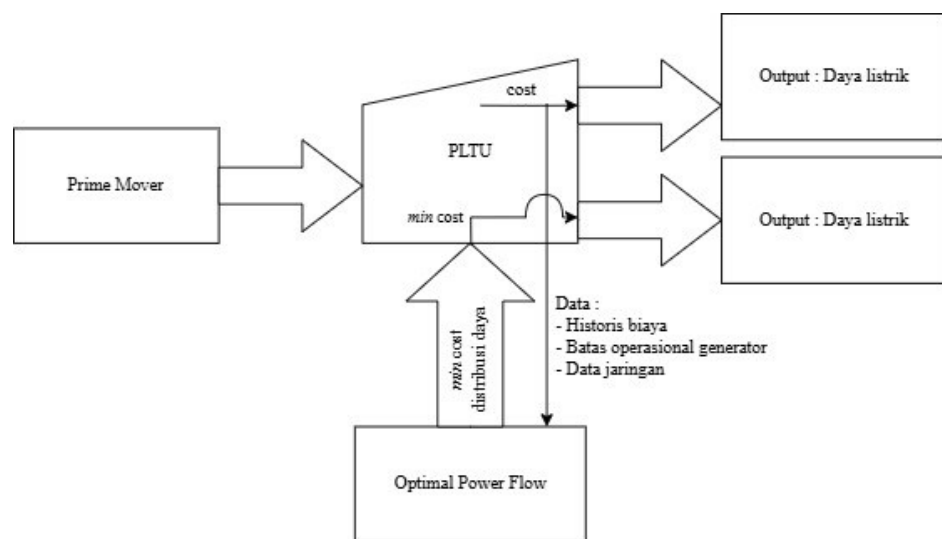


BAB III

METODE PENELITIAN

3.1 Desain Penelitian

Desain pada penelitian ini dengan judul “Penggunaan Metode *Optimal Power Flow* (OPF) untuk Pembangkitan Tenaga Listrik pada PLTU Nusantara Power Indramayu Berbasis IEEE 9-Bus” dapat dijelaskan dengan diagram sebagai berikut.



Gambar 3. 1 Diagram Desain Penelitian

Penelitian ini diawali dengan pengambilan data yang diperlukan untuk kebutuhan optimasi dengan OPF. Data yang didapatkan meliputi data biaya pembangkitan, data batas operasional generator, dan data jaringan yang nantinya akan di-*input* pada program komputasi menggunakan *software* MATLAB. Pada kasus data biaya pembangkitan, dilakukan regresi kuadratik terlebih dahulu dengan bantuan Microsoft Excel demi mendapatkan fungsi biaya yang akan digunakan sebagai fungsi objektif berbentuk fungsi kuadratik yang nantinya akan dioptimasi dalam sistem, data batas operasional generator digunakan sebagai fungsi batasan dalam OPF, sedangkan data jaringan yang dipakai yaitu data dari jaringan standar IEEE 9-Bus. Optimasi menggunakan OPF dilakukan melalui program MATLAB dengan program *dispatch*-nya sebagai algoritma optimasi utama. Hasil optimasi dengan OPF berupa

distribusi daya tiap generator serta biaya pembangkitan yang nantinya akan dibandingkan dengan kondisi awal sebelum optimasi.

3.2 Partisipan dan Lokasi Penelitian

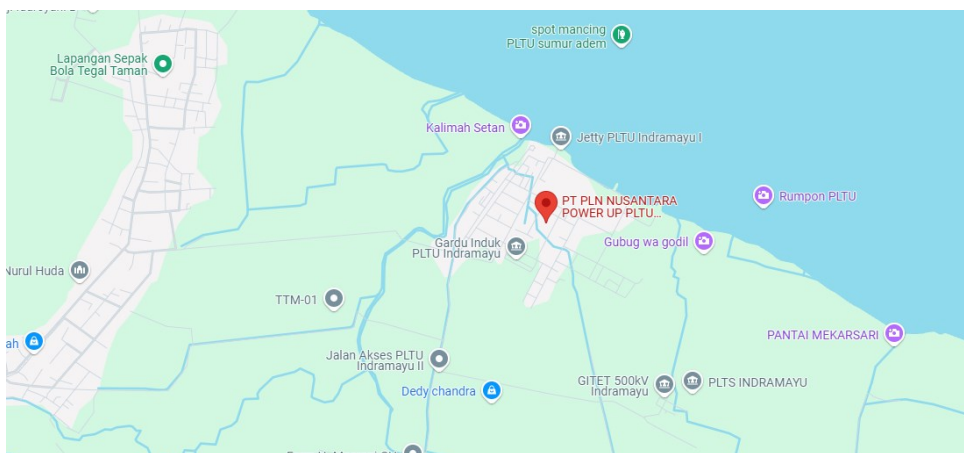
Penelitian kali ini melibatkan partisipan dan juga lokasi penelitian, partisipan yang terlibat merupakan individu dari beberapa divisi PLTU yang berkaitan dengan topik penelitian, sedangkan lokasi penelitian dipilih berdasarkan kriteria topik penelitian.

3.2.1 Partisipan Penelitian

Partisipan penelitian dalam penelitian ini ialah staff sub bidang enjiniring PT PLN Nusantara Power UP Indramayu, yang memberikan data-data penelitian berupa data spesifikasi generator, data total biaya operasi mencakup biaya bahan bakar dalam satu bulan, dan data permasalahan/gangguan pada generator. Selain itu, partisipan lainnya yaitu staff sub bidang operasi bagian perencanaan dan pengendalian operasi yang memberikan data historis operasi generator dalam tujuh jam.

3.2.2 Lokasi Penelitian

Lokasi penelitian yaitu PT PLN Nusantara Power UP Indramayu yang beralamat lengkap di Jl. Raya Sumuradem, Kecamatan Sukra, Kabupaten Indramayu, Jawa Barat 45257. Lokasi kantor PT PLN Nusantara Power UP Indramayu berada tepat di kawasan unit pembangkitannya yang dapat dilihat dari gambar berikut.

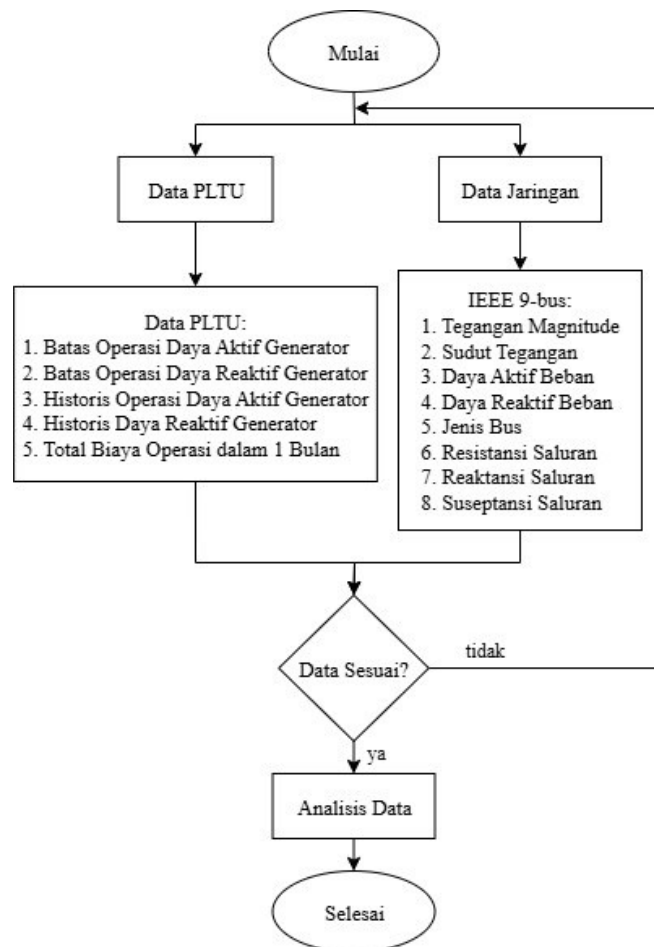


Gambar 3. 2 Lokasi PT PLN Nusantara Power UP Indramayu

(Sumber: <https://maps.app.goo.gl/GxWoGQeYHNL2QJjQA>)

3.3 Instrumen Penelitian

Instrumen penelitian ini mencakup dua sumber data yang dibutuhkan, yaitu data PLTU dan data jaringan. Data PLTU diperoleh dari PT PLN Nusantara Power UP Indramayu. Sedangkan, data jaringan diperoleh dari data jaringan standar IEEE 9-Bus yang merepresentasikan seolah-olah jaringan yang disuplai oleh PLTU Indramayu, keterbatasan tersebut berkaitan dengan akses data yang agak sukar terkait data jaringan aktual. Meskipun begitu, data jaringan IEEE 9-Bus ini tetap dapat digunakan untuk penelitian OPF ini dengan terintegrasi PLTU Indramayu karena jaringan ini merupakan objek yang sering digunakan untuk keperluan penelitian termasuk dalam bidang analisis aliran daya (Mohan & Go, 2020)(Asri, 2023)(Asija et al., 2015) walaupun karakteristik jaringannya belum tentu sesuai dengan karakteristik jaringan aktual. Berikut merupakan diagram alir dari instrumen penelitian.



Gambar 3. 3 Flowchart Proses Pengumpulan Data

Diagram alir tersebut dapat dijelaskan dengan sebagai berikut.

1. Data PLTU

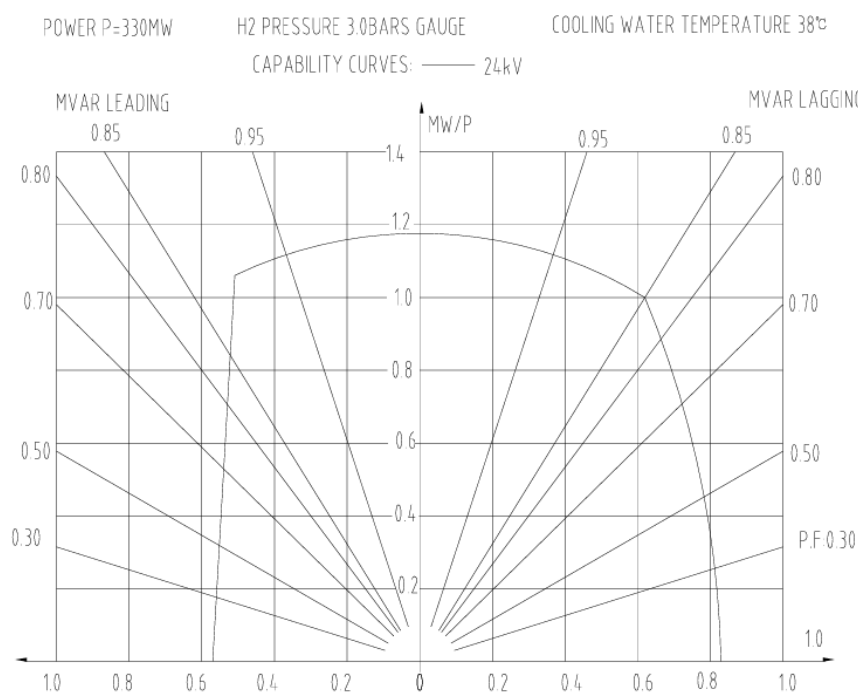
Data yang diperoleh dari PLTU Indramayu yang dibutuhkan dalam simulasi OPF adalah sebagai berikut.

a. Spesifikasi generator

Spesifikasi generator terdiri dari data-data karakteristik generator itu sendiri termasuk kurva kapabilitas generator.

1.2 General characteristic of generator		—Cooling of magnetic core H ₂	
1.2.1 Rated parameters		—Cooling of stator winding H ₂ O	
Type	255-460	—Cooling of rotor winding H ₂	
Rated speed	3000r/min	—Cooling of high voltage terminals H ₂ O	
Active power	330MW	Normal duty conditions:	
Power factor	0.85	—Stator water temperature rise 18K	
Apparent power	388.2MVA	—Rotor winding temperature rise 48K	
Rated voltage	24kV	Hydrogen pressure 0.3MPa(g)	
Voltage variation	±5%	Deionized water temperature 46 °C	
Rated current	9339A	Power 330MW	
Number of poles	2	Abnormal duty conditions:	
Frequency	50Hz	—With one cooler out of operation	
Excitation current	2495A	258.8MVA (220M	
Insulation class	F (Limited to class B)	W)	
Water temperature at inlet of H ₂ Cooler	≤38 °C	Steady state negative phase-sequence current I_2/I_s	
Type of cooling:		12%	
		Transient negative phase-sequence current $(I_2/I_k)^2 t$	
		10s	

Gambar 3. 4 Spesifikasi Umum Generator (PLTU Indramayu, 2025)



Gambar 3. 5 Kurva Kapabilitas Generator (PLTU Indramayu, 2025)

Muhammad Syamsiar, 2025

PENGUNAAN METODE OPTIMAL POWER FLOW (OPF) UNTUK OPTIMALISASI PEMBANGKITAN TENAGA LISTRIK PADA PLTU NUSANTARA POWER INDRAMAYU BERBASIS IEEE 9-BUS

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

b. Data historis operasi generator

Data historis operasi generator melibatkan 420 rekapitulasi data selama tujuh jam operasi generator. Alasan penggunaan data yang terbatas pada operasi selama tujuh jam saja berkaitan dengan keterbatasan/kesulitan akses data operasi dengan rentang waktu yang lebih panjang akibat keterbatasan izin akses serta kerahasiaan operasional. Berikut merupakan data operasional generator selama tujuh jam yang mewakili 420 data yang diberikan pihak PLTU Indramayu.

Tabel 3. 1 Data Historis Pembangkitan PLTU (PLTU Indramayu, 2025)

Waktu	P_{gen}	I_{exc}	V_{exc}	I_R	I_S	I_T	Q_{gen}
21:00:43	297,07	1901,95	348,67	7352,34	7536,56	7074,38	58,4
22:00:44	299,21	1901,78	349,41	7403,2	7585,55	7124,3	57,48
23:00:45	241,14	1710,7	309,02	5988,05	6169,22	5735,63	52,65
00:00:56	102,21	1310,45	229,34	2580,47	2775,47	2443,59	34,8
01:00:48	10,6	1114,98	191,41	373,36	372,42	274,06	12,61
02:00:48	0,3	30,23	2,58	23,44	22,03	19,69	1,11
03:00:49	0,3	29,71	2,42	23,44	22,03	19,69	1,13
03:59:51	0,3	29	2,34	23,44	21,8	19,45	1,11

c. Total biaya operasi dalam satu bulan

Total biaya operasi yang diperoleh akan dialokasikan terhadap data historis operasi daya aktif dalam tujuh jam untuk keperluan regresi kuadratik fungsi objektif.

V. BIAYA OPERASI DAN PEMELIHARAAN

Biaya operasi berdasarkan pemakaian bahan bakar, kimia dan minyak pelumas pada bulan April 2025 adalah sebagai berikut :

- Biaya pemakaian Bahan Bakar <i>Batubara</i>	Rp. :	237,613,131,700
- Biaya pemakaian Bahan Bakar <i>Biomassa</i>	Rp. :	5,118,067,737
- Biaya pemakaian Bahan Bakar <i>Minyak HSD</i>	Rp. :	1,880,294,081
- Biaya pemakaian Bahan <i>Kimia</i>	Rp. :	494,914,457
- Biaya pemakaian <i>Pelumas</i>	Rp. :	366,477,801
Total Biaya Operasi	Rp. :	245,472,885,776

Penjualan Energi Listrik ke PT. PLN (Persero) UIP2B : 355,021,448.00 kWh

Biaya Operasi per kWh (untuk Pemakaian Batubara)	:	669.29	Rp/kWh
Biaya Operasi per kWh (untuk Pemakaian Biomassa)	:	14.42	Rp/kWh
Biaya Operasi per kWh (untuk Pemakaian HSD)	:	5.30	Rp/kWh
Biaya Operasi per kWh (untuk Pemakaian Kimia)	:	1.39	Rp/kWh
Biaya Operasi per kWh (untuk Pemakaian Pelumas)	:	1.03	Rp/kWh
Biaya Operasi per kWh Penjualan Energi	:	691.43	Rp/kWh

Gambar 3. 6 Data Total Biaya Operasi (PLTU Indramayu, 2025)

Muhammad Syamsiar, 2025

PENGUNAAN METODE OPTIMAL POWER FLOW (OPF) UNTUK OPTIMALISASI PEMBANGKITAN TENAGA LISTRIK PADA PLTU NUSANTARA POWER INDRAMAYU BERBASIS IEEE 9-BUS

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

2. Data Jaringan

Data jaringan standar IEEE 9-Bus diperoleh dari *Toolbox* MATPOWER yang secara *open source* menyertakan data IEEE 9-Bus dalam format *file* (.m) MATLAB. Data yang dibutuhkan dari jaringan adalah sebagai berikut.

a. Data bus

Data bus didapatkan dari matriks *bus data* pada kode program *case9* yang berisi informasi-informasi pada bus dalam jaringan standar IEEE 9-Bus pada *software* MATLAB.

```

44 mpc.baseMVA = 100;
45
26 %% bus data
27 % bus i type Pd Qd Gs Bs area Vm Va baseKV zone Vmax Vmin
28 mpc.bus = [
29 1 3 0 0 0 0 1 1 0 345 1 1.1 0.9;
30 2 2 0 0 0 0 1 1 0 345 1 1.1 0.9;
31 3 2 0 0 0 0 1 1 0 345 1 1.1 0.9;
32 4 1 0 0 0 0 1 1 0 345 1 1.1 0.9;
33 5 1 90 30 0 0 1 1 0 345 1 1.1 0.9;
34 6 1 0 0 0 0 1 1 0 345 1 1.1 0.9;
35 7 1 100 35 0 0 1 1 0 345 1 1.1 0.9;
36 8 1 0 0 0 0 1 1 0 345 1 1.1 0.9;
37 9 1 125 50 0 0 1 1 0 345 1 1.1 0.9;
38 ];
39
40 %% generator data
41 % bus Pg Qg Qmax Qmin Vg mBase status Pmax Pmin Pc1 Pc2 Qc1min Qc1max Qc2min Qc2max ramp_agc ramp
42 mpc.gen = [
43 1 72.3 27.03 300 -300 1.04 100 1 250 10 0 0 0 0 0 0 0 0;
44 2 163 6.54 300 -300 1.025 100 1 300 10 0 0 0 0 0 0 0 0;
45 3 85 -10.95 300 -300 1.025 100 1 270 10 0 0 0 0 0 0 0 0;
46 ];
47
48 %% branch data
49 % fbus tbus r x b rateA rateB rateC ratio angle status angmin angmax
50

```

Gambar 3. 7 Data Bus Jaringan IEEE 9-Bus

Keterangan:

- Kolom 1 (ungu): Nomor bus.
- Kolom 2 (kuning): Tipe bus, 1 = PQ bus (beban), 2 = PV bus, 3 = *Slack* bus.
- Kolom 3 (hijau): Daya aktif beban.
- Kolom 4 (biru): Daya reaktif beban.
- Kolom 8 (jingga): Tegangan *magnitude* bus.
- Kolom 9 (merah): Sudut tegangan bus.

b. Data saluran

Data saluran diperoleh dari matriks *branch data* pada kode program *case9* yang merepresentasikan saluran antar bus pada jaringan standar IEEE 9-Bus pada *software* MATLAB.

```

case9.m
40 %% generator data
41 % bus Pg Qg Qmax Qmin Vg mBase status Pmax Pmin Pc1 Pc2 Qc1min Qc1max Qc2min Qc2max ramp_agc ramp
42 mpc.gen = [
43 1 72.3 27.03 300 -300 1.04 100 1 250 10 0 0 0 0 0 0 0 0;
44 2 163 6.54 300 -300 1.025 100 1 300 10 0 0 0 0 0 0 0 0;
45 3 85 -10.95 300 -300 1.025 100 1 270 10 0 0 0 0 0 0 0 0;
46 ];
47
48 %% branch data
49 % fbus tbus r x b rateA rateB rateC ratio angle status angmin angmax
50 mpc.branch = [
51 1 4 0 0.0576 0 250 250 250 0 0 1 -360 360;
52 4 5 0.017 0.092 0.158 250 250 250 0 0 1 -360 360;
53 5 6 0.039 0.17 0.358 150 150 150 0 0 1 -360 360;
54 3 6 0 0.0586 0 300 300 300 0 0 1 -360 360;
55 6 7 0.0119 0.1008 0.209 150 150 150 0 0 1 -360 360;
56 7 8 0.0085 0.072 0.149 250 250 250 0 0 1 -360 360;
57 8 2 0 0.0625 0 250 250 250 0 0 1 -360 360;
58 8 9 0.032 0.161 0.306 250 250 250 0 0 1 -360 360;
59 9 4 0.01 0.085 0.176 250 250 250 0 0 1 -360 360;
60 ];
61
62 %%----- OPF Data -----%%
63 %% generator cost data
64 % 1 startup shutdown n x1 y1 ... xn yn
65 % 2 startup shutdown n c(n-1) ... c0
66

```

Gambar 3. 8 Data Saluran Jaringan IEEE 9-Bus

Keterangan:

- Kolom 1 & 2 (ungu): Saluran (jalur antar bus).
- Kolom 3 (kuning): Resistansi saluran.
- Kolom 4 (hijau): Reaktansi saluran.
- Kolom 5 (biru): Suseptansi saluran.

3.3.1 Alat dan Bahan Penelitian

Alat yang dipakai untuk keperluan penelitian ini adalah sebagai berikut:

Tabel 3. 2 Alat-alat Penelitian

No.	Alat	Fungsi
1	Laptop	Perangkat penelitian dan simulasi
2	MATLAB	Software simulasi OPF
3	Microsoft Excel	Software alat bantu regresi fungsi biaya
4	Web Plot Digitizer	Web alat bantu ekstraksi kurva kapabilitas generator

Bahan yang diperlukan untuk penelitian ini adalah sebagai berikut:

1. Kode Program *Optimal Power Flow*

Kode program MATLAB yang digunakan dalam optimasi sistem tenaga listrik di PLTU Indramayu yang terintegrasi IEEE 9-Bus ini menggunakan kode program yang disusun dalam buku *Power System Analysis* yang disusun oleh Hadi Saadat (1999).

2. Fungsi Objektif

Fungsi objektif yang menjadi target optimasi utama dalam OPF yaitu biaya pembangkitan. Fungsi biaya yang digunakan dalam simulasi OPF akan dibentuk berdasarkan hasil pemodelan fungsi melalui regresi kuadratik dari alokasi biaya pembangkitan terhadap 420 data historis operasi daya aktif generator selama tujuh jam penarikan.

3. Batasan/*Constraints*

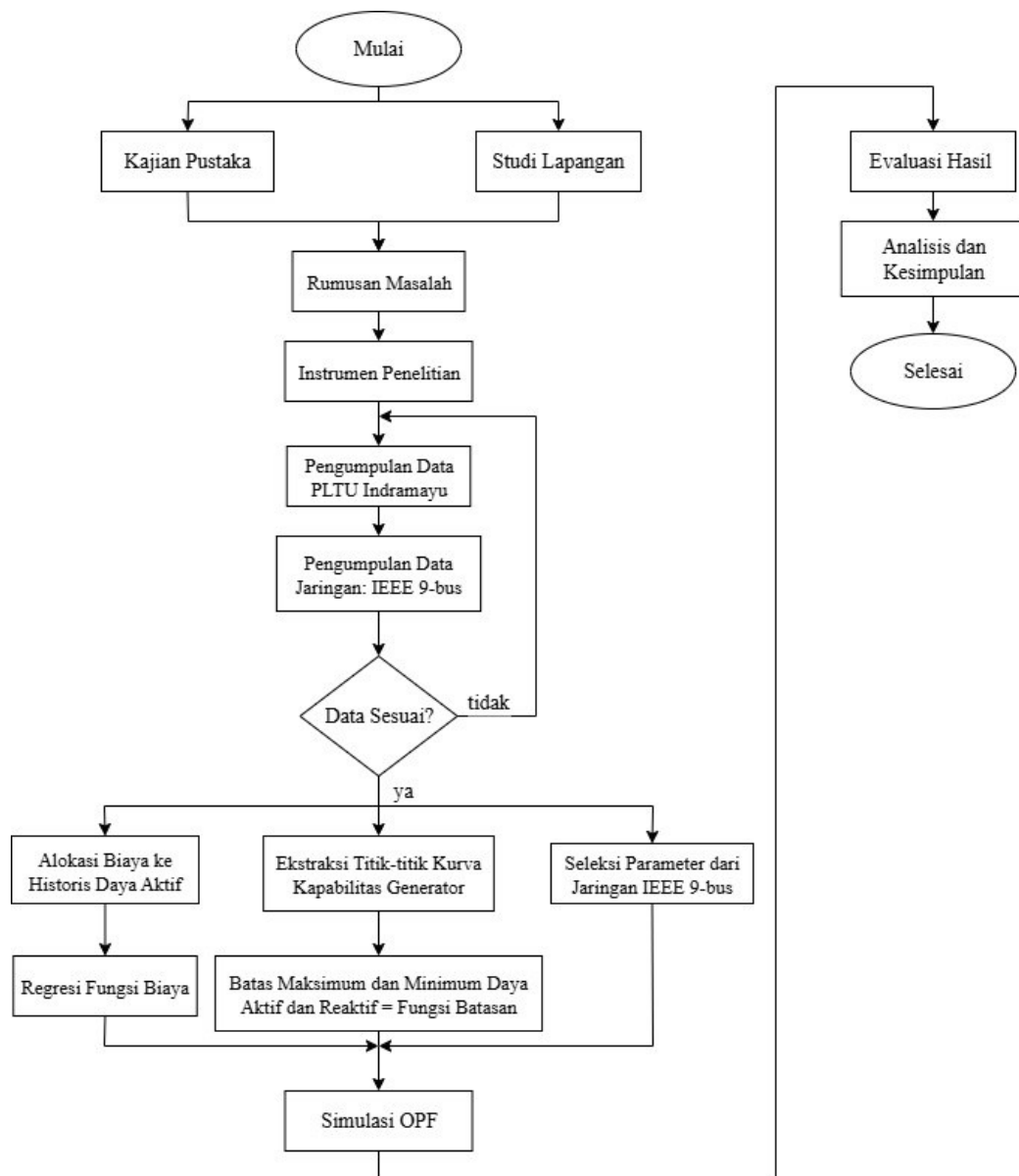
Batasan yang digunakan dalam simulasi OPF mencakup batas operasional daya aktif dan daya reaktif generator yang diperoleh dari ekstraksi titik-titik pada kurva kapabilitas generator dari spesifikasi generator menggunakan *web plot digitizer*. Melalui ekstraksi titik-titik ini, akan dilakukan perhitungan terkait berapa batas operasi maksimum dan minimum daya aktif dan daya reaktif yang dapat disuplai oleh generator tersebut.

4. Data Jaringan

Data jaringan yang digunakan adalah data jaringan standar IEEE 9-Bus dengan karakteristik jaringan yang diharapkan setidaknya hampir menyamai jaringan aktual PLTU Indramayu dengan tiga generatornya dan penggunaannya yang telah terverifikasi dalam penelitian dalam bidang analisis aliran daya secara luas. Penggunaan data jaringan sebagai pengganti ini didasarkan pada keterbatasan akses pada data jaringan aktual yang disuplai oleh pembangkitan tenaga listrik di Indramayu.

3.4 Prosedur Penelitian

Prosedur penelitian merupakan langkah-langkah dalam melakukan suatu penelitian. Dalam penelitian ini dengan judul “Penggunaan Metode *Optimal Power Flow* untuk Optimalisasi Pembangkitan Tenaga Listrik pada PLTU Nusantara Power Indramayu Berbasis IEEE 9-Bus” menggunakan *flowchart* sebagai penjelasan urutan/langkah-langkah penelitian yang diawali dengan kajian pustaka, studi lapangan hingga analisis dan kesimpulan. Prosedur penelitian pada penelitian ini akan dijelaskan dengan *flowchart* sebagai berikut.



Gambar 3. 9 *Flowchart* Prosedur Penelitian

Penelitian diawali dengan studi lapangan terkait dengan masalah yang terjadi pada proses pembangkitan di PLTU Indramayu, studi literatur dilakukan guna mencari referensi berupa artikel jurnal, buku, dsb, serta mempelajari topik OPF untuk mengkaji hingga memperdalam seluk beluk mengenai OPF itu sendiri. Selanjutnya, berdasarkan latar belakang yang telah dikaji, dibuatlah rumusan masalah dengan tujuan untuk memfokuskan penelitian agar pembahasan tidak jauh melebar dari tujuan.

Tahap selanjutnya yaitu menentukan instrumen penelitian berupa data-data yang harus dikumpulkan serta alat dan bahan yang harus disiapkan untuk penelitian. Pengumpulan data dimulai dengan mengumpulkan data-data PLTU Indramayu yang akan digunakan dalam OPF, kemudian pencarian data jaringan standar IEEE 9-Bus untuk melengkapi sistem OPF. Setelah data terkumpul, pengecekan data dilakukan guna menentukan apakah data-data yang telah terkumpulkan sudah lengkap dan sesuai. Setelah lengkap dan sesuai, pengolahan data-data dilakukan agar dapat segera di-*input* ke dalam kode program MATLAB. Data total biaya operasi akan dialokasikan terhadap data historis daya aktif selama tujuh jam, hasilnya akan dibentuk fungsi biaya dengan melakukan regresi kuadratik terhadap data historis daya aktif. Kurva kapabilitas generator yang diperoleh dari spesifikasi generator akan diekstraksi titik-titiknya sehingga dapat secara akurat diketahui batas operasi maksimum dan minimum dari pembangkitan daya aktif dan daya reaktif generator tersebut. Sementara itu, data jaringan standar IEEE 9-Bus akan diseleksi mana saja parameter yang akan digunakan dalam program simulasi utama OPF.

Setelah ketiga bahan utama simulasi diperoleh, langkah selanjutnya ialah *input* data lalu menjalankan program utama, hasil akan diperoleh dalam bentuk laporan pada jendela *command window* di aplikasi MATLAB. Simulasi dilakukan dua kali dengan simulasi pertama sesuai permintaan beban standar IEEE 9-Bus. Sedangkan simulasi kedua, menggunakan permintaan beban maksimum yang dapat disimulasikan fungsi biaya yang dimodelkan. Hasil yang telah muncul tersebut dapat dianalisis dengan membandingkannya dengan kondisi sebelum optimasi dilakukan sehingga dapat memberikan kesimpulan terkait penelitian yang sedang dilakukan.

3.5 Analisis Data Penelitian

Analisis data dimulai dengan melakukan pemodelan fungsi kuadratik melalui regresi kuadratik untuk memperoleh fungsi objektif yang menjadi komponen utama optimasi menggunakan OPF, komponen OPF lainnya yang perlu diperoleh yaitu fungsi batasan yang akan didapatkan melalui kurva kapabilitas generator. Pada data jaringan, hanya diperlukan seleksi data saja

dalam menyesuaikan parameter yang digunakan dalam kode program simulasi OPF.

3.5.1 Pemodelan Fungsi Biaya Pembangkitan

Berdasarkan data total biaya pembangkitan yang telah diperoleh dari PLTU Indramayu, akan dialokasikan terhadap 420 data historis operasi daya aktif pada generator selama tujuh jam.

$$Total\ Biaya\ Operasi = Rp245.472.885.776$$

$$Total\ Biaya\ Operasi\ per\ 7\ jam = \frac{Rp245.472.885.776}{30 \times 24} \times 7$$

$$= Rp2.386.541.945,04$$

$$= \$146.791,40 \text{ (konversi Rp ke USD per tanggal 5 Juni 2025)}$$

Lalu,

$$Total\ Daya\ Aktif = \sum_{i=1}^{420} P_i = P_1 + P_2 + \dots + P_{420}$$

$$= 297,07\ MW + 297,42\ MW + \dots + 0,3\ MW$$

$$= 44830,17\ MW$$

$$Biaya\ per\ Unit\ Daya\ Aktif = \frac{Total\ Biaya\ Pembangkitan\ (7jam)}{\sum_{i=1}^{420} P_i}$$

$$= \frac{\$146.791,40}{44830,17\ MW}$$

$$= 3,27\ \$/MW$$

$$Biaya\ tiap\ Data\ Daya\ Aktif = P_i \times Biaya\ per\ Unit\ Daya\ Aktif$$

Sehingga,

$$Data\ ke - 1 = 297,07\ MW \times 3,27\ \$/MW$$

$$= \$972,72$$

$$Data\ ke - 2 = 297,42\ MW \times 3,27\ \$/MW$$

$$= \$973,87$$

Dan seterusnya hingga data ke-420.

Setelah dialokasikan terhadap 420 data daya aktif generator, langkah selanjutnya yaitu pemodelan fungsi biaya menggunakan regresi kuadratik dengan persamaan regresi kuadratik sebagai berikut (Hidayanti et al., 2022).

$$y' = a + bx + cx^2$$

Untuk mencari setiap koefisien dari fungsi, akan dihitung berdasarkan matriks sebagai berikut.

$$\begin{bmatrix} n & \sum x & \sum x^2 \\ \sum x & \sum x^2 & \sum x^3 \\ \sum x^2 & \sum x^3 & \sum x^4 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum y \\ \sum xy \\ \sum x^2y \end{bmatrix}$$

$$a = \frac{\det 2}{\det 1}$$

$$b = \frac{\det 3}{\det 1}$$

$$c = \frac{\det 4}{\det 1}$$

Berdasarkan data yang telah diperoleh, akan didapatkan tabel perhitungan.

Tabel 3. 3 Data Perhitungan koefisien biaya

No.	$P_i = x$	$cost = y$	x^2	x^3	x^4	xy	x^2y
1	297,07	972,72	88250,58	26216601	7788165735	288966,74	85843350,8
2	297,42	973,87	33453,65	26309374	7824933892	289648,05	86147123,5
3	298,51	977,44	89108,22	26599695	7940274889	291774,98	87097748
4	299,31	980,05	89586,48	26814128	8025736700	293340,97	87799886,3
5	300,23	983,07	90138,05	27062148	8124868581	295147,05	88611998,5
...
420	0,3	0,982	0,09	0,027	0,0081	0,295	0,0884

Maka,

$$n = 420$$

$$\sum x = 44830,17$$

$$\sum y = 146791,42$$

$$\sum x^2 = 10965808,52$$

$$\sum x^3 = 2,91 \times 10^9$$

$$\sum x^4 = 8,03 \times 10^{11}$$

$$\sum xy = 35906322,81$$

$$\sum x^2y = 9533158224$$

Selanjutnya akan disusun matriks untuk menghitung determinan.

Matriks utama:

$$\begin{bmatrix} 420 & 44830,17 & 10965808,52 \\ 44830,17 & 10965808,52 & 2,91 \times 10^9 \\ 10965808,52 & 2,91 \times 10^9 & 8,03 \times 10^{11} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 146791,42 \\ 35906322,81 \\ 9533158224 \end{bmatrix}$$

Matriks 1:

$$\begin{vmatrix} 420 & 44830,17 & 10965808,52 \\ 44830,17 & 10965808,52 & 2,91 \times 10^9 \\ 10965808,52 & 2,91 \times 10^9 & 8,03 \times 10^{11} \end{vmatrix} \begin{vmatrix} 420 & 44830,17 \\ 44830,17 & 10965808,52 \\ 10965808,52 & 2,91 \times 10^9 \end{vmatrix}$$

$$det1 = 6,74741 \times 10^{19}$$

Matriks 2:

$$\begin{vmatrix} 146791,42 & 44830,17 & 10965808,52 \\ 35906322,81 & 10965808,52 & 2,91 \times 10^9 \\ 9533158224 & 2,91 \times 10^9 & 8,03 \times 10^{11} \end{vmatrix} \begin{vmatrix} 146791,42 & 44830,17 \\ 35906322,81 & 10965808,52 \\ 9533158224 & 2,91 \times 10^9 \end{vmatrix}$$

$$det2 = 0$$

Matriks 3:

$$\begin{vmatrix} 420 & 146791,42 & 10965808,52 \\ 44830,17 & 35906322,81 & 2,91 \times 10^9 \\ 10965808,52 & 9533158224 & 8,03 \times 10^{11} \end{vmatrix} \begin{vmatrix} 420 & 146791,42 \\ 44830,17 & 35906322,81 \\ 10965808,52 & 9533158224 \end{vmatrix}$$

$$det3 = 2,20936 \times 10^{20}$$

Matriks 4:

$$\begin{vmatrix} 420 & 44830,17 & 146791,42 \\ 44830,17 & 10965808,52 & 35906322,81 \\ 10965808,52 & 2,91 \times 10^9 & 9533158224 \end{vmatrix} \begin{vmatrix} 420 & 44830,17 \\ 44830,17 & 10965808,52 \\ 10965808,52 & 2,91 \times 10^9 \end{vmatrix}$$

$$det4 = 147456$$

Maka, akan diperoleh koefisien fungsi biaya sebagai berikut.

$$\begin{aligned}
 a &= \frac{0}{6,74741 \times 10^{19}} \\
 &= 0 \\
 b &= \frac{2,20936 \times 10^{20}}{6,74741 \times 10^{19}} \\
 &= 3,274389 \\
 c &= \frac{147456}{6,74741 \times 10^{19}} \\
 &= 2,18537 \times 10^{-15}
 \end{aligned}$$

Maka, akan didapatkan fungsi, $y' = 3,274389x + 2,18537 \times 10^{-15}x^2$

Dengan keterangan,

$y' = \text{biaya pembangkitan } \$/h$

$x = \text{daya aktif (MW)}$

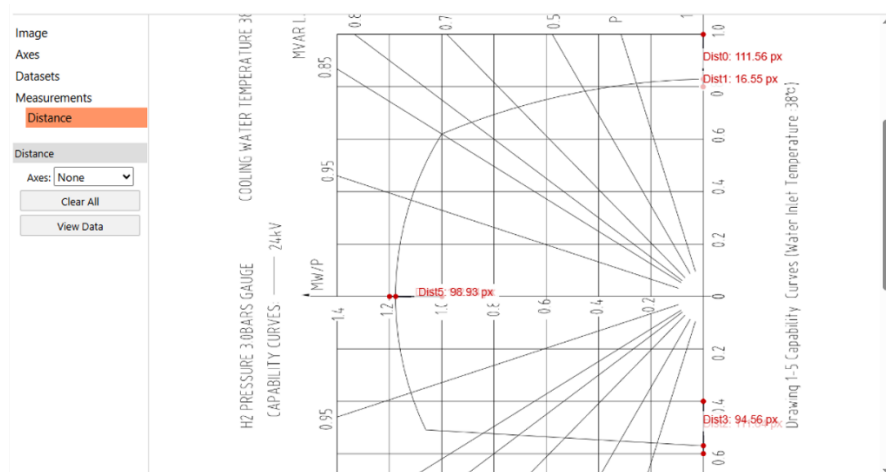
Satu catatan penting ialah fungsi yang telah dimodelkan tersebut hanya dapat digunakan pada simulasi dengan total permintaan beban pada rentang 0 MW – 399 MW dikarenakan variasi nilai sampel data historis yang menjadi bahan regresi sangat terbatas, lebih dari rentang tersebut, hasil simulasi menjadi tidak logis dan bahkan tidak akan konvergen. Hal tersebut diiringi dengan sensitivitas nilai konvergensi iterasi Newton-Raphson dengan nilai konvergensi yang semakin mengecil ketika total beban dinaikkan.

Untuk mengevaluasi fungsi yang telah diperoleh, perhitungan galat digunakan dengan galat akar rata-rata kuadrat dari model fungsi kuadrat adalah sebagai berikut.

$$\begin{aligned}
 RMSE &= \sqrt{\frac{\sum(y - y')^2}{n}} \\
 &= \sqrt{\frac{3,83435 \times 10^{-20}}{420}} \\
 &= 9,55479 \times 10^{-12}
 \end{aligned}$$

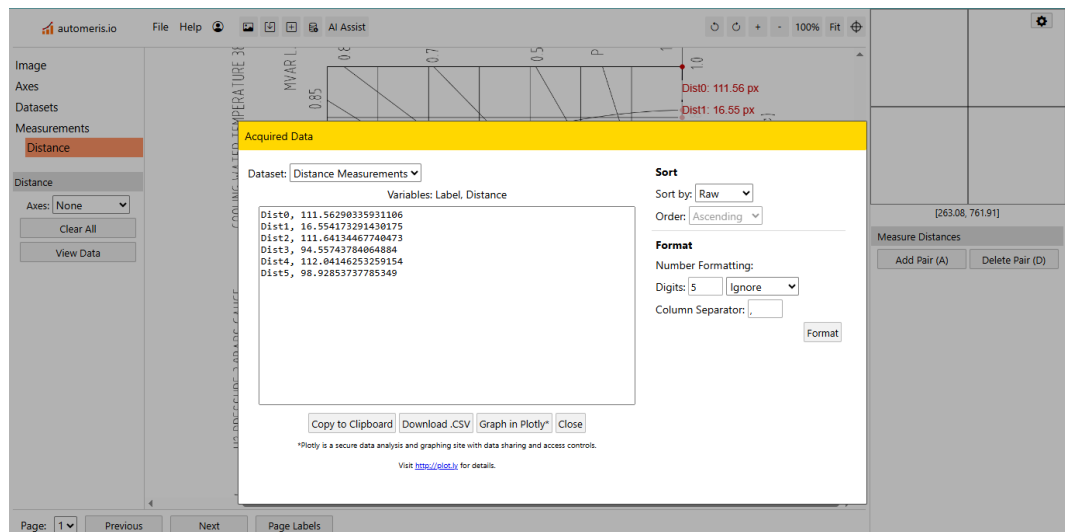
3.5.2 Penentuan Batas Operasi Daya pada Generator

Grafik kurva kapabilitas generator yang telah diperoleh akan diekstraksi untuk mengetahui titik-titik pada sumbu-x dan sumbu-y yang dipotong oleh kurva kapabilitas secara akurat. Ekstraksi kurva dilakukan dengan menggunakan bantuan web ekstraksi kurva yaitu *web plot digitizer*.



Gambar 3. 10 Hasil Ekstraksi pada Kurva Kapabilitas

Setelah hasil ekstraksi diperoleh, titik-titik akan dihitung nilai batas-batasnya berdasarkan keterangan berikut.



Gambar 3. 11 Nilai Jarak Setiap Titik

K_{et}:

- Dist0 Jarak titik 0,8 dengan 1,0 sumbu MVAR/P positif
- Dist1 Jarak titik potong kurva dengan titik 0,8 sumbu MVAR/P positif

- Dist2 Jarak titik 0,4 dengan 0,6 sumbu MVAR/P negatif
- Dist3 Jarak titik potong kurva dengan titik 0,4 sumbu MVAR/P negatif
- Dist4 Jarak titik 1,0 dengan 1,2 sumbu MW/P positif
- Dist5 Jarak titik potong kurva dengan titik 1,0 sumbu MW/P positif

Kemudian, jarak-jarak yang telah diperoleh akan dihitung berapa nilai MVAR/P dan MW/P aktual dengan metode perbandingan.

$$\frac{\text{titik potong MVAR/P sumbu positif}}{0,2} = \frac{16,55px}{111,56px}$$

$$\text{titik potong MVAR/P sumbu positif} = 0,029$$

$$Q_{max}/P = 0,029 + 0,8$$

$$Q_{max}/P = 0,829$$

$$\frac{\text{titik potong MVAR/P sumbu negatif}}{-0,2} = \frac{94,56px}{111,64px}$$

$$\text{titik potong MVAR/P sumbu negatif} = -0,16$$

$$Q_{min}/P = -0,16 - 0,4$$

$$Q_{min}/P = -0,56$$

$$\frac{\text{titik potong MW/P sumbu positif}}{0,2} = \frac{98,93px}{112,04px}$$

$$\text{titik potong MW/P sumbu positif} = 0,17$$

$$P_{max}/P = 0,17 + 1$$

$$P_{max}/P = 1,17$$

Sehingga akan diperoleh batas operasi generator sebagai berikut.

$$P_{max} = 1,17 \times 330$$

$$= 388,277 \text{ MW}$$

$$P_{min} = 0 \text{ MW}$$

$$Q_{max} = 0,83 \times 330$$

$$= 273,791 \text{ MVAR}$$

$$Q_{min} = -0,56 \times 330$$

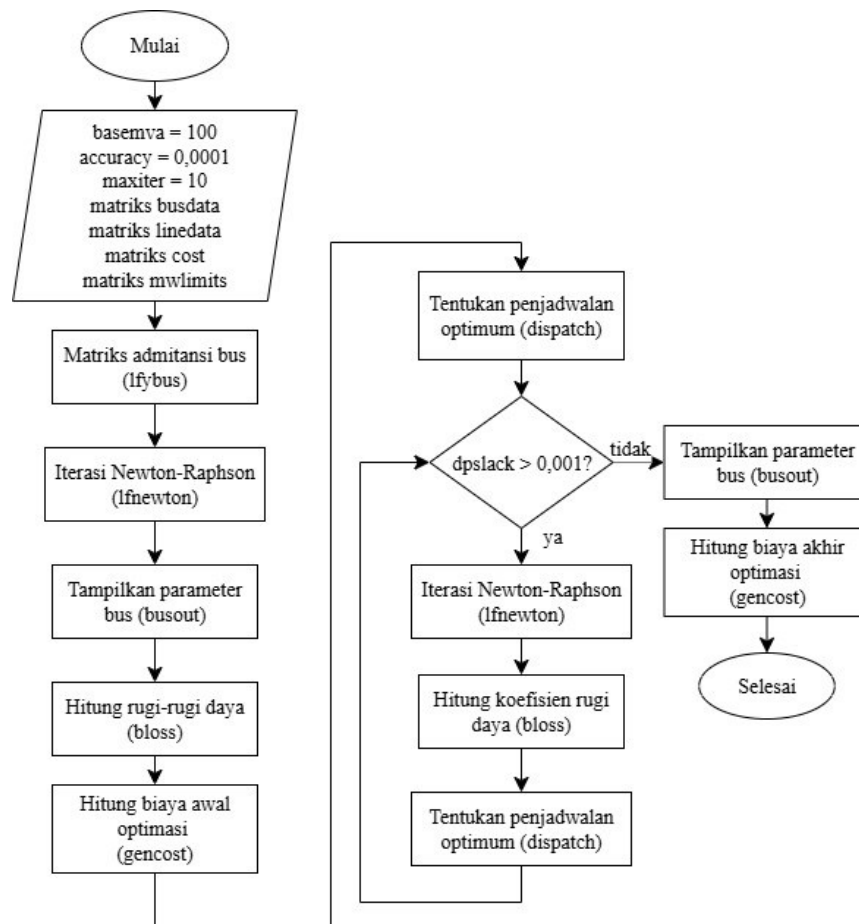
$$= -187,902 \text{ MVAR}$$

3.5.3 Kode Program Simulasi *Optimal Power Flow*

Kode program MATLAB yang digunakan pada dasarnya terdiri dari tujuh kode program yang dijalankan, yaitu program utama sebagai sistem inti dalam program optimasi OPF, program *dispatch*, program *bloss*, program *lfnewton*, program *gencost*, program *busout*, dan program *lfybus*. Berikut merupakan penjelasan setiap kode program berdasarkan diagram alir untuk mempermudah pemahaman.

1. Program Utama

Program utama merupakan program dimana semua data yang telah diolah akan di-*input* dan diprogram dengan secara langsung berhubungan dengan program-program lainnya.

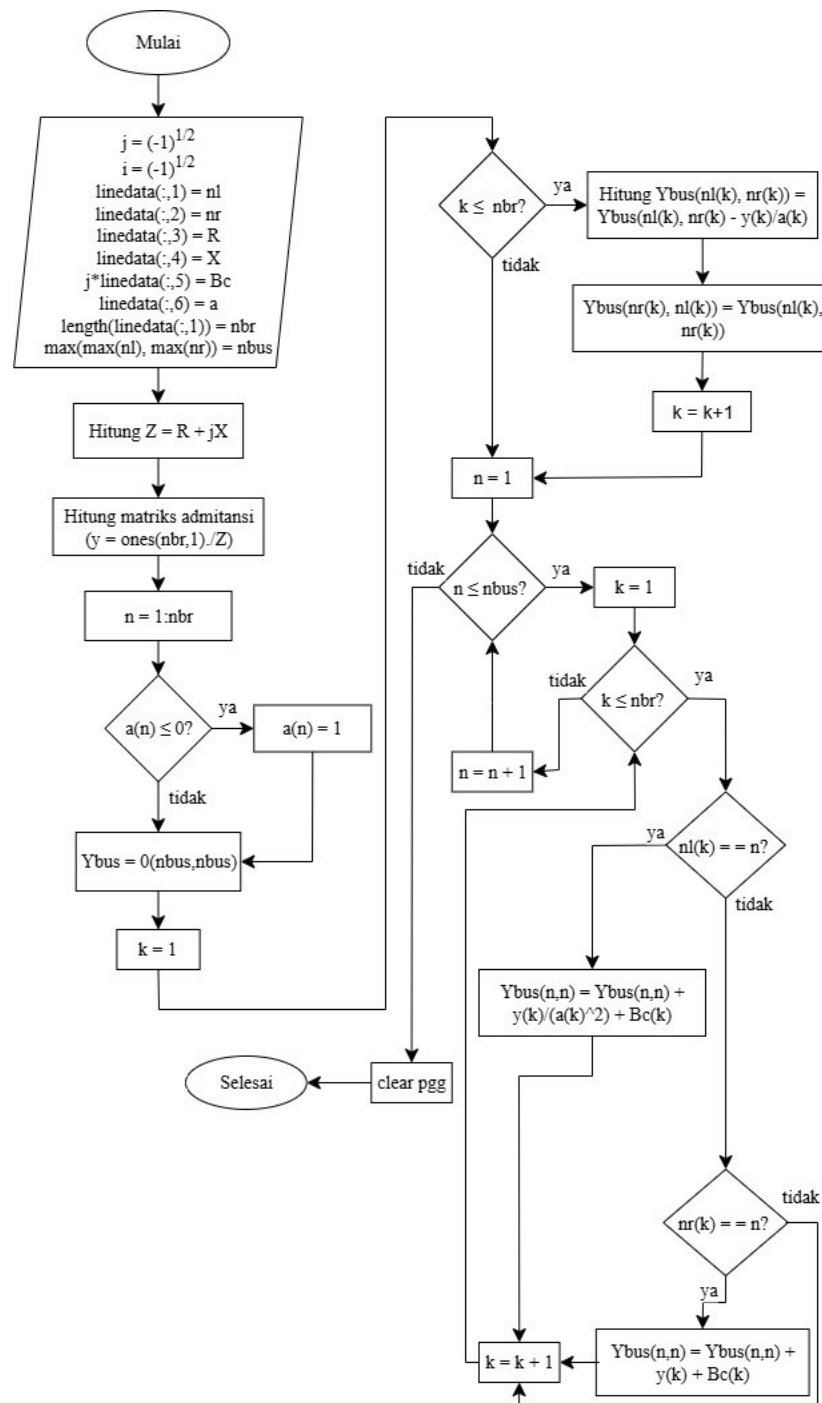


Gambar 3. 12 Flowchart Program Utama

- a. Tahap pertama yaitu inialisasi/input bahan-bahan OPF yang telah diperoleh secara manual dan definisi data seperti data *busdata*, *linedata*, *cost*, *basemva*, *maxiter*, *accuracy*, dan *mwlimits*.
- b. Tahap selanjutnya yaitu memulai perhitungan dengan serangkaian program lewat pemanggilan program-program tersebut yang diawali dengan menganalisis aliran daya sebelum optimasi. Proses awal yaitu pembentukan matriks admitansi bus pada *lfybus* seperti gambar 3.13.
- c. Lalu pemanggilan program *lfnewton* seperti gambar 3.14 yang berperan utama dalam iterasi Newton-Raphson.
- d. Kemudian pemanggilan program *busout* seperti pada gambar 3.15 untuk menampilkan parameter bus hasil analisis aliran daya.
- e. Kemudian pemanggilan program *bloss* seperti gambar 3.16 untuk menghitung rugi-rugi daya sistem.
- f. Terakhir dalam proses analisis aliran daya sebelum optimasi yaitu pemanggilan program *gencost* seperti gambar 3.17 untuk menghitung total pembangkitan generator.
- g. Proses selanjutnya yaitu mengawali optimasi dengan program *dispatch* seperti gambar 3.18 yang berfungsi mengatur penjadwalan atau penentuan daya aktif yang paling optimal pada tiap generator.
- h. Tahap selanjutnya yaitu penentuan *dpslack* atau yang disebut sebagai konvergensi *dpslack*. Nilai *dpslack* ialah nilai selisih antara daya aktif bus *slack* (MW) yang dihitung oleh fungsi aliran daya (*power flow*) lewat program *lfnewton* seperti pada gambar 3.14 dengan daya aktif bus *slack* (MW) hasil penjadwalan pembangkitan lewat program *dispatch* seperti pada gambar 3.18. Di mana ketika nilai *dpslack* lebih dari batas toleransi, akan dilakukan iterasi Newton-Raphson lagi seperti pada gambar 3.14 pada hasil optimasi, lalu dioptimasi lagi dengan program *dispatch* seperti gambar 3.18.
- i. Tahap akhir yaitu ketika nilai $dpslack \leq 0,001$ program akan mencetak solusi akhir yang sudah dioptimasi termasuk parameter bus dan total biaya pembangkitan akhir.

2. Program *lfybus*

Berikut merupakan *flowchart* program *lfybus* yang memiliki fungsi utama dalam program yaitu membentuk matriks admitansi bus.

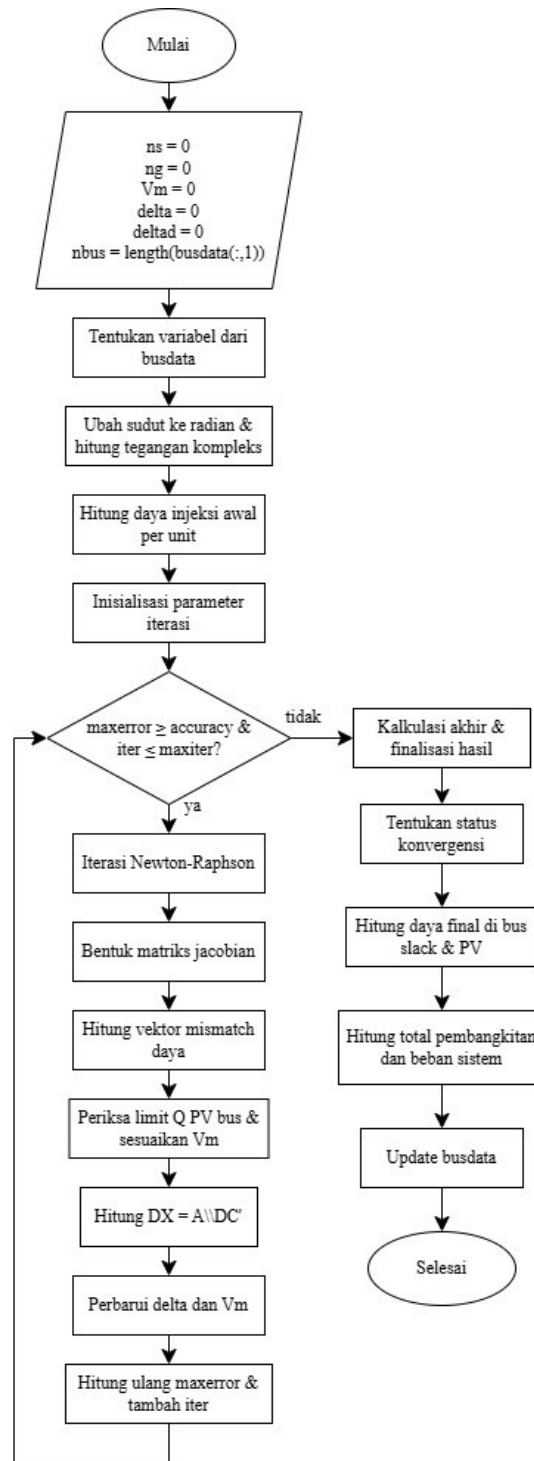


Gambar 3. 13 *Flowchart* Program *lfybus*

- a. Tahap pertama yaitu inisialisasi dan persiapan data dari matriks *linedata* seperti inisialisasi bilangan kompleks, resistansi saluran, reaktansi saluran, suseptansi saluran, bus asal, bus tujuan, rasio tap *changing*, jumlah saluran, dan jumlah bus dari data saluran.
- b. Tahap selanjutnya yaitu menghitung impedansi saluran dengan bilangan kompleks berdasarkan data yang di-*input*.
- c. Selanjutnya yaitu menghitung admitansi saluran dari hasil proses sebelumnya.
- d. Tahap selanjutnya yaitu pengecekan data untuk memastikan tidak adanya nilai *tap* rasio transformator yang tidak valid, jika ada yang tidak valid/memenuhi kondisi $a(n) \leq 0$ akan di-*set default* menjadi 1. Jika nilai valid, nilai dibiarkan sesuai data.
- e. Tahap selanjutnya yaitu membuat matriks $n \times n$ dengan semua nilai elemennya nol untuk diisi dengan nilai admitansi.
- f. Tahap selanjutnya yaitu proses *loop* untuk setiap saluran di mana memastikan kondisi $k \leq nbr$. Jika “ya”/kondisi terpenuhi, program memulai perhitungan pada proses setelahnya hingga proses $k = k + 1$.
- g. Selanjutnya yaitu proses *loop* $n \leq nbus$?. Jika kondisi “ya”, proses dilanjutkan ke *loop* $k = 1$ ke nbr . Jika kondisi memenuhi “tidak”, proses diagonal *Ybus* selesai dan program *lfybus* selesai.
- h. Tahap selanjutnya yaitu ketika kondisi kondisi “ya” terpenuhi pada proses sebelumnya, program masuk ke *loop* $k \leq nbr$?. Jika kondisi “ya” terpenuhi, akan dilanjutkan ke proses $nl(k) == n$?. Jika memenuhi kondisi “ya”, perhitungan penambahan pada *Ybus* ($n \times n$) dilakukan hingga iterasi $k = k + 1$ dan kembali ke proses poin “f” untuk memastikan $k \leq nbr$?. Jika kondisi “tidak”, lanjut ke proses $nr(k) == n$?. Jika “ya”, perhitungan penambahan pada *Ybus* ($n \times n$) dilakukan hingga iterasi $k = k + 1$ dan kembali ke proses poin “f” untuk memastikan $k \leq nbr$?. Jika kondisi “tidak”, proses iterasi $k = k + 1$ lalu kembali ke proses “f”.

3. Program *lfnewton*

Berikut merupakan *flowchart* dari program *lfnewton* yang berfungsi sebagai komputasi iterasi Newton-Raphson.

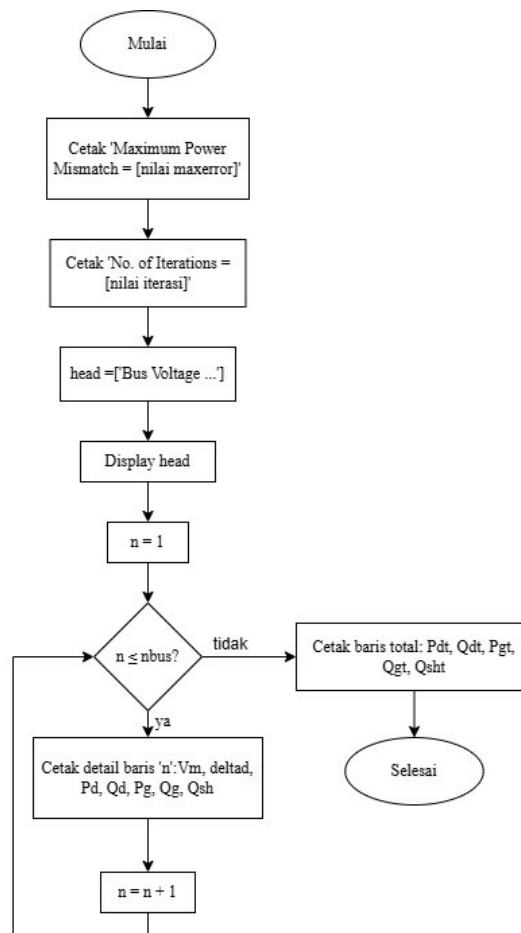


Gambar 3. 14 *Flowchart* Program *lfnewton*

- a. Tahap pertama yaitu inisialisasi dan persiapan data mentah dari matriks *busdata*. Data tersebut kemudian diolah menjadi format yang siap digunakan untuk perhitungan.
- b. Tahap selanjutnya yaitu mengekstrak data-data dari matriks *busdata* dan disimpan ke dalam vektor terpisah seperti V_m , delta, dll.
- c. Selanjutnya yaitu mengubah satuan derajat pada perhitungan *sinus* dan *cosinus* menjadi sudut radian lalu menggabungkan V_m dengan delta sehingga membentuk tegangan dalam bentuk bilangan kompleks.
- d. Tahap selanjutnya yaitu menghitung daya aktif dan daya reaktif di setiap bus dengan mengurangi beban dari pembangkitan dan mengubah satuannya menjadi satuan per unit.
- e. Tahap selanjutnya yaitu blok *decision loop* iterasi utama Newton-Raphson. Program akan terus melakukan iterasi jika $maxerror \geq accuracy$ atau jumlah iterasi (*iter*) \leq batas iterasi. Jika salah satu kondisi tidak terpenuhi, *loop* berhenti dan lanjut ke finalisasi akhir.
- f. Tahap selanjutnya pada kondisi “ya” setelah iterasi Newton-Raphson dilakukan, akan membentuk matriks *jacobian* dan menghitungnya.
- g. Tahap selanjutnya yaitu menghitung *mismatch* antara daya yang dijadwalkan dengan daya yang dihitung berdasarkan kondisi tegangan saat ini kemudian nilainya disimpan dalam vektor DC.
- h. Tahap selanjutnya yaitu pemeriksaan daya reaktif pada bus generator apakah nilainya dalam batas minimum dan maksimumnya, jika di luar rentang, program akan menyesuaikan tegangan *magnitude* untuk mencoba mengembalikannya ke dalam batas.
- i. Tahap selanjutnya yaitu program menyelesaikan persamaan $A \times DX = DC$ untuk menemukan vektor DX yang berisi nilai-nilai koreksi yang dibutuhkan untuk sudut dan tegangan.
- j. Selanjutnya yaitu nilai koreksi dari DX ditambahkan ke nilai sudut delta dan V_m .

- k. Selanjutnya yaitu perhitungan ulang *maxerror* berdasarkan nilai DC yang baru dan penambahan iterasi dan kemudian laur kembali ke blok *decision* awal.
 - l. Tahap terakhir yaitu kalkulasi dan finalisasi akhir ketika blok *decision* dalam kondisi “tidak”.
 - m. Selanjutnya yaitu menentukan status konvergensi iterasi, jika gagal akan ditampilkan teks kegagalan konvergensi.
 - n. Selanjutnya yaitu perhitungan daya aktif dan reaktif di *slack* bus dan PV bus lalu perhitungan total pembangkitan dan beban sistem.
 - o. Selanjutnya yaitu *update* matriks *busdata* dengan *V_m* dan delta akhir.
4. Program *busout*

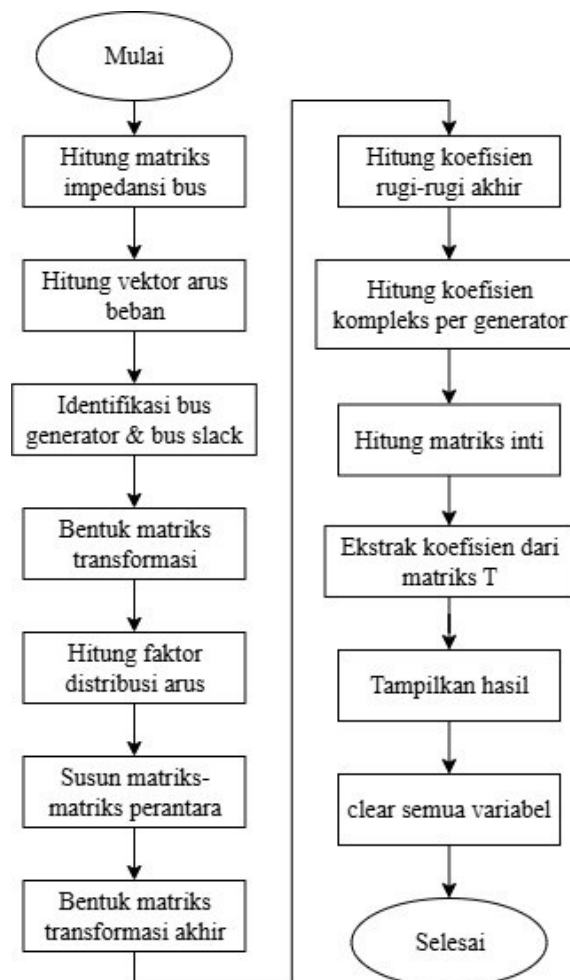
Berikut merupakan *flowchart* program *busout* yang berfungsi untuk melaporkan hasil OPF dan aliran daya per bus.



Gambar 3. 15 Flowchart Program *busout*

- a. Tahap pertama yaitu menampilkan judul yang disimpan di variabel *head*.
 - b. Tahap kedua yaitu blok perulangan “for” pada $n = 1:nbus$, jika kondisi $n \leq nbus$, satu baris lengkap yang berisi data tegangan, sudut, dll dicetak untuk bus ke- n , lalu nilai n ditambah 1 lalu kembali ke pengecekan iterasi berikutnya.
 - c. Jika kondisi $n > nbus$, program akan keluar dari *loop* dan mengeksekusi perintah mencetak baris total beserta nilai-nilainya.
 - d. Selesai.
5. Program *bloss*

Berikut merupakan *flowchart* program *bloss* yang berfungsi untuk menghitung rugi-rugi daya sistem.



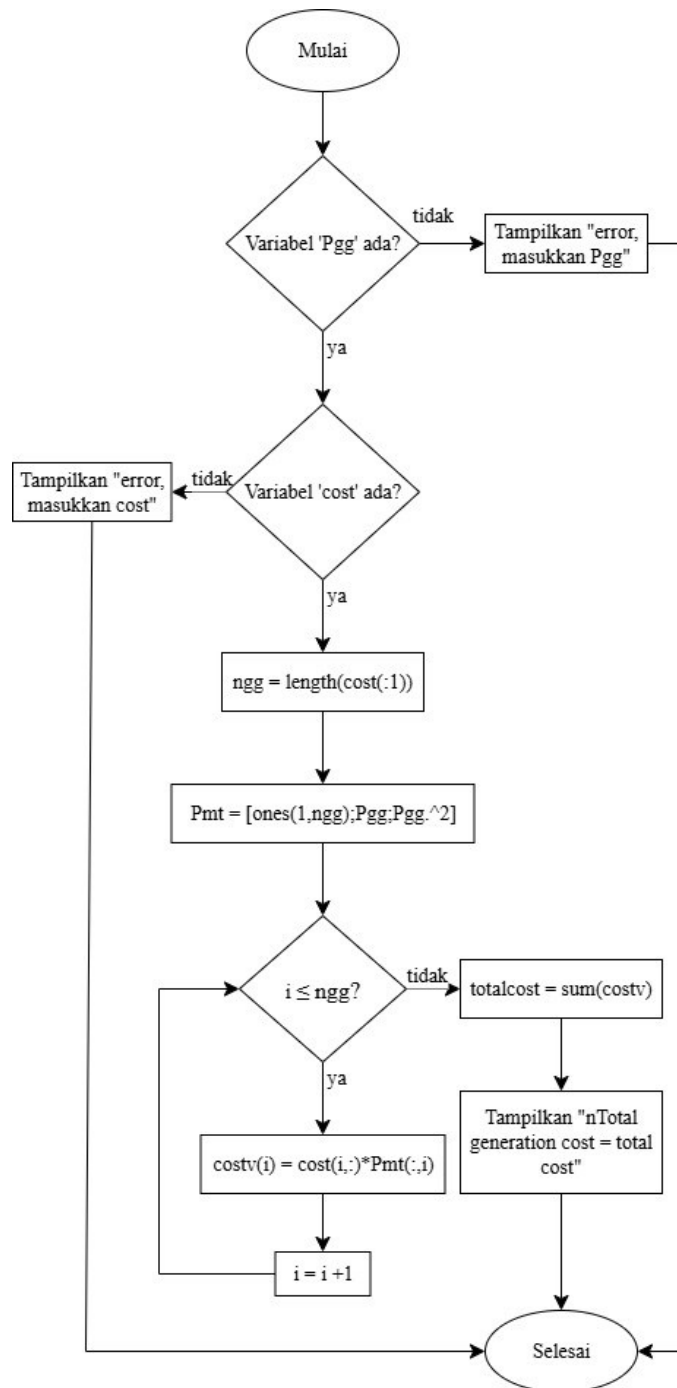
Gambar 3. 16 *Flowchart* Program *bloss*

- a. Tahap pertama yaitu menghitung Z_{bus} dari Y_{bus} yang menggambarkan hubungan tegangan dan arus di seluruh sistem.
- b. Blok selanjutnya yaitu menghitung arus yang ditarik oleh tiap beban berdasarkan data daya dan tegangan.
- c. Tahap selanjutnya yaitu mengidentifikasi tipe-tipe bus dengan menandainya dengan nomor.
- d. Tahap selanjutnya yaitu pembentukan matriks transformasi yang menghubungkan arus dari masing-masing generator dengan total arus beban sistem.
- e. Tahap selanjutnya yaitu perhitungan vektor d dan t_1 yang menggambarkan bagaimana arus beban dan arus dari bus *slack* terdistribusi dalam sistem.
- f. Tahap selanjutnya ialah pembentukan matriks-matriks $C1g$, $C1gg$, $C2$, dan lainnya.
- g. Selanjutnya yaitu pembentukan matriks transformasi dari parameter-parameter utama yaitu karakteristik fisik jaringan, kondisi operasi beban, dan konfigurasi sistem.
- h. Tahap selanjutnya yaitu menghitung *B-Coefficients* dengan menggunakan Z_{bus} dan matriks transformasi.
- i. Tahap selanjutnya yaitu perhitungan koefisien kompleks per generator yang merepresentasikan karakteristik operasi setiap generator.
- j. Tahap selanjutnya yaitu perhitungan matriks inti yaitu matriks T yang mengandung informasi yang dibutuhkan untuk menurunkan formula rugi-rugi daya.
- k. Tahap selanjutnya yaitu mengekstrak koefisien dari matriks T dengan mengambil bagian BB kemudian mengambil elemen-elemen matriksnya untuk mendapatkan matriks B , vektor B_0 , dan scalar B_{00} .
- l. Tahap selanjutnya yaitu setelah *B-Coefficients* didapatkan, program menampilkannya dan menggunakannya untuk menghitung total rugi-rugi daya.

m. Tahap terakhir yaitu program akan membersihkan semua variabel perantara yang sudah tidak diperlukan lagi.

6. Program *gencost*

Berikut merupakan *flowchart* program *gencost* untuk menghitung biaya pembangkitan generator.

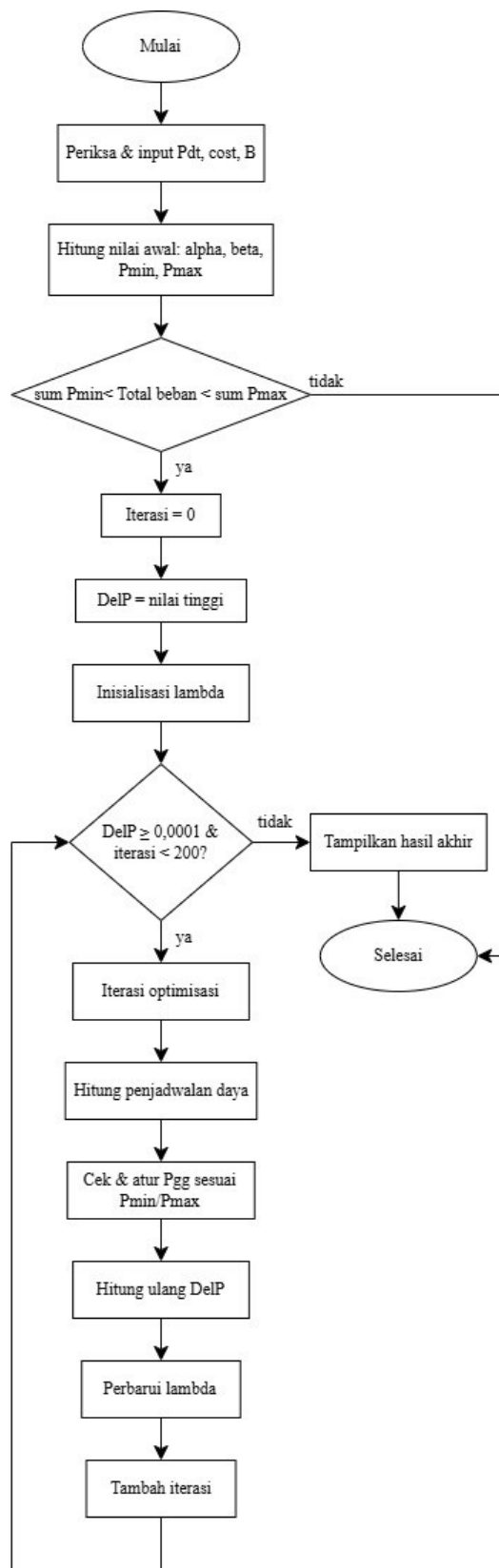


Gambar 3. 17 *Flowchart* Program *gencost*

- a. Tahap pertama yaitu pemeriksaan variabel *Pgg*, jika tidak ada di *workspace*, akan tampil di layar *command window* permintaan *input* data dan program keseluruhan akan berakhir.
- b. Jika terdapat data *Pgg*, tahap selanjutnya yaitu pemeriksaan variabel *cost*, jika tidak ada di *workspace*, akan tampil di layar *command window* permintaan *input* data lalu program keseluruhan akan berakhir.
- c. Jika terdapat data *cost*, tahap selanjutnya yaitu menghitung jumlah generator dengan mengambil jumlah baris dari matriks *cost* di program utama.
- d. Selanjutnya yaitu membuat matriks *Pmt* dari variabel *Pgg* untuk menyederhanakan perhitungan biaya polinomial.
- e. Tahap selanjutnya yaitu perulangan/*loop* sebanyak jumlah generator untuk menghitung biaya per generator di mana inisialisasi $i = 1$.
- f. Tahap selanjutnya yaitu jika $i \leq \text{ngg}$, perkalian matriks antara $\text{cost}(i, :)$ yang komponen matriksnya berisi data koefisien biaya pembangkitan dengan matriks $\text{pmt}(i, :)$ yang komponen matriksnya berisi daya aktif kuadrat generator, daya aktif generator, dan angka 1 dilakukan untuk membentuk $\text{costv}(i)$.
- g. Tahap selanjutnya yaitu iterasi ditambahkan untuk menghitung biaya generator selanjutnya dan kemudian alur kembali ke proses “e”.
- h. Tahap selanjutnya pada kondisi “tidak” pada blok *decision* $i \leq \text{ngg}$? pada proses “e”, dilakukan proses penjumlahan semua elemen costv yang mana elemen ini merepresentasikan semua biaya pembangkitan tiap generator.
- i. Tahap terakhir yaitu program menampilkan *totalcost* ke layar.

7. Program *dispatch*

Program *dispatch* merupakan program penjadwalan pembangkitan daya aktif generator untuk mencapai biaya pembangkitan paling optimal. Berikut merupakan *flowchart* program *dispatch* yang berfungsi sebagai inti proses OPF terjadi.



Gambar 3. 18 Flowchart Program *dispatch*

- a. Tahap pertama yaitu pengecekan variabel P_{dt} , $cost$, dan B . Jika belum ada, akan permintaan *input* data.
- b. Tahap selanjutnya yaitu menghitung α dan β dari matriks $cost$ dan batas pembangkitan generator diambil dari matriks $mwlimits$.
- c. Tahap selanjutnya yaitu program melakukan pengecekan sederhana seperti apakah total beban yang diminta dalam rentang batas pembangkitan. Jika beban di luar rentang, program akan menampilkan pesan *error* dan berhenti.
- d. Tahap selanjutnya yaitu inisialisasi penghitung iterasi ke-nol
- e. Selanjutnya yaitu penentuan $DelP$ yang mengukur *error* atau selisih antara daya yang dibangkitkan dengan daya yang dibutuhkan.
- f. Tahap selanjutnya yaitu inisialisasi λ yang merupakan biaya marginal sistem sebagai biaya tambahan untuk menghasilkan satu MWh daya aktif tambahan.
- g. Tahap selanjutnya yaitu *loop* utama. Jika kedua kondisi $DelP \geq 0,0001$ dan iterasi < 200 , proses akan berlanjut.
- h. Tahap selanjutnya yaitu iterasi optimisasi untuk menghasilkan daya pembangkitan sementara berdasarkan biaya marginal.
- i. Tahap selanjutnya yaitu menghitung penjadwalan daya setiap generator sebagai tebakan pertama dalam iterasi.
- j. Tahap selanjutnya yaitu pemeriksaan P_{gg} dimana jika hasil perhitungan P_{gg} keluar dari rentang batasnya, nilainya akan dipaksa untuk sama dengan batas tersebut. Generator yang sudah mencapai batasnya tidak akan diikutsertakan dalam pembaruan λ .
- k. Tahap selanjutnya yaitu perhitungan ulan $DelP$.
- l. Tahap terakhir dari *loop* ialah memperbarui λ dan menambah iterasi.
- m. Tahap terakhir dari program yaitu program akan menampilkan hasil nilai λ .