

# **LAMPIRAN**

*Halaman ini sengaja dikosongkan*

## 1. IEEE 30 BUS

```

clear all; clc
tic
%Inisiasi Awal
alpha = 2; %Parameter
betamin = 1.0; %Parameter
gamma = 1; %Parameter
firefly = 6; %Parameter
it = 0; %Iterasi Awal
max_iterasi = 30; %iterasi maksimal

gen = 6; %Banyaknya pembangkit
Beban = 283.4; %pload
min_max = [50 200;20 80;15 50;10 35;10 30;12 40]; %minimum dan maksimum
daya pembangkit

KoefBiaya = [0.00375 2 0;0.0175 1.75 0;0.0625 1 0;
             0.00834 3.25 0;0.025 3 0;0.025 3 0]; %Matriks Koefisien Biaya

NRand = [0.22 0.26 0.3 0.34 0.38 0.42]; %Nilai random yang ditentukan
%inisiasi posisi awal partiel
for j=1:firefly %Baris 1 sampai ke 6
    for i=2:gen %Kolom 2 sampai ke 6
        P(j,i)= (min_max(i,2)-min_max(i,1))*NRand(j) + min_max(i,1);
    %Penentuan besar daya tiap pembangkit disetiap partikel(firefly)
    end
    P(j,1)=Beban-sum(P(j,2:gen)); %Penentuan besar daya untuk Pembangkit
    Pertama
end

for i=1:gen
    for j=1:firefly

F(i,j)=(KoefBiaya(i,1).*(P(j,i)).^2)+(KoefBiaya(i,2).*P(j,i))+KoefBiaya(i
,3); %Fungsi Biaya F = ap^2+bp+c untuk mencari lightbest
    end
end
Ft=sum(F)'; %Total Biaya tiap partikel

[Gbest,In]=min(Ft); %mencari total biaya minimum
lightbest =Gbest; %biaya total yang terpilih
bestPosisi=P(In,:); %partikel terbaik
Pp=P;

for i=1:firefly
    for j=1:firefly
        r=sqrt(sum((Pp(i,:)-Pp(j,:)).^2)); %jarak antar kunang-kunang
        if Ft(i)>Ft(j) %apakah lebih terang dan attractive
            beta0=1; beta=(beta0)*exp(-gamma*r.^2); %Parameter
        attractiveness kunang-kunang beta=beta0*exp(-gamma*r^2)
        tmpf=alpha.*((rand(1,1))-0.5);
        for k=2:gen
            Pp(i,k)=Pp(i,k).*(1-beta)+(Pp(j,k).*beta)+tmpf; %Penentuan
            besar daya tiap pembangkit disetiap partikel(firefly) menggunakan rumus
            firefly
        end
        Pp(i,1)=Beban-sum(Pp(i,2:gen)); %penentuan besar daya
        pembangkit 1
    end
end

```

```

end
end
%Constraint_Check
for j=1:firefly
    [Pp(j,:)] = Constraint_Check(Pp(j,:), Beban, min_max(:,1)',
min_max(:,2)', gen);
end

while it < max_iterasi %iterasi untuk algoritma firefly
    it = it + 1;
    for i=1:gen
        for j=1:firefly

Fp(i,j) = (KoeffBiaya(i,1) .* (Pp(j,i)).^2) + (KoeffBiaya(i,2) .* Pp(j,i)) + KoeffBiaya
a(i,3); %mencari fungsi biaya menggunakan daya pembangkit yang diperoleh
dari rumus firefly
        end
        end
        Fpt = sum(Fp)'; %total biaya tiap partikel

        [Gbest, In] = min(Fpt); %Total biaya minimum
        lightbest(it) = Gbest; %Total biaya yang terpilih
        bestPosisi(1,:) = Pp(In,:); %partikel yang terpilih

        for i=1:firefly
            for j=1:firefly
                r = sqrt(sum((Pp(i,:) - Pp(j,:)).^2)); %jarak antar kunang-kunang
                if Fpt(i) > Fpt(j) %apakah lebih terang dan attractive
                    beta0 = 1; beta = (beta0) * exp(-gamma * r.^2); %Parameter
attractivness kunang-kunang beta = beta0 * exp(-gamma * r.^2)
                    tmpf = alpha .* ((rand(1,1)) - 0.5);
                    for k=2:gen
                        Pp(i,k) = Pp(i,k) .* (1 - beta0) + (Pp(j,k) .* beta0) + tmpf;
%Penentuan besar daya tiap pembangkit disetiap partikel (firefly)
menggunakan rumus firefly
                    end
                    Pp(i,1) = Beban - sum(Pp(i,2:gen)); %penentuan besar daya
pembangkit 1
                    end
                end
            end
            %Constrain_Check
            for j=1:firefly
                [Pp(j,:)] = Constraint_Check(Pp(j,:), Beban, min_max(:,1)',
min_max(:,2)', gen);
            end
            H(it) = Gbest;
            it
            Gbest
        end
        History = H';
        bestPosisi
        %grafik total biaya minimum tiap iterasi
        plot(lightbest)
        xlabel('Iterasi'), ylabel('Total Biaya ($)')

```

## 2. JARINGAN 500 kV Jawa-Bali

```
clear all; clc
```

```

tic
%inisiiasi awal
alpha = 2; %Parameter
betamin = 1.0; %Parameter
gamma = 1; %Parameter
firefly = 8; %parameter
it = 0; %Iterasi Awal
max_iterasi = 100; %iterasi maksimum

gen = 8; %Banyaknya pembangkit
Beban = 10361; %Pload
min_max = [1500 3400;1425 3254;1040 2200;238 1050;600 1220;150 827;400
1008;400 700]; %matriks daya minimum dan maksimum

KoefBiaya = [65.94 395668.05 31630.21;52.19 37370.6 8220765.38;65.94
395668.05 107892572.17;
132.15 777148.777 13608770.96;21.88 197191.76 163648.18;533.92
2004960.63 86557397.4;
0 6000 0;0 395668.05 0]; %Matriks koefisien biaya

NRand = [0.4 0.42 0.44 0.46 0.48 0.5 0.52 0.54]; %Nilai random yang
ditentukan
%inisiiasi posisi awal partiel
for j=1:firefly %Baris 1 sampai ke 8
    for i=2:6 %Kolom 2 sampai ke 6
        P(j,i)= (min_max(i,2)-min_max(i,1))*NRand(j) + min_max(i,1);
    %Penentuan besar daya tiap pembangkit disetiap partikel(firefly)
    end
    P(j,7)=1000; %Penentuan besar daya untuk Pembangkit cirata
    P(j,8)=700; %Penentuan besar daya untuk Pembangkit saguling
    P(j,1)=Beban-sum(P(j,2:gen)); %Penentuan besar daya untuk Pembangkit
suralaya
end

for i=1:gen
    for j=1:firefly

F(i,j)=(KoefBiaya(i,1).*(P(j,i)).^2)+(KoefBiaya(i,2).*P(j,i))+KoefBiaya(i
,3); %Fungsi Biaya F = ap^2+bp+c untuk mencari lightbest
    end
end

Ft=sum(F)'; %Total Biaya tiap partikel

[Gbest,In]=min(Ft); %mencari total biaya minimum
lightbest =Gbest; %biaya total yang terpilih
bestPosisi=P(In,:); %partikel terbaik
Pp=P;

for i=1:firefly
    for j=1:firefly
        r=sqrt(sum((Pp(i,:)-Pp(j,:)).^2)); %jarak antar kunang-kunang
        if Ft(i)>Ft(j) %apakah lebih terang dan attractive
            beta0=1; beta=(beta0)*exp(-gamma*r.^2); %Parameter
atractivness kunang-kunang beta=beta0*exp(-gamma*r^2)
            tmpf=alpha.*((rand(1,1))-0.5);
            for k=2:6
                Pp(i,k)=Pp(i,k).*(1-beta)+(Pp(j,k).*beta)+tmpf; %Penentuan
besar daya tiap pembangkit disetiap partikel(firefly) menggunakan rumus
firefly

```

```

        end
        Pp(i,1)=Beban-sum(Pp(i,2:gen)); %penentuan besar daya
    pembangkit 1
        end
    end
end
%Constrain_Check
for j=1:firefly
    [Pp(j,:)]=Constraint_Check(Pp(j,:), Beban, min_max(:,1)',
min_max(:,2)', gen);
end

while it <max_iterasi %iterasi untuk algoritma firefly
    it =it+1;
    for i=1:gen
        for j=1:firefly

Fp(i,j)=(KoeffBiaya(i,1).*(Pp(j,i)).^2)+(KoeffBiaya(i,2).*Pp(j,i))+KoeffBiaya
a(i,3); %mencari fungsi biaya menggunakan daya pembangkit yang diperoleh
dari rumus firefly
        end
        end
        Fpt=sum(Fp)'; %total biaya tiap partikel

        [Gbest,In]=min(Fpt); %Total biaya minimum
        lightbest(it)=Gbest; %Total biaya yang teerpilih
        bestPosisi(1,:)=Pp(In,:); %partikel yang terpilih

        for i=1:firefly
            for j=1:firefly
                r=sqrt(sum((Pp(i,:)-Pp(j,:)).^2)); %jarak antar kunang-kunang
                if Fpt(i)>Fpt(j) %apakah lebih terang dan attractive
                    beta0=1; beta=(beta0)*exp(-gamma*r.^2); %Parameter
attractivness kunang-kunang beta=beta0*exp(-gamma*r.^2)
                    tmpf=alpha.*((rand(1,1))-0.5);
                    for k=2:gen
                        Pp(i,k)=Pp(i,k).*(1-beta0)+(Pp(j,k).*beta0)+tmpf;
%Penentuan besar daya tiap pembangkit disetiap partikel(firefly)
menggunakan rumus firefly
                    end
                    Pp(i,1)=Beban-sum(Pp(i,2:gen)); %penentuan besar daya
pembangkit 1
                    end
                end
            end
            %Constrain_Check
            for j=1:firefly
                [Pp(j,:)]=Constraint_Check(Pp(j,:), Beban, min_max(:,1)',
min_max(:,2)', gen);
            end
            H(it)=Gbest;
            it
            Gbest
        end
        History = H';
        bestPosisi
        %grafik total biaya minimum tiap iterasi
        plot(lightbest)
        xlabel('Iterasi'),ylabel('Total Biaya')

```

### 3. Jadwal Kamis

```

clear all; clc
tic
%inisiasi awal
alpha = 2; %Parameter
betamin = 1.0; %Parameter
gamma = 1; %Parameter
firefly = 6; %Parameter
it = 0; %Iterasi Awal
max_iterasi = 500; %iterasi maksimum

gen = 38; %Banyaknya pembangkit
%matriks beban tiap jam
Beban = [3205 3035 3035 3085 3290 2880 2790 3220
3275 3275 3275 3195 3210 3260 3357 3447 3525
3650 3820 3770 3540 3360 3345 3205];
for m=1:24 %Jam 1 sampai 24
    min_max = [225 370;225 370;53 102;53 102;53 102;53 102;53
102;53 102;
53 102;53 102;53 102;115 143;115 143;115 143;43 85;43 85;90
175;
90 175;5 16;5 16;5 16;5 16;5 16;50 95;50 95;50
95;
110 150;72 138;72 138;72 138;72 138;72 138;162 202;44 85;44
85;
44 85;90 165;90 165]; %matriks daya minimum dan maksimum

    KoefBiaya = [3244978 111712.1 10.3;3244978 111712.1
10.3;5467532.4 217963.6 34.1;
5467532.4 217963.6 34.1;5467532.4 217963.6 34.1;5467532.4
217963.6 34.1;5467532.4 217963.6 34.1;
5467532.4 217963.6 34.1;5467532.4 217963.6 34.1;5467532.4
217963.6 34.1;5467532.4 217963.6 34.1;
10936203.3 72527 368.9;10936203.3 72527 368.9;10936203.3
72527 368.9;1327126.7 217378.4 132.1;
1327126.7 217378.4 132.1;5017369.5 169242.6 193.5;5017369.5
169242.6 193.5;352707.3 350680.7 903.9;
352707.3 350680.7 903.9;352707.3 350680.7 903.9;687181.8
683240.9 1762.4;687181.8 683240.9 1762.4;
5730795 202052.9 108.1;5730795 202052.9 108.1;5730795
202052.9 108.1;11560815 53685.1 460.8;
14706521.2 433337.8 49.5;14706521.2 433337.8 49.5;14706521.2
433337.8 49.5;
14706521.2 433337.8 49.5;672630 144191.7 519.2;2417820.7
473895.4 120.8;2417820.7 473895.4 120.8;
2417820.7 473895.4 120.8;2949187.7 205217.5 83.8;2949187.7
205217.5 83.8;]; %Matriks koefisien biaya

    NRand = [0.4 0.42 0.44 0.46 0.48 0.5]; %Nilai random yang ditentukan
    %inisiasi posisi awal partiel
    for j=1:firefly %Baris 1 sampai ke 6
        for i=2:gen %Kolom 2 sampai ke 38
            P(j,i)= (min_max(i,2)-min_max(i,1))*NRand(j) + min_max(i,1);
        %Penentuan besar daya tiap pembangkit disetiap partikel(firefly)
        end
        P(j,1)=Beban(m)-sum(P(j,2:gen)); %Penentuan besar daya untuk
        Pembangkit pertama
    end
end

```

```

for i=1:gen
    for j=1:firefly

F(i,j)=KoeffBiaya(i,1)+(KoeffBiaya(i,2).*P(j,i))+(KoeffBiaya(i,3).*(P(j,i)).
^2); %Fungsi Biaya F = ap^2+bp+c untuk mencari lightbest
    end
end

Ft=sum(F)'; %Total Biaya tiap partikel

[Gbest,In]=min(Ft); %mencari total biaya minimum
lightbest =Gbest; %biaya total yang terpilih
bestPosisi=P(In,:); %partikel terbaik
Pp=P;

for i=1:firefly
    for j=1:firefly
        r=sqrt(sum((Pp(i,:)-Pp(j,:)).^2)); %jarak antar kunang-kunang
        if Ft(i)>Ft(j) %apakah lebih terang dan attractive
            beta0=1; beta=(beta0)*exp(-gamma*r.^2); %Parameter
attractivness kunang-kunang beta=beta0*exp(-gamma*r^2)
            tmpf=alpha.*(rand(1,1))-0.5;
            for k=2:6
                Pp(i,k)=Pp(i,k).*(1-beta)+(Pp(j,k).*beta)+tmpf;
%Penentuan besar daya tiap pembangkit disetiap partikel (firefly)
menggunakan rumus firefly
            end
            Pp(i,1)=Beban(m)-sum(Pp(i,2:gen)); %penentuan besar daya
pembangkit 1
        end
    end
end
%Constraint_Check
for j=1:firefly
    [Pp(j,:)]=Constraint_Check(Pp(j,:), Beban(m), min_max(:,1)',
min_max(:,2)', gen);
end

while it <max_iterasi %iterasi untuk algoritma firefly
    it =it+1;
    for i=1:gen
        for j=1:firefly

Fp(i,j)=KoeffBiaya(i,1)+(KoeffBiaya(i,2).*Pp(j,i))+(KoeffBiaya(i,3).*(Pp(j,i)
).^2); %mencari fungsi biaya menggunakan daya pembangkit yang diperoleh
dari rumus firefly
        end
    end
    Fpt=sum(Fp)'; %total biaya tiap partikel

    [Gbest,In]=min(Fpt); %Total biaya minimum
    lightbest(it)=Gbest; %Total biaya yang terpilih
    bestPosisi(1,:)=Pp(In,:); %partikel yang terpilih

    for i=1:firefly
        for j=1:firefly
            r=sqrt(sum((Pp(i,:)-Pp(j,:)).^2)); %jarak antar kunang-
kumbang
            if Fpt(i)>Fpt(j) %apakah lebih terang dan attractive

```



```

        beta0=1; beta=(beta0)*exp(-gamma*r.^2); %Parameter
    attractiveness kunang-kunang beta=beta0*exp(-gamma*r^2)
    tmpf=alpha.*((rand(1,1))-0.5);
    for k=2:gen
        Pp(i,k)=Pp(i,k).*(1-beta0)+(Pp(j,k).*beta0)+tmpf;
    %Penentuan besar daya tiap pembangkit disetiap partikel(firefly)
    menggunakan rumus firefly
    end
    Pp(i,1)=Beban(m)-sum(Pp(i,2:gen)); %penentuan besar
    daya pembangkit 1
    end
    end
    end
    %Constrain_Check
    for j=1:firefly
        [Pp(j,:)]=Constraint_Check(Pp(j,:), Beban(m), min_max(:,1)',
min_max(:,2) ', gen);
    end
    it
    Gbest
    end
    Ftpj(m)=Gbest; %Total biaya minimum tiap jam
    Po(m,:)=abs(bestPosisi); %Partikel terpilih tiap jam
end

dataonoff = dataonoff(); %pemanggilan data on off
for i=1:24 %jam 1 sampai jam 24
    for j=1:38 %pembangkit 1 sampai pembangkit 38
        if dataonoff(i,j)==0 %jika pada jam ke i pembangkit j off maka
            Po(i,j)=0; %nilai daya = 0
            Fo(i,j)=0; %fungsi biaya = 0
        else
            Fo(i,j)=KoeffBiaya(i,1)+(KoeffBiaya(j,2).*Po(i,j))+(KoeffBiaya(j,3).*(Po(i,j)
)).^2); %Fungsi biaya yang lainnya
        end
    end
    jam(i) = i;
end
Fot=sum(Fo)'; %total biaya tiap partikel
Fttotal = sum(Fot); %total biaya dari 24 jam
FA = Fot;
LR = [776761138 705621586 705621586 719392000 801061977 685123630
648335634 778544533 792479469 792479469 792479469 779497649
780598672 794364084 847649083 863101025 849444762 873033012
920727099 905112311 848997762 802277589 798842450 767137625];
%grafik perbandingan dengan metode Lagrange
plot(jam,FA,jam,LR)
xlabel('Waktu (Jam)'),ylabel('Total Biaya (Rp)')
legend('Firefly','Lagrange', 'location', 'northwest')

```

#### 4. Jadwal Sabtu

```

clear all; clc
tic

alpha = 2; %inisiasi awal
betamin = 1.0; %Parameter
gamma = 1; %Parameter
firefly = 6; %Parameter
it = 0; %Iterasi Awal

```

```

max_iterasi = 500; %iterasi maksimum

gen = 38; %Banyaknya pembangkit
%matriks beban tiap jam
Beban = [3140 3000 3000 3000 3110 2712 2682 3020
3105 3105 3105 3025 2890 2849 2806 2804 2814
3700 3700 2685 3330 3120 3095 3080];
for m=1:24 %Jam 1 sampai 24
    min_max = [225 370;225 370;53 102;53 102;53 102;53 102;53
102;53 102;
53 102;53 102;53 102;115 143;115 143;115 143;43 85;43 85;90
175;
90 175;5 16;5 16;5 16;5 16;5 16;50 95;50 95;50
95;
110 150;72 138;72 138;72 138;72 138;72 138;162 202;44 85;44
85;
44 85;90 165;90 165]; %matriks daya minimum dan maksimum

    KoefBiaya = [3244978 111712.1 10.3;3244978 111712.1
10.3;5467532.4 217963.6 34.1;
5467532.4 217963.6 34.1;5467532.4 217963.6 34.1;5467532.4
217963.6 34.1;5467532.4 217963.6 34.1;
5467532.4 217963.6 34.1;5467532.4 217963.6 34.1;5467532.4
217963.6 34.1;5467532.4 217963.6 34.1;
10936203.3 72527 368.9;10936203.3 72527 368.9;10936203.3
72527 368.9;1327126.7 217378.4 132.1;
1327126.7 217378.4 132.1;5017369.5 169242.6 193.5;5017369.5
169242.6 193.5;352707.3 350680.7 903.9;
352707.3 350680.7 903.9;352707.3 350680.7 903.9;687181.8
683240.9 1762.4;687181.8 683240.9 1762.4;
5730795 202052.9 108.1;5730795 202052.9 108.1;5730795
202052.9 108.1;11560815 53685.1 460.8;
14706521.2 433337.8 49.5;14706521.2 433337.8 49.5;14706521.2
433337.8 49.5;
14706521.2 433337.8 49.5;672630 144191.7 519.2;2417820.7
473895.4 120.8;2417820.7 473895.4 120.8;
2417820.7 473895.4 120.8;2949187.7 205217.5 83.8;2949187.7
205217.5 83.8;]; %Matriks koefisien biaya

    NRand = [0.4 0.42 0.44 0.46 0.48 0.5]; %Nilai random yang ditentukan
%inisiiasi posisi awal partiel
    for j=1:firefly %Baris 1 sampai ke 6
        for i=2:gen %Kolom 2 sampai ke 38
            P(j,i)= (min_max(i,2)-min_max(i,1))*NRand(j) + min_max(i,1);
%Penentuan besar daya tiap pembangkit disetiap partikel(firefly)
        end
        P(j,1)=Beban(m)-sum(P(j,2:gen)); %Penentuan besar daya untuk
Pembangkit pertama
    end

    for i=1:gen
        for j=1:firefly

F(i,j)=KoefBiaya(i,1)+(KoefBiaya(i,2).*P(j,i))+(KoefBiaya(i,3).*(P(j,i)).
^2); %Fungsi Biaya F = ap^2+bp+c untuk mencari lightbest
        end
    end

Ft=sum(F)'; %Total Biaya tiap partikel

```

```

[Gbest,In]=min(Ft); %mencari total biaya minimum
lightbest =Gbest; %biaya total yang terpilih
bestPosisi=P(In,:); %partikel terbaik
Pp=P;

for i=1:firefly
    for j=1:firefly
        r=sqrt(sum((Pp(i,:)-Pp(j,:)).^2)); %jarak antar kunang-kunang
        if Ft(i)>Ft(j) %apakah lebih terang dan attractive
            beta0=1; beta=(beta0)*exp(-gamma*r.^2); %Parameter
atractivness kunang-kunang beta=beta0*exp(-gamma*r^2)
            tmpf=alpha.*((rand(1,1))-0.5);
            for k=2:6
                Pp(i,k)=Pp(i,k).*(1-beta)+(Pp(j,k).*beta)+tmpf;
%Penentuan besar daya tiap pembangkit disetiap partikel(firefly)
menggunakan rumus firefly
            end
            Pp(i,1)=Beban(m)-sum(Pp(i,2:gen)); %penentuan besar daya
pembangkit 1
            end
        end
    end
    %Constraint_Check
    for j=1:firefly
        [Pp(j,:)]=Constraint_Check(Pp(j,:), Beban(m), min_max(:,1)',
min_max(:,2)', gen);
    end

    while it <max_iterasi %iterasi untuk algoritma firefly
        it =it+1;
        for i=1:gen
            for j=1:firefly

Fp(i,j)=KoeffBiaya(i,1)+(KoeffBiaya(i,2).*Pp(j,i)+(KoeffBiaya(i,3).*(Pp(j,i)
)).^2); %mencari fungsi biaya menggunakan daya pembangkit yang diperoleh
dari rumus firefly
            end
            end
            Fpt=sum(Fp)'; %total biaya tiap partikel

            [Gbest,In]=min(Fpt); %Total biaya minimum
            lightbest(it)=Gbest; %Total biaya yang terpilih
            bestPosisi(1,:)=Pp(In,:); %partikel yang terpilih

            for i=1:firefly
                for j=1:firefly
                    r=sqrt(sum((Pp(i,:)-Pp(j,:)).^2)); %jarak antar kunang-
kunang
                    if Fpt(i)>Fpt(j) %apakah lebih terang dan attractive
                        beta0=1; beta=(beta0)*exp(-gamma*r.^2); %Parameter
atractivness kunang-kunang beta=beta0*exp(-gamma*r^2)
                        tmpf=alpha.*((rand(1,1))-0.5);
                        for k=2:gen
                            Pp(i,k)=Pp(i,k).*(1-beta0)+(Pp(j,k).*beta0)+tmpf;
%Penentuan besar daya tiap pembangkit disetiap partikel(firefly)
menggunakan rumus firefly
                        end
                            Pp(i,1)=Beban(m)-sum(Pp(i,2:gen)); %penentuan besar
daya pembangkit 1
                        end
                    end
                end
            end

```

```

end
%Constrain_Check
for j=1:firefly
    [Pp(j,:)] = Constraint_Check(Pp(j,:), Beban(m), min_max(:,1)',
min_max(:,2)', gen);
end
it
Gbest
end
Ftpj(m) = Gbest; %Total biaya minimum tiap jam
Po(m,:) = abs(bestPosisi); %Partikel terpilih tiap jam
end

dataonoff1 = dataonoff1(); %pemanggilan data on off
for i=1:24 %pemanggilan data on off
    for j=1:38 %jam 1 sampai jam 24
        if dataonoff1(i,j) == 0 %jika pada jam ke i pembangkit j off maka
            Po(i,j) = 0; %nilai daya = 0
            Fo(i,j) = 0; %fungsi biaya = 0
        else
            Fo(i,j) = KoefBiaya(i,1) + (KoefBiaya(j,2) .* Po(i,j)) + (KoefBiaya(j,3) .* (Po(i,j)
).^2); %Fungsi biaya yang lainnya
        end
    end
    jam(i) = i;
end
Fot = sum(Fo'); %total biaya tiap partikel
Fttotal = sum(Fot); %total biaya dari 24 jam
FA = Fot;
LR = [776761138 705621586 705621586 719392000 801061977 685123630
648335634 778544533 792479469 792479469 792479469 779497649
780598672 794364084 847649083 863101025 849444762 873033012
920727099 905112311 848997762 802277589 798842450 767137625];
%grafik perbandingan dengan metode Lagrange
plot(jam, FA, jam, LR)
xlabel('Waktu (Jam)'), ylabel('Total Biaya (Rp)')
legend('Firefly', 'Lagrange', 'location', 'northwest')

```

## 5. Jadwal Minggu

```

clear all; clc
tic

alpha = 2; %inisiasi awal
betamin = 1.0; %Parameter
gamma = 1; %Parameter
firefly = 6; %Parameter
it = 0; %Iterasi Awal
max_iterasi = 500; %iterasi maksimum

gen = 38; %Banyaknya pembangkit
%matris beban tiap jam
Beban = [2956 2860 2860 2860 2869 2869 2640 2450
2520 2620 2620 2570 2570 2545 2587 2587 2790
3685 3685 3685 3370 3220 3195 3050];
for m=1:24 %Jam 1 sampai 24
    min_max = [225 370; 225 370; 53 102; 53 102; 53 102; 53 102; 53 102; 53 102; 53 102; 53 102];
end

```

```

53 102;53 102;53 102;115 143;115 143;115 143;43 85;43 85;90
175;
90 175;5 16;5 16;5 16;5 16;5 16;50 95;50 95;50
95;
110 150;72 138;72 138;72 138;72 138;72 138;162 202;44 85;44
85;
44 85;90 165;90 165]; %matriks daya minimum dan maksimum

KoefBiaya = [3244978 111712.1 10.3;3244978 111712.1
10.3;5467532.4 217963.6 34.1;
5467532.4 217963.6 34.1;5467532.4 217963.6 34.1;5467532.4
217963.6 34.1;5467532.4 217963.6 34.1;
5467532.4 217963.6 34.1;5467532.4 217963.6 34.1;5467532.4
217963.6 34.1;5467532.4 217963.6 34.1;
10936203.3 72527 368.9;10936203.3 72527 368.9;10936203.3
72527 368.9;1327126.7 217378.4 132.1;
1327126.7 217378.4 132.1;5017369.5 169242.6 193.5;5017369.5
169242.6 193.5;352707.3 350680.7 903.9;
352707.3 350680.7 903.9;352707.3 350680.7 903.9;687181.8
683240.9 1762.4;687181.8 683240.9 1762.4;
5730795 202052.9 108.1;5730795 202052.9 108.1;5730795
202052.9 108.1;11560815 53685.1 460.8;
14706521.2 433337.8 49.5;14706521.2 433337.8 49.5;14706521.2
433337.8 49.5;14706521.2 433337.8 49.5;
14706521.2 433337.8 49.5;672630 144191.7 519.2;2417820.7
473895.4 120.8;2417820.7 473895.4 120.8;
2417820.7 473895.4 120.8;2949187.7 205217.5 83.8;2949187.7
205217.5 83.8]; %Matriks koefisien biaya

NRand = [0.4 0.42 0.44 0.46 0.48 0.5]; %Nilai random yang ditentukan
%inisiasi posisi awal partiel
for j=1:firefly %Baris 1 sampai ke 6
    for i=2:gen %Kolom 2 sampai ke 38
        P(j,i) = (min_max(i,2)-min_max(i,1))*NRand(j) + min_max(i,1);
%Penentuan besar daya tiap pembangkit disetiap partikel(firefly)
    end
    P(j,1) = Beban(m) - sum(P(j,2:gen)); %Penentuan besar daya untuk
Pembangkit pertama
end

for i=1:gen
    for j=1:firefly

F(i,j) = KoefBiaya(i,1) + (KoefBiaya(i,2) .* P(j,i)) + (KoefBiaya(i,3) .* (P(j,i)).
^2); %Fungsi Biaya F = ap^2+bp+c untuk mencari lightbest
    end
end

Ft = sum(F)'; %Total Biaya tiap partikel

[Gbest, In] = min(Ft); %mencari total biaya minimum
lightbest = Gbest; %biaya total yang terpilih
bestPosisi = P(In,:); %partikel terbaik
Pp = P;

for i=1:firefly
    for j=1:firefly
        r = sqrt(sum((Pp(i,:) - Pp(j,:)).^2)); %jarak antar kunang-kunang
        if Ft(i) > Ft(j) %apakah lebih terang dan attractive

```

```

        beta0=1; beta=(beta0)*exp(-gamma*r.^2); %Parameter
atractivness kunang-kunang beta=beta0*exp(-gamma*r^2)
        tmpf=alpha.*((rand(1,1))-0.5);
        for k=2:6
            Pp(i,k)=Pp(i,k).*(1-beta)+(Pp(j,k).*beta)+tmpf;
%Penentuan besar daya tiap pembangkit disetiap partikel(firefly)
menggunakan rumus firefly
        end
        Pp(i,1)=Beban(m)-sum(Pp(i,2:gen)); %penentuan besar daya
pembangkit 1
    end
end
    end
    %Constraint_Check
    for j=1:firefly
        [Pp(j,:)]=Constraint_Check(Pp(j,:), Beban(m), min_max(:,1)',
min_max(:,2)', gen);
    end

    while it <max_iterasi %iterasi untuk algoritma firefly
        it =it+1;
        for i=1:gen
            for j=1:firefly

Fp(i,j)=KoefBiaya(i,1)+(KoefBiaya(i,2).*Pp(j,i))+(KoefBiaya(i,3).*(Pp(j,i)
).^2); %mencari fungsi biaya menggunakan daya pembangkit yang diperoleh
dari rumus firefly
            end
            end
            Fpt=sum(Fp)'; %total biaya tiap partikel

            [Gbest,In]=min(Fpt); %Total biaya minimum
            lightbest(it)=Gbest; %Total biaya yang terpilih
            bestPosisi(1,:)=Pp(In,:); %partikel yang terpilih

            for i=1:firefly
                for j=1:firefly
                    r=sqrt(sum((Pp(i,:)-Pp(j,:)).^2)); %jarak antar kunang-
kunang
                    if Fpt(i)>Fpt(j) %apakah lebih terang dan attractive
                        beta0=1; beta=(beta0)*exp(-gamma*r.^2); %Parameter
atractivness kunang-kunang beta=beta0*exp(-gamma*r^2)
                        tmpf=alpha.*((rand(1,1))-0.5);
                        for k=2:gen
                            Pp(i,k)=Pp(i,k).*(1-beta0)+(Pp(j,k).*beta0)+tmpf;
%Penentuan besar daya tiap pembangkit disetiap partikel(firefly)
menggunakan rumus firefly
                        end
                        Pp(i,1)=Beban(m)-sum(Pp(i,2:gen)); %penentuan besar
daya pembangkit 1
                    end
                end
            end
            end
            %Constrain_Check
            for j=1:firefly
                [Pp(j,:)]=Constraint_Check(Pp(j,:), Beban(m), min_max(:,1)',
min_max(:,2)', gen);
            end
            it
            Gbest
        end
    end
end

```

```

Ftpj(m)=Gbest; %Total biaya minimum tiap jam
Po(m,:)=abs(bestPosisi); %Partikel terpilih tiap jam
end

dataonoff2 = dataonoff2(); %pemanggilan data on off
for i=1:24 %pemanggilan data on off
    for j=1:38 %jam 1 sampai jam 24
        if dataonoff2(i,j)==0 %jika pada jam ke i pembangkit j off maka
            Po(i,j)=0; %nilai daya = 0
            Fo(i,j)=0; %fungsi biaya = 0
        else
            Fo(i,j)=KoeffBiaya(i,1)+(KoeffBiaya(j,2).*Po(i,j))+(KoeffBiaya(j,3).*(Po(i,j)
            )).^2); %Fungsi biaya yang lainnya
        end
    end
    jam(i) = i;
end
Fot=sum(Fo)'; %total biaya tiap partikel
Ftttotal = sum(Fot); %total biaya dari 24 jam
FA = Fot;
LR = [776761138 705621586 705621586 719392000 801061977 685123630
648335634 778544533 792479469 792479469 792479469 779497649
780598672 794364084 847649083 863101025 849444762 873033012
920727099 905112311 848997762 802277589 798842450 767137625];
%grafik perbandingan dengan metode Lagrange
plot(jam,FA,jam,LR)
xlabel('Waktu (Jam)'),ylabel('Total Biaya (Rp)')
legend('Firefly','Lagrange','location','northwest')

```