

DAFTAR PUSTAKA

- [1] Andika rahmat Ferdian , Siti Arbaiyah, Diniy Istiyanto S.N. 2012. Pengantar Kecerdasan Buatan . Semarang: dinyistianto.blogspot.com
- [2] Anesya violita, Ardyono Priyadi dan Imam robandi. 2012. Optimasi Economic Dispatch pada sistem kelistrikan jawa bali 500 kv menggunakan Differential Evolutionary algorithm.Surabaya: jurnal teknik ITS Vol 1.
- [3] Cekdin, cekmas.2006. Sistem tenaga listrik. Palembang : Andi.
- [4] [PDF] 2013-2-00404-MTIFBab2001”Landasanteori”[https:// www.Scribd.com > document2013](https://www.Scribd.com/document/2013-2-00404-MTIFBab2001). diakses pada 11 Juni 2017
- [5] [PDF] Swarm Intelligent-ResearchGate ”SWARM INTELLIGENCE (Teori &CaseStudy)”PSO,ACO,ABC,ACOSVR,etc’ .[http://researchgate.net>buk....](http://researchgate.net/buk...), diakses pada 17 Juli 2017

DAFTAR RIWAYAT HIDUP

A. Biodata Pribadi

1. Nama lengkap : M. Syaifuddin Zuhri
2. Tempat/ tanggal lahir : Sidoarjo, 26 September 1988
3. Jenis kelamin : Laki – laki
4. Kewarganegaraan : Indonesia
5. Tinggi, Berat Badan : 165 cm, 68 kg
6. Agama : Islam
7. Status : Belum menikah
8. Alamat : Bhumi Jati Permai blok BB no.07
Jati Rembe, Benjeng Gresik
9. No. Telp : 081357707479
10. Email : Syaifuddin.5077@yahoo.com



B. Riwayat Pendidikan

1. TK : TK Muslimah XII Sidoarjo 1994-1995
2. MI : MI Nur Rohman Sambi Bulu Sidoarjo 1995-2001
3. SMP : SMP Negeri 2 Sukodono Sidoarjo 2001-2003
4. SMK : SMK Negeri 3 Surabaya 2003-2006

Lampiran :

```
%-----SKRIPSI-----%
%---PARTICLE SWARM OPTIMIZATION---%
%----M.Syaifuddin Zuhri/11.632.013----%

clc
clear all
%Initiation parameter
w_awal=0.1;
w_akhir=0.9;
w=0.5;
c1=1.2;
c2=1.2;
pload=54.71;
%velocity awal
velocity_initial=zeros(6,30);
n_par=30;
n_iter=10;
mwlimit=[4 25
         4 14
         2 8
         2 9.5
         4 16.5
         1 5.5];

%cost uniturut paling mahal
costunit=[659.7977 1.5332 0.2700
          -0.3276 24.0480 -0.0036
           9.0869 16.7520 0.8420
          -0.2883 8.6040 -0.0066
           0.0050 6.3503 0.0002
          -0.1872 23.9940 0.0072];

%fuel cost unit
fuelcost=[1.0
          1.0
          1.0
          1.0
          1.0];

%inisiasi partikel
[x,y]=size(mwlimit);
[xx,yy]=size(costunit);
%total cost unit=fuel cost*cost unit
for ng=1:xx
    for ngg=1:yy
        cost(ng,ngg)=fuelcost(ng)*costunit(ng,ngg);
    end
end
cost
for ng=1:x % matriks baris / jumlah pembangkit
    for ngg=1:n_par % matriks kolom / jumlah partikel
        P_part(ng,ngg)=(mwlimit(ng,2)-mwlimit(ng,1))*rand
        + mwlimit(ng,1); % Daya pembangkit 1, partikel ke-1-
        30, dst dan daya pembangkit 2, partikel ke-1-30
    end
end
P_partinitial1=P_part;
for ngg=1:n_par
    error(1,ngg)=pload-sum(P_part(:,ngg)); % Beban -
    jumlah total daya pembangkit pada partikel 1 dst / error
    pd partikel ke-1 dst
end
error;
%cek total p1 p2 p3
for ngg=1:n_par
```

```
Psum_awal(1,ngg)=sum(P_partinitial1(:,ngg));
end
Psum_awal;
%update berdasar error
for ng=1:x % jmlh pembangkit
    for ngg=1:n_par
        P_partinitial2(ng,ngg)=P_partinitial1(ng,ngg)+error(ng
        g)/x;
    end
end
P_partinitial2;
P_sblm_batas = P_partinitial2;
%cek total p1 p2 p3
for ngg=1:n_par
    Psum_sblm_batas(1,ngg)=sum(P_partinitial2(:,ngg));
end
Psum_sblm_batas;
% n_par= 1;
% x=4 ;
%
% mwlimit=[330 660
           330 660
           330 660
           330 660];
%
% P_part = [ 700
            200
            700
            200 ] ;
%update p1 p2 p3 agar sesuai mwlimit
for aaa=1:25
    for ngg=1:n_par
        if
            or(P_partinitial2(1,ngg)<mwlimit(1,1),P_partinitial2(1,
            ngg)>mwlimit(1,2));
                if P_partinitial2(1,ngg)<mwlimit(1,1);
                    error2=P_partinitial2(1,ngg)-mwlimit(1,1);
                    P_partinitial2(1,ngg)=mwlimit(1,1);
                end
            end
        P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-
        1);
        P_partinitial2(3,ngg)=P_partinitial2(3,ngg)+error2/(x-
        1);
        P_partinitial2(4,ngg)=P_partinitial2(4,ngg)+error2/(x-
        1);
        P_partinitial2(5,ngg)=P_partinitial2(5,ngg)+error2/(x-
        1);
        P_partinitial2(6,ngg)=P_partinitial2(6,ngg)+error2/(x-
        1);
        if P_partinitial2(5,ngg)<mwlimit(5,1);
            error2=P_partinitial2(5,ngg)-mwlimit(5,1);
            P_partinitial2(5,ngg)=mwlimit(5,1);
        end
        P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-
        2);
        P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-
        2);
        P_partinitial2(3,ngg)=P_partinitial2(3,ngg)+error2/(x-
        2);
        P_partinitial2(4,ngg)=P_partinitial2(4,ngg)+error2/(x-
        2);
```



```

P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-2);

P_partinitial2(3,ngg)=P_partinitial2(3,ngg)+error2/(x-2);

P_partinitial2(4,ngg)=P_partinitial2(4,ngg)+error2/(x-2);
    if P_partinitial2(4,ngg)< mwlimit(4,1);
        error2=P_partinitial2(4,ngg)-
mwlimit(4,1);
        P_partinitial2(4,ngg)=mwlimit(4,1);

P_partinitial2(3,ngg)=P_partinitial2(3,ngg)+error2/(x-3);

P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-3);

P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-3);
    if P_partinitial2(3,ngg)< mwlimit(3,1);
        error2=P_partinitial2(3,ngg)-
mwlimit(3,1);
        P_partinitial2(3,ngg)=mwlimit(3,1);
P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-4);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-4);
    if
        P_partinitial2(2,ngg)<
mwlimit(2,1);
        error2=P_partinitial2(2,ngg)-
mwlimit(2,1);
        P_partinitial2(2,ngg)=mwlimit(2,1);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-5);
    else if P_partinitial2(2,ngg)>
mwlimit(2,2);
        error2=P_partinitial2(2,ngg)-
mwlimit(2,2);
        P_partinitial2(2,ngg)=mwlimit(2,2);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-5);
    end
    end
    if
P_partinitial2(3,ngg)>mwlimit(3,2);
        error2=P_partinitial2(3,ngg)-
mwlimit(3,2);
        P_partinitial2(3,ngg)=mwlimit(3,2);

P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-4);

P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-4);
    if P_partinitial2(2,ngg)< mwlimit(2,1);
        error2=P_partinitial2(2,ngg)-
mwlimit(2,1);
        P_partinitial2(2,ngg)=mwlimit(2,1);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-5);
    else if P_partinitial2(2,ngg)>
mwlimit(2,2);
        error2=P_partinitial2(2,ngg)-
mwlimit(2,2);
        P_partinitial2(2,ngg)=mwlimit(2,2);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-5);
    end
    end
P_partinitial2(2,ngg)=mwlimit(2,2);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-5);
    end
end

```

```

        end
        end
        end
        end
    end
    else if P_partinitial2(1,ngg)>mwlimit(1,2);
        error2=P_partinitial2(1,ngg)-mwlimit(1,2);
        P_partinitial2(1,ngg)=mwlimit(1,2);
P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-1);
P_partinitial2(3,ngg)=P_partinitial2(3,ngg)+error2/(x-1);
P_partinitial2(4,ngg)=P_partinitial2(4,ngg)+error2/(x-1);
P_partinitial2(5,ngg)=P_partinitial2(5,ngg)+error2/(x-1);
P_partinitial2(6,ngg)=P_partinitial2(6,ngg)+error2/(x-1);
        if P_partinitial2(5,ngg)<mwlimit(5,1);
            error2=P_partinitial2(5,ngg)-mwlimit(5,1);
            P_partinitial2(5,ngg)=mwlimit(5,1);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-2);
P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-2);
P_partinitial2(3,ngg)=P_partinitial2(3,ngg)+error2/(x-2);
P_partinitial2(4,ngg)=P_partinitial2(4,ngg)+error2/(x-2);
        if P_partinitial2(4,ngg)<mwlimit(4,1);
            error2=P_partinitial2(4,ngg)-mwlimit(4,1);
            P_partinitial2(4,ngg)=mwlimit(4,1);
P_partinitial2(3,ngg)=P_partinitial2(3,ngg)+error2/(x-3);
P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-3);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-3);
        if P_partinitial2(3,ngg)<mwlimit(3,1);
            error2=P_partinitial2(3,ngg)-mwlimit(3,1);
            P_partinitial2(3,ngg)=mwlimit(3,1);
P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-4);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-4);
        if P_partinitial2(2,ngg)<mwlimit(2,1);
            error2=P_partinitial2(2,ngg)-mwlimit(2,1);
            P_partinitial2(2,ngg)=mwlimit(2,1);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-5);
        else if P_partinitial2(2,ngg)>
mwlimit(2,2);

```

```

        error2=P_partinitial2(2,ngg)-mwlimit(2,2);
        P_partinitial2(2,ngg)=mwlimit(2,2);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-5);
        end
        end
        else if P_partinitial2(3,ngg)>mwlimit(3,2);
            error2=P_partinitial2(3,ngg)-mwlimit(3,2);
            P_partinitial2(3,ngg)=mwlimit(3,2);
P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-4);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-4);
        if P_partinitial2(2,ngg)<mwlimit(2,1);
            error2=P_partinitial2(2,ngg)-mwlimit(2,1);
            P_partinitial2(2,ngg)=mwlimit(2,1);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-5);
        else if P_partinitial2(2,ngg)>mwlimit(2,2);
            error2=P_partinitial2(2,ngg)-mwlimit(2,2);
            P_partinitial2(2,ngg)=mwlimit(2,2);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-5);
        end
        end
        end
        else if P_partinitial2(4,ngg)>mwlimit(4,2);
            error2=P_partinitial2(4,ngg)-mwlimit(4,2);
            P_partinitial2(4,ngg)=mwlimit(4,2);
P_partinitial2(3,ngg)=P_partinitial2(3,ngg)+error2/(x-3);
P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-3);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-3);
        if P_partinitial2(3,ngg)<mwlimit(3,1);
            error2=P_partinitial2(3,ngg)-mwlimit(3,1);
            P_partinitial2(3,ngg)=mwlimit(3,1);
P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-4);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-4);
        if P_partinitial2(2,ngg)<mwlimit(2,1);
            error2=P_partinitial2(2,ngg)-mwlimit(2,1);
            P_partinitial2(2,ngg)=mwlimit(2,1);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-5);
        else if P_partinitial2(2,ngg)>mwlimit(2,2);

```



```

end
end
end
end
end
end
end
end
if
or(P_partinitial2(4,ngg)<mwlimit(4,1),P_partinitial2(4,
ngg)>mwlimit(4,2));
if P_partinitial2(4,ngg)<mwlimit(4,1);
error2=P_partinitial2(4,ngg)-mwlimit(4,1);
P_partinitial2(4,ngg)=mwlimit(4,1);

P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-
1);

P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-
1);

P_partinitial2(3,ngg)=P_partinitial2(3,ngg)+error2/(x-
1);

P_partinitial2(5,ngg)=P_partinitial2(5,ngg)+error2/(x-
1);

P_partinitial2(6,ngg)=P_partinitial2(6,ngg)+error2/(x-
1);
if P_partinitial2(5,ngg)<mwlimit(5,1);
error2=P_partinitial2(5,ngg)-mwlimit(5,1);
P_partinitial2(5,ngg)=mwlimit(5,1);

P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-
2);

P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-
2);

P_partinitial2(3,ngg)=P_partinitial2(3,ngg)+error2/(x-
2);

P_partinitial2(4,ngg)=P_partinitial2(4,ngg)+error2/(x-
2);
if P_partinitial2(4,ngg)< mwlimit(4,1);
error2=P_partinitial2(4,ngg)-mwlimit(4,1);
P_partinitial2(4,ngg)=mwlimit(4,1);

P_partinitial2(3,ngg)=P_partinitial2(3,ngg)+error2/(x-
3);

P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-
3);

P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-
3);
if P_partinitial2(3,ngg)< mwlimit(3,1);
error2=P_partinitial2(3,ngg)-
mwlimit(3,1);
P_partinitial2(3,ngg)=mwlimit(3,1);

P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-
4);

P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-
4);
if P_partinitial2(2,ngg)< mwlimit(2,1);

```

```

error2=P_partinitial2(2,ngg)-
mwlimit(2,1);
P_partinitial2(2,ngg)=mwlimit(2,1);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-
5);
else if P_partinitial2(2,ngg)>
mwlimit(2,2);
error2=P_partinitial2(2,ngg)-
mwlimit(2,2);
P_partinitial2(2,ngg)=mwlimit(2,2);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-
5);
end
end
else if P_partinitial2(3,ngg)>mwlimit(3,2);
error2=P_partinitial2(3,ngg)-
mwlimit(3,2);
P_partinitial2(3,ngg)=mwlimit(3,2);

P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-
4);

P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-
4);
if P_partinitial2(2,ngg)< mwlimit(2,1);
error2=P_partinitial2(2,ngg)-
mwlimit(2,1);
P_partinitial2(2,ngg)=mwlimit(2,1);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-
5);
else if P_partinitial2(2,ngg)>
mwlimit(2,2);
error2=P_partinitial2(2,ngg)-
mwlimit(2,2);
P_partinitial2(2,ngg)=mwlimit(2,2);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-
5);
end
end
end
end
else if P_partinitial2(4,ngg)>mwlimit(4,2);
error2=P_partinitial2(4,ngg)-mwlimit(4,2);
P_partinitial2(4,ngg)=mwlimit(4,2);

P_partinitial2(3,ngg)=P_partinitial2(3,ngg)+error2/(x-
3);

P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-
3);

P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-
3);
if P_partinitial2(3,ngg)< mwlimit(3,1);
error2=P_partinitial2(3,ngg)-
mwlimit(3,1);
P_partinitial2(3,ngg)=mwlimit(3,1);

P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-
4);

P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-
4);
if P_partinitial2(2,ngg)< mwlimit(2,1);
error2=P_partinitial2(2,ngg)-
mwlimit(2,1);
P_partinitial2(2,ngg)=mwlimit(2,1);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-
5);

```

```

else if P_partinitial2(2,ngg)>
mwlmit(2,2);
error2=P_partinitial2(2,ngg)-
mwlmit(2,2);
P_partinitial2(2,ngg)=mwlmit(2,2);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-
5);
end
end
else if P_partinitial2(3,ngg)>mwlmit(3,2);
error2=P_partinitial2(3,ngg)-
mwlmit(3,2);
P_partinitial2(3,ngg)=mwlmit(3,2);

P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-
4);

P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-
4);
if P_partinitial2(2,ngg)< mwlmit(2,1);
error2=P_partinitial2(2,ngg)-
mwlmit(2,1);
P_partinitial2(2,ngg)=mwlmit(2,1);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-
5);
else if P_partinitial2(2,ngg)>
mwlmit(2,2);
error2=P_partinitial2(2,ngg)-
mwlmit(2,2);
P_partinitial2(2,ngg)=mwlmit(2,2);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-
5);
end
end
end
end
end
end
else if P_partinitial2(5,ngg)> mwlmit(5,2);
error2=P_partinitial2(5,ngg)-mwlmit(5,2);
P_partinitial2(5,ngg)=mwlmit(5,2);

P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-
2);

P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-
2);

P_partinitial2(3,ngg)=P_partinitial2(3,ngg)+error2/(x-
2);

P_partinitial2(4,ngg)=P_partinitial2(4,ngg)+error2/(x-
2);
if P_partinitial2(4,ngg)< mwlmit(4,1);
error2=P_partinitial2(4,ngg)-
mwlmit(4,1);
P_partinitial2(4,ngg)=mwlmit(4,1);

P_partinitial2(3,ngg)=P_partinitial2(3,ngg)+error2/(x-
3);

P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-
3);

P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-
3);
if P_partinitial2(3,ngg)< mwlmit(3,1);
error2=P_partinitial2(3,ngg)-
mwlmit(3,1);

```

```

P_partinitial2(3,ngg)=mwlmit(3,1);
P_partinitial2(2,ngg)=P_partinitial2(2,ngg)+error2/(x-
4);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-
4);
if P_partinitial2(2,ngg)<
mwlmit(2,1);
error2=P_partinitial2(2,ngg)-
mwlmit(2,1);
P_partinitial2(2,ngg)=mwlmit(2,1);
P_partinitial2(1,ngg)=P_partinitial2(1,ngg)+error2/(x-
5);
else if P_partinitial2(2,ngg)>
mwlmit(2,2);
P_part(3,ngg)=P_part(3,ngg)+error2/(x-2);
P_part(4,ngg)=P_part(4,ngg)+error2/(x-2);
if P_part(4,ngg)< mwlmit(4,1);
error2=P_part(4,ngg)-mwlmit(4,1);
P_part(4,ngg)=mwlmit(4,1);
P_part(3,ngg)=P_part(3,ngg)+error2/(x-3);
P_part(2,ngg)=P_part(2,ngg)+error2/(x-3);
P_part(1,ngg)=P_part(1,ngg)+error2/(x-3);
if P_part(3,ngg)< mwlmit(3,1);
error2=P_part(3,ngg)-mwlmit(3,1);
P_part(3,ngg)=mwlmit(3,1);
P_part(2,ngg)=P_part(2,ngg)+error2/(x-
4);
P_part(1,ngg)=P_part(1,ngg)+error2/(x-
4);
if P_part(2,ngg)< mwlmit(2,1);
error2=P_part(2,ngg)-mwlmit(2,1);
P_part(2,ngg)=mwlmit(2,1);

P_part(1,ngg)=P_part(1,ngg)+error2/(x-5);
else if P_part(2,ngg)> mwlmit(2,2);
error2=P_part(2,ngg)-mwlmit(2,2);
P_part(2,ngg)=mwlmit(2,2);

P_part(1,ngg)=P_part(1,ngg)+error2/(x-5);
end
end
else if P_part(3,ngg)>mwlmit(3,2);
error2=P_part(3,ngg)-mwlmit(3,2);
P_part(3,ngg)=mwlmit(3,2);
P_part(2,ngg)=P_part(2,ngg)+error2/(x-
4);
P_part(1,ngg)=P_part(1,ngg)+error2/(x-
4);
if P_part(2,ngg)< mwlmit(2,1);
error2=P_part(2,ngg)-mwlmit(2,1);
P_part(2,ngg)=mwlmit(2,1);

P_part(1,ngg)=P_part(1,ngg)+error2/(x-5);
else if P_part(2,ngg)> mwlmit(2,2);
error2=P_part(2,ngg)-mwlmit(2,2);
P_part(2,ngg)=mwlmit(2,2);

P_part(1,ngg)=P_part(1,ngg)+error2/(x-5);
end
end
end
end
else if P_part(4,ngg)>mwlmit(4,2);
error2=P_part(4,ngg)-mwlmit(4,2);
P_part(4,ngg)=mwlmit(4,2);
P_part(3,ngg)=P_part(3,ngg)+error2/(x-3);
P_part(2,ngg)=P_part(2,ngg)+error2/(x-3);
P_part(1,ngg)=P_part(1,ngg)+error2/(x-3);
if P_part(3,ngg)< mwlmit(3,1);

```



```

        P_part(2,ngg)=mwlimit(2,2);
P_part(1,ngg)=P_part(1,ngg)+error2/(x-5);
        end
        end
        end
        end
        end
        end
        end
        end
        end
        P_part

%cek total p1 p2 p3 setelah update
for ngg=1:n_par
    Psum5(1,ngg)=sum(P_part(:,ngg));
end
Psum5

% %cek anggota partikel apa ada yang melanggar
mwlimit
in_constr=0;
for ng=1:x
    for ngg=1:n_par
        if
and(P_part(ng,ngg)<=mwlimit(ng,2),P_part(ng,ngg)>=
mwlimit(ng,1))
            in_constritung=0;
            in_constr=in_constr+in_constritung;
        else
            in_constritung=1;
            in_constr=in_constr+in_constritung;
        end
    end
end

if in_constr~=0;
    fprintf('salah')
else
    fprintf('benar')
end
%end sub pengecekan partikel terhadap mwlimit
end

Gbest_fix
gencost1 =
cost(1,1)+cost(1,2)*Gbest_fix(1,1)+cost(1,3)*(Gbest_fix(1,1)^2);
gencost2 =
cost(2,1)+cost(2,2)*Gbest_fix(2,1)+cost(2,3)*(Gbest_fix(2,1)^2);
gencost3 =
cost(3,1)+cost(3,2)*Gbest_fix(3,1)+cost(3,3)*(Gbest_fix(3,1)^2);
gencost4 =
cost(4,1)+cost(4,2)*Gbest_fix(4,1)+cost(4,3)*(Gbest_fix(4,1)^2);
gencost5 =
cost(5,1)+cost(5,2)*Gbest_fix(5,1)+cost(5,3)*(Gbest_fix(5,1)^2);
gencost6 =
cost(6,1)+cost(6,2)*Gbest_fix(6,1)+cost(6,3)*(Gbest_fix(6,1)^2);

gencost_Gbest_fix= gencost1 + gencost2 + gencost3 +
gencost4 + gencost5 + gencost6 ;
fprintf('\nTotal generation cost = % 10.2f juta/h \n',
gencost_Gbest_fix)

```


HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini :

Nama : M. Syaifuddin Zuhri
NIM : 11.632.013
Tempat, Tanggal Lahir : Sidoarjo, 26 September 1988
Institusi : Universitas Muhammadiyah Gresik

Menyatakan dengan sesungguhnya bahwa Skripsi yang berjudul :

**" SIMULASI ECONOMIC DISPATCH PADA PEMBANGKIT LISTRIK
TENAGA GAS DAN UAP DI PT. PETROKIMIA GRESIK
MENGUNAKAN *PARTICLE SWARM OPTIMIZATION* (PSO) "**

Benar benar merupakan hasil karya yang saya buat sendiri berdasarkan penelitian yang telah saya lakukan (*bukan plagiat*).

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya dan apabila pernyataan ini tidak benar, saya bersedia mendapatkan sanksi.

Gresik, 26 April 2018



Menyatakan,

M. Syaifuddin Zuhri
NIM. 11.632.013