{9,25+9,125}{2} = 9,1875$$

$$\sum_{i=1}^N P_i = \frac{9,1875-7,92}{0,003124} + \frac{9,1875-7,85}{0,00388} + \frac{9,1875-7,97}{0,00964} = 876,7 \text{ MW.}$$

$$e = P_{\text{load}} - \sum_{i=1}^N P_i = 850 - 876,7 = -26,7$$

❖ Iterasi 6

$$\lambda = \frac{9,125+9,1875}{2} = 9,16$$

$$\sum_{i=1}^N P_i = \frac{9,16-7,92}{0,003124} + \frac{9,16-7,85}{0,00388} + \frac{9,16-7,97}{0,00964} = 858 \text{ MW.}$$

$$e = P_{\text{load}} - \sum_{i=1}^N P_i = 850 - 858 = -8$$

❖ Iterasi 7

$$\lambda = \frac{9,125+9,16}{2} = 9,14$$

$$\sum_{i=1}^N P_i = \frac{9,14-7,92}{0,003124} + \frac{9,14-7,85}{0,00388} + \frac{9,14-7,97}{0,00964} = 844,4 \text{ MW.}$$

$$e = P_{\text{load}} - \sum_{i=1}^N P_i = 850 - 844,4 = 5,6 \text{ (memenuhi } < 7,5)$$

➤ Substitusi  $\lambda$  :

$$P_1 = \frac{9,14-7,92}{0,003124} = \frac{1,22}{0,003124} = 390,52 \text{ MW}$$

$$P_2 = \frac{9,14-7,85}{0,00388} = \frac{1,29}{0,00388} = 332,47 \text{ MW}$$

$$P_3 = \frac{9,14-7,97}{0,00964} = \frac{1,17}{0,00964} = 121,37 \text{ MW}$$

**Tabel 2.1 Perbandingan hasil perhitungan optimasi pembagian beban secara ekonomis dengan metode *Lagrange* dan metode *Iterasi Lambda***

	Metode <i>Lagrange</i>	Metode <i>Iterasi Lambda</i>	PowerGen
P <sub>1</sub>	393,2 MW	390,52 MW	393,2 MW
P <sub>2</sub>	334,6 MW	332,47 MW	334,6 MW
P <sub>3</sub>	122,2 MW	121,37 MW	122,2 MW

```

E:\KU9037-1\TEORIP-1\POWERG-1.WOL\POWERG-1.WOL\SOFTWA-1\EDC.EXE
lambda = 9.1199 totalgen = 830.7
lambda = 9.1458 totalgen = 848.3
lambda = 9.1588 totalgen = 857.2
lambda = 9.1523 totalgen = 852.7
lambda = 9.1491 totalgen = 850.5
lambda = 9.1474 totalgen = 849.4
lambda = 9.1482 totalgen = 850.0
lambda = 9.1486 totalgen = 850.3
lambda = 9.1484 totalgen = 850.1
lambda = 9.1483 totalgen = 850.1
lambda = 9.1483 totalgen = 850.0
lambda = 9.1483 totalgen = 850.0

generator output limit inc cost penalty fact operating cost
          mw          $/mwhr          $/hr
-----
UNIT1    393.2          9.1483          1.0000          3916.38
UNIT2    334.6          9.1483          1.0000          3153.86
UNIT3    122.2          9.1483          1.0000          1124.16
totals    850.0
lambda = 9.1483          8194.40
  
```

**Gambar 2.11 Hasil perhitungan dengan PowerGen dengan P<sub>load</sub> = 850 MW**

Dari hasil perhitungan pembagian pembebanan pada masing-masing pembangkit dengan metode *Lagrange* dan metode *Iterasi Lambda* dapat penulis simpulkan bahwa metode *Lagrange* merupakan metode optimasi yang paling sederhana, penyelesaian matematis yang mudah dan proses perhitungannya yang lebih cepat. Pada metode *iterasi lambda* cara penyelesaiannya mirip dengan metode *Lagrange*, hanya harga  $\lambda$  dicari dengan iterasi sampai kendala  $\text{Error} = e = P_{\text{load}} - \sum_{i=1}^N P_i$ . Akibat cara ini maka perhitungan akan memakan waktu yang lama dan tergantung dari pemilihan harga mula  $\lambda$ . Disamping itu pula hasil perhitungan optimasi dengan metode *Lagrange* memiliki nilai yang akurat jika *dicompare* dengan *software* POWERGEN. Sehingga dalam Skripsi ini penulis menggunakan metode *Lagrange* sebagai metode optimasi.



### 2.4.2 *Unit Commitment*

Permintaan suplai daya listrik selalu berubah sesuai dengan kebutuhan pelanggan. Apabila terdapat sejumlah unit pembangkit dalam suatu sistem yang digunakan untuk memenuhi kebutuhan beban listrik, harus ada pembangkit yang dinyalakan dan dimatikan sesuai kebutuhan beban listrik yang harus disuplai agar pembangkitan bersifat ekonomis (Wood and Wollenberg, 1996:131).

Beberapa batasan yang dapat dijadikan pertimbangan dalam permasalahan *Unit Commitment* yaitu :

- a. *Spinning reserve* (cadangan daya), yaitu jumlah daya tersedia yang dapat dibangkitkan oleh semua unit yang berjalan pada sebuah sistem dikurangi dengan beban yang disuplai dan rugi-rugi daya. Cadangan yang harus diperhatikan untuk menjaga frekuensi sistem apabila ada satu unit atau lebih yang jatuh. Jika sebuah unit jatuh, harus ada cadangan daya yang cukup dari unit lain untuk menutupi kekurangan daya dalam waktu yang bersamaan.
- b. Batasan-batasan unit termis, yaitu :
  1. *Minimum up-time*, yaitu jika sekali sebuah unit dinyalakan, unit tersebut tidak boleh dimatikan sebelum melebihi waktu minimum penyalaan yang telah ditentukan.
  2. *Minimum down-time*, yaitu jika sekali sebuah unit dimatikan, unit tersebut tidak boleh dimatikan sebelum melebihi waktu minimum pematian yang telah ditentukan.

3. Batasan pegawai, yaitu jika sebuah pembangkit terdiri atas dua unit atau lebih, mereka tidak dapat dinyalakan bersamaan karena adanya keterbatasan pegawai untuk menangani kedua unit saat start-up.

Untuk menyalakan sebuah unit, suhu dan tekanan harus ditingkatkan secara perlahan, sejumlah energi akan dilepaskan untuk menjadikan unit *on-line*. Energi yang dilepaskan tidak menghasilkan keluaran daya MW dan di dalam pembangkitan disebut sebagai biaya *start-up*.

Kondisi unit saat dimatikan digolongkan menjadi 2, yaitu *cooling* (unit dimatikan sama sekali dan dipanaskan kembali saat akan tiba jadwal operasinya) dan *banking* (unit dipertahankan pada suhu tertentu dengan memasukkan sejumlah energi ke boiler sehingga saat tiba jadwal operasinya, unit dapat langsung bekerja tanpa melalui proses pemanasan lagi).

Probabilitas kombinasi yang harus dicoba untuk setiap keadaan beban dalam menjalankan *Unit Commitment* dengan  $M$  periode dengan  $N$  unit pembangkit adalah

$$C(N,1) + C(N,2) + \dots + C(N,N-1) + C(N,N) = 2^N - 1 \quad (2.20)$$

dengan  $C(N, j)$  merupakan kombinasi sejumlah  $N$  pada keadaan  $j$  dalam suatu waktu yang dirumuskan

$$C(N, j) = \left[ \frac{N!}{(N-j)!j!} \right] \quad (2.21)$$

$$j! = 1 \times 2 \times 3 \times \dots \times j$$

Apabila *Unit Commitment* dijalankan untuk  $M$  periode, maka jumlah kombinasi yang dicoba adalah  $(2^N - 1)^M$ . Sebagai contoh, untuk  $M = 24$  dan  $N = 5, 10, 20$  dan  $40$  jumlah kombinasinya menjadi

N	$(2^N - 1)^{24}$
5	$6,2 \times 10^{35}$
10	$1,73 \times 10^{72}$
20	$3,12 \times 10^{144}$
40	(terlalu besar)

Karena jumlahnya sangat besar, maka dipergunakan sejumlah metode untuk menyelesaikan permasalahan ini, diantaranya adalah *Priority List*, *Dynamic Programming* dan *Lagrange Relaxation*.

#### 2.4.2.1 *Unit Commitment* menggunakan Metode *Priority List* (Daftar Prioritas)

*Priority List* (Daftar Prioritas) merupakan metode yang paling sederhana dalam *Unit Commitment* yang dilakukan dengan cara menyusun suatu prioritas untuk mematikan unit pembangkit. Prioritas ini disusun berdasarkan *full load average cost*, yaitu nilai dari biaya *incremental* pembangkit pada kapasitas maksimal yang diperoleh dengan cara mengalikan *heat rate* pada kapasitas maksimal dengan biaya bahan bakar. Unit-unit diurutkan dari unit dengan biaya produksi termurah hingga termahal yang selanjutnya akan dijadikan pertimbangan dalam pemilihan kombinasi unit yang dioperasikan. Unit dengan biaya produksi termurah akan menempati urutan pertama dan menjadi pilihan utama saat melayani beban listrik.

Skema daftar prioritas dibuat dalam algoritma pematian unit yang sederhana, yaitu :

- a. Saat beban berkurang, hitung kemampuan pembangkitan dengan mematikan unit berikutnya dalam daftar prioritas untuk mencatu permintaan beban. Jika suplai tidak mencukupi, lanjutkan operasi seperti sebelumnya, jika suplai cukup, lanjutkan ke langkah berikutnya.
- b. Hitung jumlah jam,  $H$ , sebelum unit tersebut akan dibutuhkan kembali. Asumsikan beban berkurang dan akan bertambah lagi beberapa jam kemudian.
- c. Jika  $H$  kurang dari waktu *shut-down* minimum unit, jaga komitmen dan lanjutkan ke langkah terakhir, jika tidak, lanjutkan ke langkah berikutnya.
- d. Hitung dua biaya. Biaya pertama adalah jumlah biaya produksi per jam untuk  $H$  berikutnya dengan unit dinyalakan, kemudian hitung kembali jumlah yang sama untuk keadaan unit dimatikan dan menyertakan biaya *start-up* untuk memilih kedua kondisi unit dalam keadaan *banking* atau *cooling*, pilih yang lebih murah. Jika terdapat kondisi yang cukup untuk mematikan unit, unit harus dimatikan, jika tidak, jaga unit dalam keadaan *banking*.
- e. Ulang seluruh prosedur untuk unit berikutnya dalam daftar prioritas. Jika unit berikutnya juga dimatikan, lanjutkan ke unit berikutnya dan seterusnya.

#### **2.4.2.2 Unit Commitment menggunakan Metode *Dynamic Programming***

*Dynamic Programming* merupakan suatu metode dalam *Unit Commitment* untuk mencari pilihan yang optimal diantara beberapa alternatif

yang bisa ditempuh. Penggunaan metode *Dynamic Programming* dilakukan untuk mencari alternatif yang optimum, berupa kombinasi unit pembangkit termis yang terbaik untuk melayani beban tertentu agar didapat biaya bahan bakar yang minimal (Marsudi, 2005:171)

Jika dalam sebuah sistem terdapat  $n$  unit pembangkit yang siap beroperasi dan akan dioperasikan menurut jalur sub sistem termis, biaya start-stop unit pembangkit untuk sementara tidak akan diperhitungkan dahulu, maka algoritma optimasi biaya bahan bakarnya menjadi seperti berikut :

- a. Bila  $n = 1$ , maka beban sistem akan diatasi oleh satu-satunya unit yang ada.
- b. Bila  $n = 2$ , maka besar kapasitas pembangkitan dapat dicari menggunakan kombinasi dari kedua unit tersebut agar dicapai biaya bahan bakar yang minimum.
- c. Bila ada unit ke-3, dengan kurva biaya bahan bakar diketahui, maka dengan cara seperti tersebut diatas, kurva biaya minimum yang sudah didapat digabungkan dengan kurva biaya unit ketiga untuk mendapatkan kurva biaya unit ketiga untuk mendapatkan kurva biaya minimum dengan unit ketiga dalam sistem untuk menghadapi berbagai nilai beban sistem.
- d. Hal serupa dilakukan untuk unit ke-4 dan seterusnya sampai dengan unit ke- $n$ .

Di dalam *Dynamic Programming* terdapat *state* ( $N$ ) dan *stage* ( $X$ ), *state* adalah kemungkinan penyelesaian dengan kondisi tertentu sedangkan *stage* adalah kumpulan dari *state* pada level tertentu. Pada permasalahan *unit commitment state* adalah kombinasi *on/off* unit pembangkit sedangkan *stage* adalah periode

pembebanan. Dalam *Dynamic Programming*, jumlah maksimum nilai N adalah  $2^n - 1$ .

## 2.5 Metode Kuadrat Terkecil

Sebuah medan data sering dipenuhi oleh data-data yang tidak beraturan. Meskipun semua kontrol parameter tetap konstan, hasil yang diperoleh bervariasi. Sebuah proses secara kuantitatif yang menghitung bentuk perolehan yang disebut regresi atau *fitting* (pendekatan) kurva menjadi sangat penting. Pendekatan kurva terbaik dapat diperoleh dengan metode kuadrat terkecil (Efunda, 2008).

Prinsip teori regresi adalah bahwa setiap fungsi  $y = f(x)$  dapat dimasukkan ke pasangan titik-titik  $(x_1, y_1)$ ,  $(x_2, y_2)$  untuk meminimumkan jumlah kuadrat kesalahan pada setiap titik-titik, dalam hal ini (Pabla, 1994:18):

$$\sum_{i=1}^n \{y_i - f(x_i)\}^2 = \text{minimum} \quad (2.22)$$

Jumlah kuadrat kesalahan-kesalahan dipakai bila hal itu memberikan “kecocokan yang baik”.

### 2.5.1 Garis Kuadrat Terkecil (*Least Square Line*)

Kuadrat terkecil garis menggunakan bentuk persamaan garis lurus

$$y = a + bx \quad (2.23)$$

untuk menghitung sejumlah data  $(x_1, y_1)$ ,  $(x_2, y_2)$ , .....  $(x_n, y_n)$ , dimana  $n \geq 2$ .

Pendekatan kurva terbaik  $f(x)$  memiliki kuadrat terkecil dengan

$$\Pi = \sum_{i=1}^n [y_i - f(x_i)]^2 = \sum_{i=1}^n [y_i - (a + bx_i)]^2 = \text{min.} \quad (2.24)$$

dengan  $a$  dan  $b$  merupakan koefisien yang tidak diketahui, sementara semua nilai  $x_i$  dan  $y_i$  telah tersedia. Untuk memperoleh kesalahan kuadrat terkecil, koefisien  $a$  dan  $b$  harus diturunkan dan harus bernilai sama dengan nol.

$$\begin{cases} \frac{\partial \Pi}{\partial a} = 2 \sum_{i=1}^n [y_i - (a + bx_i)] = 0 \\ \frac{\partial \Pi}{\partial b} = 2 \sum_{i=1}^n x_i [y_i - (a + bx_i)] = 0 \end{cases} \quad (2.25)$$

Dengan mengembangkan persamaan diatas, persamaan yang diperoleh menjadi

$$\begin{cases} \sum_{i=1}^n y_i = a \sum_{i=1}^n 1 + b \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i y_i = a \sum_{i=1}^n x_i + b \sum_{i=1}^n x_i^2 \end{cases} \quad (2.26)$$

Koefisien  $a$  dan  $b$  dapat diperoleh dari

$$\begin{cases} a = \frac{(\sum y)(\sum x^2) - (\sum x)(\sum xy)}{n \sum x^2 - (\sum x)^2} \\ b = \frac{n \sum xy - (\sum x)(\sum y)}{n \sum x^2 - (\sum x)^2} \end{cases} \quad (2.27)$$

Dengan  $\sum \dots$  terdiri atas  $\sum_{i=1}^n \dots$

### 2.5.2 Parabola Kuadrat Terkecil (*Least Square Parabola*)

Kuadrat terkecil parabola menggunakan kurva dengan derajat kedua

$$y = a + bx + cx^2 \quad (2.28)$$

untuk menghitung sejumlah data  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ , dimana  $n \geq 3$ .

Pendekatan kurva terbaik  $f(x)$  memiliki kuadrat terkecil dengan

$$\Pi = \sum_{i=1}^n [y_i - f(x_i)]^2 = \sum_{i=1}^n [y_i - (a + bx_i + cx_i^2)]^2 = \min. \quad (2.29)$$

dengan  $a$ ,  $b$  dan  $c$  merupakan koefisien yang tidak diketahui, sementara semua nilai  $x_i$  dan  $y_i$  telah tersedia. Untuk memperoleh kesalahan kuadrat terkecil, koefisien  $a$  dan  $b$  harus diturunkan dan harus bernilai sama dengan nol.

$$\begin{cases} \frac{\partial \Pi}{\partial a} = 2 \sum_{i=1}^n [y_i - (a + bx_i + cx_i^2)] = 0 \\ \frac{\partial \Pi}{\partial b} = 2 \sum_{i=1}^n x_i [y_i - (a + bx_i + cx_i^2)] = 0 \\ \frac{\partial \Pi}{\partial c} = 2 \sum_{i=1}^n x_i^2 [y_i - (a + bx_i + cx_i^2)] = 0 \end{cases} \quad (2.30)$$

Dengan mengembangkan persamaan diatas, persamaan yang diperoleh menjadi

$$\begin{cases} \sum_{i=1}^n y_i = a \sum_{i=1}^n 1 + b \sum_{i=1}^n x_i + c \sum_{i=1}^n x_i^2 \\ \sum_{i=1}^n x_i y_i = a \sum_{i=1}^n x_i + b \sum_{i=1}^n x_i^2 + c \sum_{i=1}^n x_i^3 \\ \sum_{i=1}^n x_i^2 y_i = a \sum_{i=1}^n x_i^2 + b \sum_{i=1}^n x_i^3 + c \sum_{i=1}^n x_i^4 \end{cases} \quad (2.31)$$

Koefisien  $a$  dan  $b$  dapat ditemukan dengan menyelesaikan persamaan diatas.



## 2.6 JavaFX 2

Pada konferensi JavaOne 2007, Sun Microsystems memperkenalkan platform JavaFX untuk membantu pengembangan konten dan pengembangan aplikasi dalam membuat aplikasi RIA untuk perangkat mobile, desktop, televisi, dan perangkat konsumen lainnya. Awalnya terdiri dari platform JavaFX Handphone dan bahasa JavaFX Script. Beberapa rilis publik disampaikan setelah pengumuman awal, versi 1.3 dirilis pada 22 April 2010. Setelah akuisisi Oracle dari Sun Microsystems, Oracle mengumumkan pada konferensi JavaOne 2010 bahwa dukungan untuk bahasa JavaFX Script akan dihentikan. Namun, itu juga mengumumkan bahwa JavaFX Script API akan dikembalikan kepada bahasa Java seutuhnya dan akan dirilis sebagai bagian dari produk JavaFX 2.0. Pengumuman ini berarti bahwa kemampuan JavaFX akan tersedia untuk semua pengembang Java, tanpa perlu bagi mereka untuk belajar bahasa scripting baru. Dengan pengumuman ini, Oracle telah berkomitmen untuk membuat JavaFX sebagai lingkungan pembuat aplikasi RIA yang utama (Carl Dea, 2005).

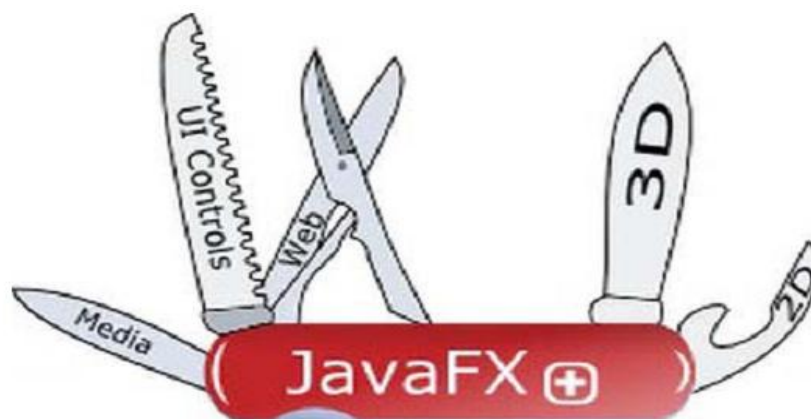
JavaFX masih satu keluarga dengan Java. JavaFX sendiri merupakan sebuah keluarga yang memiliki anggota keluarga JavaFX Mobile dan JavaFX Script dan semua itu berjalan diatas lingkungan *runtime* Java.

Java memang sudah populer sebelumnya, dengan kemampuannya yang dapat di terapkan pada berbagai perangkat dan mampu menghubungkan antar perangkat itu dengan mudah karena masih satu lingkungan *runtime*. Sedangkan JavaFX merupakan pengembangan dari teknologi Java yang memperluas kemampuan dan fitur dari teknologi Java itu sendiri. Dan JavaFX Script adalah

pemeran utama yang memainkan peranan penting dalam pengembangan aplikasi menggunakan JavaFX ini.

JavaFX dirancang untuk menyediakan para developer java sebuah platform yang baru, ringan, dengan performa yang tinggi. Tujuannya adalah ingin menggantikan swing untuk membangun aplikasi GUI. Namun itu bukan berarti Swing sudah tidak digunakan lagi. Sejumlah besar aplikasi telah dibangun dengan menggunakan swing itu berarti, Swing telah menjadi bagian dari Java API untuk waktu yang cukup lama. Terutama bahwa aplikasi ini bisa digabungkan dengan JavaFX secara fungsi.

JavaFX API berisi apa saja yang kita inginkan untuk membuat platform grafis, – UI control, animasi, efek dan lain – lain. Sebuah user interface masih dibangun dengan deretan layer yang terdiri dari 'scene graph'. 'Scene graph' ditampilkan diatas top level container yang disebut stage. Sang perancang javaFX API rupanya memodelkan secara analogis mirip sebuah pertunjukan theater atau sebuah permainan dimana aktor memerankan sebuah adegan didepan penonton.



**Gambar 2.12 JavaFX**

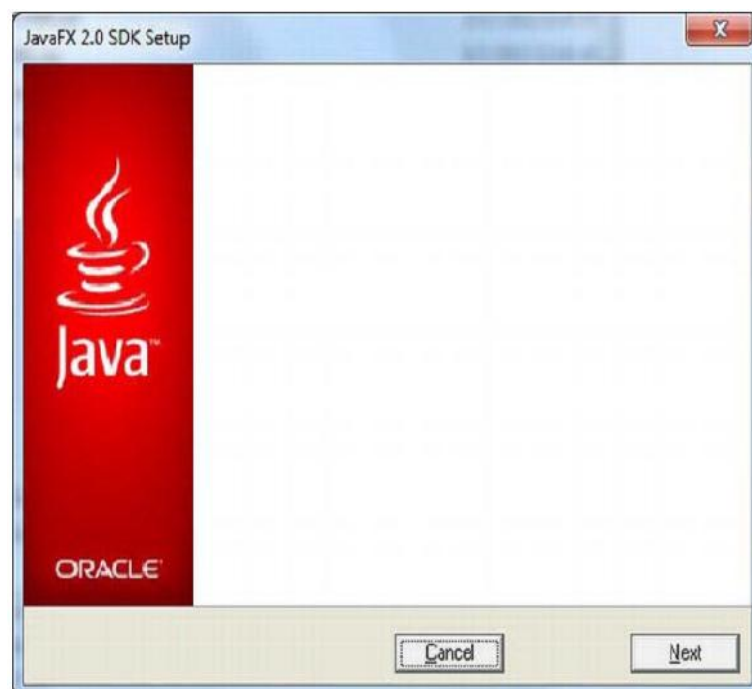
### 2.6.1 Instalasi JavaFX 2

Untuk proses instalasi anda memerlukan beberapa *software* berikut :

1. JDK 7 ke atas
2. JavaFx SDK (dapat anda download di situs resmi oracle)
3. Editor java (netbeans, eclipse, dsb)

Proses instalasi :

1. Download JavaFX 2.0SDK dan NetBeans IDE 7.1.x dari lokasi :
  - JavaFX 2.0SDK  
<http://www.oracle.com/technetwork/java/javafx/download/index.html>
  - NetBeans 7.1 beta SDK: <http://netbeans.org>
2. Setelah anda download JavaFX 2.0SDK, anda dapat mulai menginstalnya berikut gambar tahap-tahap penginstalan.



**Gambar 2.13 JavaFX 2.0 SDK Setup Wizard**

3. Klik next, tentukan *home directory* dari JavaFX.



**Gambar 2.14 JavaFX SDK home directory**

4. Pilih folder instalasi, jika sudah klik next. Tunggu sampai proses selesai.



**Gambar 2.15 Completing the install**

Setelah selesai, anda dapat mengatur *classpath* (pada windows) sebagai berikut:

```
set JAVAFOX_HOME=C:\Program Files (x86)\Oracle\JavaFX 2.0 SDK
```

```
set JAVA_HOME=C:\Program Files (x86)\Java\jdk1.7.0
```

```
set CLASSPATH=%JAVAFOX_HOME%\rt\lib\jfxrt.jar;
```

Proses instalasi selesai, dan anda sudah dapat menjalankan atau membuat project JavaFX (khusus editor netbeans).

Jika editor anda eclipse, anda dapat mendownload beberapa plugin yang telah tersedia di *marketplace* eclipse dan mulai membuat suatu project.

## 2.7 Penelitian Sebelumnya

Dasar atau acuan yang berupa teori-teori atau temuan-temuan melalui hasil berbagai penelitian sebelumnya merupakan hal yang sangat perlu dan dapat dijadikan sebagai data pendukung. Salah satu data pendukung yang menurut penulis perlu dijadikan bagian tersendiri adalah penelitian terdahulu yang relevan dengan permasalahan yang sedang dibahas dalam penelitian ini. Dalam hal ini, fokus penelitian terdahulu yang dijadikan acuan adalah terkait dengan masalah pembebanan ekonomis. Oleh karena itu, penulis melakukan langkah kajian terhadap beberapa hasil penelitian berupa Skripsi dan jurnal-jurnal melalui internet.

Beberapa penelitian sebelumnya antara lain sebagai berikut :

1. Perancangan *prototype* sistem manajemen energi dengan menggunakan mikrokontroller untuk optimasi pembangkitan tenaga listrik oleh Sumariyono pada tahun 2010, Program Studi Teknik Elektro, Universitas Muhammadiyah Gresik. Dimana penelitian tersebut didiskusikan perancangan suatu *prototype* untuk sistem manajemen energi model *Unit Commitment* metode *Priority List* dan *Economic Dispatch* metode

*Lagrangian* berbasis mikrokontroler dengan menggunakan bahasa C++ sebagai bahasa pemrogramannya.

2. Optimalisasi biaya bahan bakar untuk penjadwalan unit-unit pembangkit thermal sistem tenaga dengan metode iterasi lambda oleh Sartika Veronika Angdrie pada tahun 2010, Program Studi Teknik Elektro, UNSRAT. Dari penelitian yang dilakukan memberikan hasil atau gambaran pembagian beban dengan menggunakan iterasi lambda memberikan hasil yang lebih optimal dibandingkan dengan total biaya yang dikeluarkan oleh PT. PLN (Persero) wilayah Sulutenggo.

Dari beberapa contoh hasil penelitian di atas, maka dapat digambarkan beberapa persamaan dan perbedaannya. Persamaan Skripsi ini dengan hasil-hasil penelitian sebelumnya adalah pada salah satu metode yang digunakan dalam membahas pokok permasalahan, yaitu *economic dispatch* dengan metode *Lagrange* dan *unit commitment* dengan metode *Priority List*.

Sedangkan perbedaan antara Skripsi ini dengan hasil-hasil penelitian sebelumnya adalah objek penelitian adalah PLTGU, maka langkah-langkah optimasi yang harus ditempuh akan terjadi suatu perbedaan dalam pembentukan karakteristik pembangkitan dan menjadi lebih kompleks dan unik karena PLTGU termasuk jenis pembangkit *combined cycle*. Optimasi dilakukan dengan model *Economic Dispatch* dengan metode *Lagrange* yang dilakukan untuk membagi beban secara ekonomis pada tiap blok pembangkit. Dan optimasi juga dilakukan dengan model *Unit Commitment* yang menggabungkan metode pencarian solusi *Priority List* dan *Dynamic Programming* dengan objek berupa masukan bahan

bakar dan daya keluaran unit pembangkit. Dimana perhitungan pembebanan ekonomis akan dilakukan dengan menggunakan *software* JAVA FX 2.