LAMPIRAN

- **Program Metode Micro Genetic Algorithm**

```
closeall
clearall
clc;
clear;
basemva = 1000;  accuracy = 0.0001;  maxiter = 10;

%Parameter GA
    Nvar=6;   % 6 variabel 6 pembangkit
    Nbit=10;    %tiap variabel dikodekan dalam 10 bit
    JumGen=Nbit*Nvar;  % tiap kromosom ada 8*10 bit
    UkPop=30;     % Ada 30 kromosong tiap generasi
    Psilang=0.5;     % Probabilitas Pindah Silang
    Pmutasi=0,1;  % Probabilitas Mutasi
    MaxG=20;       % diiterasi sebanyak 30 kali
    Uktur=40;
    Prob=0.75;

    BilKecil =10^-1;
    Fthreshold=1/BilKecil;  % kondisi berhenti selain MAXG, kalau
nilai fitness = Fthreshold akan berhenti
    Bgraf=18.5;
%==================================================================
========
%tampilan Grafik
hfig=figure;
holdon;
title('Aplikasi Micro Genetic Algorithm pada Economic Dispatch
Hadi Saadat 26 Bus');
set(hfig,'position',[50,50,600,400]);
set(hfig,'DoubleBuffer','on');
hbestplot=plot(1:MaxG,zeros(1,MaxG));
xlabel('Generasi');
ylabel('Total biaya pembangkitan dalam Rupiah (Rp)');
holdoff;
drawnow;
%==================================================================
=======
saadat_26_bus;
loadflow;
gencost;
%
%==================================================================
=======
It=1;
while It<= 1 %maxiter
%while sum(costv)>=6900000000
tic;
%inisilisasi
Populasi=InisialisasiPopulasi(UkPop,JumGen);
Ra=mwlimits(:,1);
```

```
Rb=mwlimits(:,2);
MaxF=0;
MinF=1;
%=================================================================
=======
%loop evolusi
for generasi=1:MaxG,
%dekodekan Kromosom
for ii=1:UkPop,
   Kromosom = Populasi(ii,:);
   x= DekodekanKromosom(Kromosom,Nvar,Nbit,Ra,Rb);
busdata(1,7)  = x(1); % x1 terletak pada kolom 1 sebanyak jumlah
particle dalam kolom (Matriks 1x50)
busdata(2,7)  = x(2); % x1 terletak pada kolom 2 sebanyak jumlah
particle dalam kolom (Matriks 1x50)
busdata(3,7)  = x(3);
busdata(4,7)  = x(4);
busdata(5,7)  = x(5);
busdata(26,7)  = x(6);
x(1)=Pgg(1);
if x(1)>mwlimits(1,2)
x(1)=mwlimits(1,2);
end
loadflow;
gencost;
% Evaluasi fitness populasi ke 2 sampai ke UkPop%
Fitness(ii)=1/(totalcost+BilKecil);
%-----------------------------------------------------------------
---------
%cek konvergensi nilai fitness
if (Fitness(ii) > MaxF),
       MaxF = Fitness(ii);
       harga1=totalcost
       IndeksIndividuTerbaik = ii;
       BestX = x;
end
if (Fitness(ii) < MinF),
        MinF = Fitness(ii);
end
end
%-----------------------------------------------------------------
---------


%Grafis 2D
plotvector=get(hbestplot,'YData');
plotvector(generasi)=harga1;
set(hbestplot,'YData',plotvector)
drawnow;
%-----------------------------------------------------------------
---------
      LinearFitness=
LinearFitnessRanking(UkPop,Fitness,MaxF,MinF);
     LFR=LinearFitness
%-----------------------------------------------------------------
---------
%Tournament selection
```

```
fitsemen=LFR;
indx=find(fitsemen>=mean(fitsemen));
keep=length(indx);
if keep==0
break
end
fitsemen=fitsemen(indx);
Ps=Populasi(indx,:);
M=UkPop-keep;
% Create mating pool using tournament selection
Ntourn=2;
for ic=1:M;
rc=ceil(keep*rand(1,Ntourn));
[c,ci]=max(fitsemen(rc));% indicies of mother
ma=rc(ci);
rc=ceil(keep*rand(1,Ntourn));
[c,ci]=max(fitsemen(rc)); % indicies of father
pa=rc(ci);
%-------------------------------------------------------------------
---------
% crossover
% generate mask
mask=round(rand(1,JumGen));
Ps(keep+ic,:)=mask.*Populasi(ma,:)+not(mask).*Populasi(pa,:);
end
%elitismmmm
% Elitisme:
% - Buat satu kopi kromosom terbaik jika ukuran populasi ganjil
% - Buat dua kopi kromosom terbaik jika ukuran populasi genap
% ukuran populasi ganjil
Ps(1,:) = Populasi(IndeksIndividuTerbaik,:);
nilaiterbaik = Populasi(IndeksIndividuTerbaik,:);

if MaxF>=Fthreshold,
break;
end
%--------------------------------------------------
   Populasi=Ps;
end

x=BestX
busdata(1,7)  = x(1); % x1 terletak pada kolom 1 sebanyak jumlah
particle dalam kolom (Matriks 1x50)
busdata(2,7)  = x(2); % x1 terletak pada kolom 2 sebanyak jumlah
particle dalam kolom (Matriks 1x50)
busdata(3,7)  = x(3);
busdata(4,7)  = x(4);
busdata(5,7)  = x(5);
busdata(26,7) = x(6);

x(1)=Pgg(1);
if x(1)<mwlimits(1,1)
Pgg(1)=mwlimits(1,1);
end
if x(1)>mwlimits(1,2)
Pgg(1)=mwlimits(1,2);
```

```
end
loadflow;
gencost
fprintf('\n')

for ii = 1:Nvar
fprintf('Pembangkit Unit %1.0f',ii)
fprintf(' = %10.2f',Pgg(ii))
fprintf(' MW')
fprintf('     Biaya Pembangkitan Unit %1.0f',ii)
fprintf(' =  %10.2f',costv(ii))
fprintf(' Rp/jam\n')
end
fprintf('\n')
fprintf('Total Pembangkitan = %10.2f',sum(Pgg))
fprintf(' MW')
fprintf('\n')
fprintf('Total Biaya Pembangkitan1 = %10.2f',sum(costv))
fprintf(' Rp/jam\n')
fprintf('Total Losses System = %10.2f',sum(Pgg)-Pdt)
fprintf(' MW\n')
fprintf('\n')
toc
Pgg(1)
%+================================

%                ===================================
It=It+1;

end
```

- **Program Metode Genetic Algorithm**

```
clc% Me-refresh command window
clearall% Menghapus semua semua variabel yang sedang aktif
closeall

Nvar         = 6;              % Jumlah variabel pada fungsi yang
dioptimasi
Nbit         = 10;             % Jumlah bit yang mengkodekan satu
variabel
JumGen       = Nbit*Nvar;      % Jumlah gen dalam kromosom
% Rb          = [-5.12 -5.12];         % Batas bawah interval
% Ra          = [ 5.12  5.12];          % Batas atas interval

UkPop        = 30;            % Jumlah kromosom dalam populasi
Psilang      = 0.5;            % Probabilitas pindah silang
Pmutasi      = 0.1;           % Probabilitas mutasi
MaxG         = 20;           % Jumlah generasi

BilKecil     = 10^-1;          % Digunakan untuk menghindari
pembagian dengan 0
Fthreshold   = 1/BilKecil;     % Threshold untuk nilai Fitness
% Bgraf       = Fthreshold;     % Untuk menangani tampilan grafis
Bgraf=18.5;

%=================================================================
========
%tampilan Grafik
hfig=figure;
holdon;
title('Aplikasi Micro Genetic Algorithm pada Economic Dispatch
Hadi Saadat 26 Bus');
set(hfig,'position',[50,50,600,400]);
set(hfig,'DoubleBuffer','on');
hbestplot=plot(1:MaxG,zeros(1,MaxG));
xlabel('Generasi');
ylabel('Total biaya pembangkitan dalam Rupiah (Rp)');
holdoff;
drawnow;
%=================================================================
=======
saadat_26_bus;
loadflow;
gencost;
%
%=================================================================
=======

% Inisialisasi populasi
% Membangkitkan sejumlah UkPop kromosom, masing-masing kromosom
berisi bilangan
% biner (0 dan 1) sejumlah JumGen
Populasi = fix(2*rand(UkPop,JumGen));
```

```
% Loop evolusi

tic;
%inisilisasi
%Populasi=InisialisasiPopulasi(UkPop,JumGen);
Ra=mwlimits(:,1);
Rb=mwlimits(:,2);
MaxF=0;
MinF=1;

for generasi=1:MaxG,

% Mendekodekan kromosom yang berisi bilangan biner menjadi
individu x yang
% bernilai real dalam interval yang ditentukan [Ra,Rb]
for ii=2:UkPop,
   Kromosom=Populasi(1,:);
for n=1:Nvar,
x(n) = 0;
for b=1:Nbit,
x(n) = x(n) + Kromosom((n-1)*Nbit+b)*2^(-b);
end
x(n) = Rb(n) + (Ra(n)-Rb(n))*x(n);
end

% %  Mengevaluasi nilai fitness populasi ke-1
%    Fitness(1) = 1 / ((1000*(x(1) - 2*x(2))^2 + (1-x(1))^2) +
BilKecil);
%    MaxF = Fitness(1);
%    MinF = Fitness(1);
%    IndeksIndividuTerbaik = 1;

%    busdata(1,7)   = x(1); % x1 terletak pada kolom 1 sebanyak
jumlah particle dalam kolom (Matriks 1x50)
busdata(2,7)   = x(2); % x1 terletak pada kolom 2 sebanyak jumlah
particle dalam kolom (Matriks 1x50)
busdata(3,7)   = x(3);
busdata(4,7)   = x(4);
busdata(5,7)   = x(5);
busdata(26,7)   = x(6);
x(1)=Pgg(1);
if x(1)>mwlimits(1,2)
x(1)=mwlimits(1,2);
end
loadflow;
gencost;
% Evaluasi fitness populasi ke 2 sampai ke UkPop%
Fitness(ii)=1/(totalcost+BilKecil);
%-----------------------------------------------------------------
---------
%cek konvergensi nilai fitness
if (Fitness(ii) > MaxF),
       MaxF = Fitness(ii);
       harga1=totalcost
```

```
 IndeksIndividuTerbaik = ii;
       BestX = x;
end
if (Fitness(ii) < MinF),
        MinF = Fitness(ii);
end
end
%-----------------------------------------------------------------
----------

%Grafis 2D
plotvector=get(hbestplot,'YData');
plotvector(generasi)=harga1;
set(hbestplot,'YData',plotvector)
drawnow;
%-----------------------------------------------------------------
----------
%       LinearFitness=
LinearFitnessRanking(UkPop,Fitness,MaxF,MinF);
%      LFR=LinearFitness
%-----------------------------------------------------------------
----------


%    for ii=2:UkPop,
% % Dekodekan Kromosom populasi ke 2 sampai ke UkPop
%     Kromosom = Populasi(ii,:);
%         for n=1:Nvar,
%             x(n) = 0;
%             for b=1:Nbit,
%             x(n) = x(n) + Kromosom((n-1)*Nbit+b)*2^(-b);
%             end
%             x(n) = Rb(n) + (Ra(n)-Rb(n))*x(n);
%         end
%
% % Evaluasi fitness populasi ke 2 sampai ke UkPop
% %          Fitness(i) = 1 / ((1000*(x(1) - 2*x(2))^2 + (1-x(1))^2)
+ BilKecil);
%
% %    busdata(1,7)  = x(1); % x1 terletak pada kolom 1 sebanyak
jumlah particle dalam kolom (Matriks 1x50)
%    busdata(2,7)  = x(2); % x1 terletak pada kolom 2 sebanyak
jumlah particle dalam kolom (Matriks 1x50)
%    busdata(3,7)  = x(3);
%    busdata(4,7)  = x(4);
%    busdata(5,7)  = x(5);
%    busdata(26,7)  = x(6);
%    x(1)=Pgg(1);
%    if x(1)<mwlimits(1,1)
%       Pgg(1)=mwlimits(1,1);
%    end
%    if x(1)>mwlimits(1,2)
%       Pgg(1)=mwlimits(1,2);
%    end
%     loadflow;
%    gencost;
% % Evaluasi fitness populasi ke 2 sampai ke UkPop%
```

```
%     Fitness(ii)=1/(totalcost+BilKecil);
% %-----------------------------------------------------------------
-----------
% %cek konvergensi nilai fitness
%      if (Fitness(ii) > MaxF),
%         MaxF = Fitness(ii);
%         IndeksIndividuTerbaik = ii;
%         BestX = x
%      end
%
%      if (Fitness(ii) < MinF),
%         MinF = Fitness(ii);
%      end
%    end


if MaxF >= Fthreshold,
break;
end

   TempPopulasi = Populasi;

% Elitisme:
% - Buat satu kopi kromosom terbaik jika ukuran populasi ganjil
% - Buat dua kopi kromosom terbaik jika ukuran populasi genap
if mod(UkPop,2)==0,             % ukuran populasi genap
      IterasiMulai = 3;
TempPopulasi(1,:) = Populasi(IndeksIndividuTerbaik,:);
TempPopulasi(2,:) = Populasi(IndeksIndividuTerbaik,:);
else% ukuran populasi ganjil
      IterasiMulai = 2;
TempPopulasi(1,:) = Populasi(IndeksIndividuTerbaik,:);
end

% LinearFitness = LinearFitnessRanking(UkPop,Fitness,MaxF,MinF);
% SF berisi nilai fitness yang terurut dari kecil ke besar
(ascending)
% IndF berisi index dari nilai fitness yang menyatakan nomor urut
kromosom
  [SF,IndF] = sort(Fitness);
% LinearFitness = nilai fitness baru hasil pen-skala-an
for rr=1:UkPop,
LFR(IndF(UkPop-rr+1)) = MaxF-(MaxF-MinF)*((rr-1)/(UkPop-1));
end
    LinearFitness = LFR;

% Roulette-wheel selection dan pindah silang
% Memilih orang tua menggunakan LinearFitness, yaitu nilai fitness
hasil
% pen-skala-an. Pemilihan dilakukan secara proporsional sesuai
dengan
% nilai fitness-nya.
for jj=IterasiMulai:2:UkPop,
       JumFitness = sum(LinearFitness);
```

```
KumulatifFitness  = 0;
        RN = rand;
ii = 1;
while ii <= UkPop,
          KumulatifFitness = KumulatifFitness + LinearFitness(ii);
if (KumulatifFitness/JumFitness) > RN,
            Pindex = ii;
break;
end
ii = ii + 1;
end
       IP1 = ii;
       IP2 = ii;

% Memindah-silangkan bagian kromosom Bapak dan Ibu yang dipotong
% secara random, sehingga dihasilkan dua buah kromosom Anak
if (rand < Psilang),
          Bapak=Populasi(IP1,:);
          Ibu=Populasi(IP2,:);
% Membangkitkan satu titik potong (TP bernilai antara 1 sampai
JumGen-1)
            TP = 1 + fix(rand*(JumGen-1));
% Anak 1 berisi bagian depan Bapak dan bagian belakang Ibu
Anak(1,:) = [Bapak(1:TP) Ibu(TP+1:JumGen)];
Anak(2,:) = [Ibu(1:TP) Bapak(TP+1:JumGen)];
TempPopulasi(jj,:) = Anak(1,:);
TempPopulasi(jj+1,:) = Anak(2,:);
else
TempPopulasi(jj,:) = Populasi(IP1,:);
TempPopulasi(jj+1,:) = Populasi(IP2,:);
end
end


% Mutasi dilakukan pada semua kromosom
% Mutasi gen dengan probabilitas sebesar Pmutasi
% Gen-gen yang terpilih diubah nilainya: 0 menjadi 1, dan 1
menjadi 0

for kk=IterasiMulai:UkPop,
    KromMut = TempPopulasi(kk,:);
    Kromosom= TempPopulasi(kk,:);
for ii=1:JumGen,
if (rand < Pmutasi),
if Kromosom(ii)==0,
KromMut(ii) = 1;
else
KromMut(ii) = 0;
end
end
end
TempPopulasi(kk,:)=KromMut;
end
```

```
% Generational Replacement: mengganti semua kromosom sekaligus
   Populasi = TempPopulasi;
end

for ii = 1:Nvar
fprintf('Pembangkit Unit %1.0f',ii)
fprintf(' = %10.2f',Pgg(ii))
fprintf(' MW')
fprintf('    Biaya Pembangkitan Unit %1.0f',ii)
fprintf(' =  %10.2f',costv(ii))
fprintf(' Rp/jam\n')
end
fprintf('\n')
fprintf('Total Pembangkitan = %10.2f',sum(Pgg))
fprintf(' MW')
fprintf('\n')
fprintf('Total Biaya Pembangkitan1 = %10.2f',sum(costv))
fprintf(' Rp/jam\n')
fprintf('Total Losses System = %10.2f',sum(Pgg)-Pdt)
fprintf(' MW\n')
fprintf('\n')
toc
Pgg(1)
```

- **Program Metode Lagrange**

```
clear
basemva = 100;  accuracy = 0.0001;  maxiter = 10;
%         Bus Bus  Voltage Angle   ---Load---- -----Generator-----
Static Mvar
%         No  code Mag.    Degree  MW    Mvar  MW  Mvar Qmin Qmax
Qc/-Ql
busdata=[1    1    1.025   0.0     51    41     0   0     0    0
4
          2   2    1.020   0.0     22    15    79   0    40  250
0
          3   2    1.025   0.0     64    50    20   0    40  150
0
          4   2    1.050   0.0     25    10   100   0    25   80
2
          5   2    1.045   0.0     50    30   300   0    40  160
5
          6   0    1.00    0.0     76    29     0   0     0    0
2
          7   0    1.00    0.0      0     0     0   0     0    0
0
          8   0    1.00    0.0      0     0     0   0     0    0
0
          9   0    1.00    0.0     89    50     0   0     0    0
3
         10   0    1.00    0.0      0     0     0   0     0    0
0
         11   0    1.00    0.0     25    15     0   0     0    0
1.5
         12   0    1.00    0.0     89    48    00  00     0    0
2
         13   0    1.00    0.0     31    15     0   0     0    0
0
         14   0    1.00    0.0     24    12     0   0     0    0
0
         15   0    1.00    0.0     70    31     0   0     0    0
0.5
         16   0    1.00    0.0     55    27     0   0     0    0
0
         17   0    1.00    0.0     78    38     0   0     0    0
0
         18   0    1.00    0.0    153    67     0   0     0    0
0
         19   0    1.00    0.0     75    15     0   0     0    0
5
         20   0    1.00    0.0     48    27     0   0     0    0
0
         21   0    1.00    0.0     46    23     0   0     0    0
0
         22   0    1.00    0.0     45    22     0   0     0    0
0
         23   0    1.00    0.0     25    12     0   0     0    0
0
         24   0    1.00    0.0     54    27     0   0     0    0
0
```

```
        25   0    1.00    0.0     28    13     0   0     0    0
0
        26   2    1.015   0.0     40    20    60   0    15   50
0];

%                                                      Line code
%          Bus bus   R        X         1/2 B          = 1 for lines
%          nl  nr  p.u.      p.u.       p.u.           >1 or<1 tr. tap at
bus nl
linedata=[1   2   0.00055  0.00480  0.03000   1
1  18   0.00130  0.01150  0.06000   1
          2   3   0.00146  0.05130  0.05000   0.96
          2   7   0.01030  0.05860  0.01800   1
          2   8   0.00740  0.03210  0.03900   1
2  13   0.00357  0.09670  0.02500   0.96
2  26   0.03230  0.19670  0.00000   1
3  13   0.00070  0.00548  0.00050   1.017
          4   8   0.00080  0.02400  0.00010   1.050
4  12   0.00160  0.02070  0.01500   1.050
          5   6   0.00690  0.03000  0.09900   1
          6   7   0.00535  0.03060  0.00105   1
6  11   0.00970  0.05700  0.00010   1
6  18   0.00374  0.02220  0.00120   1
6  19   0.00350  0.06600  0.04500   0.95
6  21   0.00500  0.09000  0.02260   1
          7   8   0.00120  0.00693  0.00010   1
          7   9   0.00095  0.04290  0.02500   0.95
8  12   0.00200  0.01800  0.02000   1
9  10   0.00104  0.04930  0.00100   1
10  12   0.00247  0.01320  0.01000   1
10  19   0.05470  0.23600  0.00000   1
10  20   0.00660  0.01600  0.00100   1
10  22   0.00690  0.02980  0.00500   1
11  25   0.09600  0.27000  0.01000   1
11  26   0.01650  0.09700  0.00400   1
12  14   0.03270  0.08020  0.00000   1
12  15   0.01800  0.05980  0.00000   1
13  14   0.00460  0.02710  0.00100   1
13  15   0.01160  0.06100  0.00000   1
13  16   0.01793  0.08880  0.00100   1
14  15   0.00690  0.03820  0.00000   1
15  16   0.02090  0.05120  0.00000   1
16  17   0.09900  0.06000  0.00000   1
16  20   0.02390  0.05850  0.00000   1
17  18   0.00320  0.06000  0.03800   1
17  21   0.22900  0.44500  0.00000   1
19  23   0.03000  0.13100  0.00000   1
19  24   0.03000  0.12500  0.00200   1
19  25   0.11900  0.22490  0.00400   1
20  21   0.06570  0.15700  0.00000   1
20  22   0.01500  0.03660  0.00000   1
21  24   0.04760  0.15100  0.00000   1
22  23   0.02900  0.09900  0.00000   1
22  24   0.03100  0.08800  0.00000   1
23  25   0.09870  0.11680  0.00000   1];
```

```
cost = [240  7.0   0.007
200  10    0.0095
220  8.5   0.009
200  11    0.009
220  10.5  0.0080
190  12    0.0075];

mwlimits =[100  500
            50   200
            80   300
            50   150
            50   200
            50   120];

lfybus% Forms the bus admittance matrix
lfnewton% Power flow solution by Newton-Raphson method
busout% Prints the power flow solution on the screen
bloss% Obtains the loss formula coefficients
gencost% Computes the total generation cost $/h
dispatch% Obtains optimum dispatch of generation
% dpslack is the difference (absolute value) between
% the scheduled slack generation determined from the
% coordination equation, and the slack generation,
% obtained from the power flow solution.

while dpslack>.001 %Repeat till dpslack is within tolerance
lfnewton% New power flow solution
bloss% Loss coefficients are updated
dispatch% Optimum dispatch of gen. with new B-coefficients
end
busout% Prints the final power flow solution
gencost% Generation cost with optimum scheduling of gen.
```