

REPUBLIK INDONESIA KEMENTERIAN HUKUM  
DAN HAK ASASI MANUSIA

# SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202421342, 6 Maret 2024

## Pencipta

Nama : **Ikrima Alfi, S.T., M.Eng.**  
Alamat : Jalan Kurma 211 Mundu Tempel RT 06 RW 02 Caturtunggal Depok,  
Depok, Sleman, DI Yogyakarta, 55283  
Kewarganegaraan : Indonesia

## Pemegang Hak Cipta

Nama : **Universitas Teknologi Yogyakarta**  
Alamat : Jl. Siliwangi Jl. Jombor Lor, Mlati Krajan, Sendangadi, Kec. Mlati,  
Kabupaten Sleman, Daerah Istimewa Yogyakarta 55284, Mlati, Sleman,  
Di Yogyakarta 55284

Kewarganegaraan : Indonesia

Jenis Ciptaan : **Program Komputer**

Judul Ciptaan : **KODE PROGRAM UNTUK OPTIMASI PERENCANAAN  
PERLUASAN JARINGAN TRANSMISI DAYA LISTRIK  
MENGUNAKAN ALGORITMA GENETIKA**

Tanggal dan tempat diumumkan untuk pertama kali : 6 Maret 2024, di Yogyakarta  
di wilayah Indonesia atau di luar wilayah Indonesia

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali  
dilakukan Pengumuman.

Nomor pencatatan : 000596700

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.

Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA  
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL  
u.b

Direktur Hak Cipta dan Desain Industri

Anggoro Dasananto  
NIP. 196412081991031002

## Disclaimer:

Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.

**KODE PROGRAM UNTUK OPTIMASI PERENCANAAN  
PERLUASAN JARINGAN TRANSMISI DAYA LISTRIK  
MENGUNAKAN ALGORITMA GENETIKA**

**Oleh:  
Ikrima Alfi, S.T., M.Eng.**

**UNIVERSITAS TEKNOLOGI YOGYAKARTA  
2024**

## SPESIFIKASI SISTEM

a. Perangkat lunak MATLAB R2021a

MATLAB merupakan kependekan dari MATrix LABoratory dikarenakan setiap data pada MATLAB menggunakan dasar matriks. Kode Program dikembangkan

b. Program bantu Matpower 7.1

Penelitian ini menggunakan program bantu Matpower7.1 yang berbasis pada MatLab untuk melakukan simulasi operasi sistem tenaga, yaitu simulasi operasi sistem tenaga terutama simulasi *power flow / load flow calculation*.

Matpower 7.1 memiliki fitur untuk menjalankan perhitungan *power flow* dengan *default* algoritmanya adalah metode *Newton-Raphson*. Parameter-parameter yang dapat diamati dari hasil dari simulasi *power flow* ini adalah:

1. Tegangan *busbar* (magnitude dan sudutnya)
2. Daya pada *busbar* (daya aktif dan reaktif)
3. Aliran daya pada setiap saluran termasuk losses pada setiap saluran transmisi

## KODE PROGRAM

### I. Inisialisasi Populasi

Inisialisasi populasi merupakan langkah awal dalam Algoritma Genetik (AG). Populasi dalam penelitian ini dilambangkan dengan bilangan antara 0 dan 9 yang disebut sebagai pengkodean *decimal*, yang tersusun atas kolom dan baris sehingga membentuk suatu matriks berisi bilangan antara 0 dan 9. Pada satu deret baris matriks tersusun atas beberapa kolom. Satu deret baris matriks ini pada AG dikenal dengan istilah kromosom sedangkan jumlah kolom tersebut dikenal dengan istilah jumlah gen.

```
function Populasi = PopulasiJawaBalibr(UkPop,Nbranch)

nAwal=[2 2 1 1 1 2 2 1 2 2 1 1 2 2 2 2 2 2 2 2 1 2 1 1 2 1 2 2 2 2];
larangan=find(nAwal==2);
%membangkitkan populasi
kromosom=ones(1,Nbranch);
Populasi=ones(UkPop,Nbranch);

%cek data n
cek=ones(1,length(kromosom));
%g=input('masukkan nilai n =2 ');
g=[1 2 6 7 9 10 13 14 15 16 17 18 19 20 22 25 27 28 29 30];
for i=1:length(g)
    cek(g)=2;
end

for k=1:UkPop;
for i=1:length(kromosom);
    kromosom(i)=1+round(3*rand);
    while ((kromosom(i)==1) & (cek(i)==2))
        kromosom(i)=1+round(3*rand);
    end
end
end

Populasi(k,:)=kromosom;
end
```

### II. Evaluasi Individu

Suatu individu atau kromosom dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran performasinya. Fungsi yang digunakan untuk mengukur nilai kecocokan atau derajat optimalitas suatu kromosom disebut dengan *fitness function*. Nilai yang dihasilkan dari fungsi tersebut menandakan seberapa optimal solusi yang diperoleh. Berikut program untuk fungsi fitness.

```
function [fitness,Ct] = EvaluasiIndividuJawaBali2(Populasi,Nbranch)

%Parameter Object Solusi
```



```

Nbranch=30;
Nbus=25;
NYE=20;
Cmwh=36.1;
define_constants;
gmax=[4025;3251;1008;700;2640;2239;764.5;4714];
nAwal=[2 2 1 1 1 2 2 1 2 2 1 1 2 2 2 2 2 2 2 2 1 2 1 1 2 1 2 2 2 2];
selisih=abs(Populasi-nAwal);

mpc=loadcase(caseJawaBali500kV);

for i=1:Nbranch
    if (selisih(i)>0)
        A= repmat(mpc.branch(i,:), selisih(i), 1);
        mpc.branch=[mpc.branch;A];
    else
        end
end

end
runpf(mpc);

%Menentukan cost investasi
cost=zeros(1,Nbranch);
jarak=[62.759 12.48 130.81 50.84 16.84 30.143 5.923 37.9 15.57 48.158 48
53 279.5 46.757 81.9 25.166 39 119.3 134.8 230 77.1 304.5 254.2 73.5
23.9 46.5 74.91 204.5 204.5 87.863];

for i=1:Nbranch
    if (selisih(i)==1) & (RATE_A==1990)
        cost(i)=0.52*jarak(i);
    elseif (selisih(i)==2) & (RATE_A==1990)
        cost(i)=0.65*jarak(i);
    elseif (selisih(i)==3) & (RATE_A==1990)
        cost(i)=1.17*jarak(i);
    elseif (selisih(i)==1) & (RATE_A==2200)
        cost(i)=0.60*jarak(i);
    elseif (selisih(i)==2) & (RATE_A==2200)
        cost(i)=0.75*jarak(i);
    elseif (selisih(i)==3) & (RATE_A==2200)
        cost(i)=1.35*jarak(i);
    else
        cost(i)=0;
    end
end

CostInvest=sum(cost)*1000000;

% Menentukan Cost loss
results=runpf(mpc);
Ploss = results.branch(:, PF) + results.branch(:, PT);
TotPloss=sum(Ploss);
Closs=8760*NYE*Cmwh*TotPloss; % NYE=20, Cmwh=36.1

%menentukan fungsi obyektif
Ct=CostInvest+Closs;

% menentukan constraints tegangan

```

```

Vbus=results.bus(:,VM);
deltaV=zeros(Nbus,1);
for i=1:Nbus
if Vbus(i)>1.05
    deltaV(i)=(Vbus(i)-1.05)*10^9;
elseif Vbus(i)<0.95
    deltaV(i)=(0.95-Vbus(i))*10^9;
elseif Vbus(i)>=0.95 & Vbus(i)<=1.05
    deltaV(i)=0;
end
end
TotalV=sum(deltaV);

%menentukan constraints Line Loading
Sf=sqrt(results.branch(:, PF).^2+results.branch(:, QF).^2);
St=sqrt(results.branch(:, PT).^2+results.branch(:, QT).^2);
Smax=mpc.branch(:,RATE_A);
S=(Sf+St)./Smax.*100;
TotBranch=sum(Populasi);
deltaS=zeros(TotBranch,1);
for i=1:TotBranch
if S(i)>50
    deltaS(i)=S(i)-50;
    else deltaS(i)=0;
end
end
TotalS=sum(deltaS)*10^7;

%Mengecek constraint Generation
genbus=results.gen(:,PG);
deltaGen=zeros(3,1);
for k=1:3
if genbus(k)>gmax(k)
    deltaGen(k)=genbus(k)-gmax(k);
    else deltaGen(k)=0;
end
end
TotalGen=sum(deltaGen)*10^6;

penalty_factor=TotalV+TotalS+TotalGen;
fitness=(1/(Ct+penalty_factor))*10^9;

```

### III. Linear Fitness Ranking

Linear Fitness Ranging merupakan proses pengurutan ranking dari hasil evaluasi individu. Berikut program untuk Linear Fitness Ranging.

```

function LFR = 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

```

#### IV. Roulette Wheel

Bagian ini digunakan untuk menentukan calon-calon anggota populasi untuk generasi berikutnya. Calon-calon anggota dipilih dengan cara memutar roda rolet (*roulette wheel*) sebanyak jumlah kromosom yang akan ditempatkan pada generasi berikutnya. Apabila pada suatu putaran roda rolet berhenti pada satu sektor, maka kromosom yang menempati sektor tersebut dipilih sebagai calon anggota populasi berikutnya.

```
function Pindex = RouletteWheel(UkPop,LinearFitness);

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
```

#### V. Pindah Silang

Operator pindah silang (*crossover*) merupakan salah satu operator penting dalam algoritma genetik. Operator ini bertujuan untuk menambah keanekaragaman kromosom dalam populasi dengan menyilangkan kromosom yang diperoleh dari operasi reproduksi sebelumnya.

```
function Anak = PindahSilang(Bapak,Ibu,Nbranch);

% Membangkitkan satu titik potong (TP bernilai antara 1 sampai JumGen-1)
TP = 1 + fix(rand*(Nbranch-1));

% Anak 1 berisi bagian depan Bapak dan bagian belakang Ibu
Anak(1,:) = [Bapak(1:TP) Ibu(TP+1:Nbranch)];
Anak(2,:) = [Ibu(1:TP) Bapak(TP+1:Nbranch)];
```

#### VI. Mutasi

Tujuan mutasi adalah untuk mengatasi gangguan terhadap parameter dan menjamin bahwa seluruh ruang lingkup dalam penelitian dapat dicapai. Umumnya, jika probabilitas mutasi ( $P_m$ ) lebih besar, rata-rata konvergensi lebih cepat tetapi menghasilkan suatu hasil keadaan kesalahan yang sama lebih besar.

```
function MutKrom = Mutasi(Kromosom, JumGen, Pmutasi);
```

```
MutKrom = Kromosom;  
for ii=1:JumGen,  
    if (rand < Pmutasi),  
        if Kromosom(ii)==0,  
            MutKrom(ii) = fix(rand*4);  
        else  
            MutKrom(ii) = Kromosom(ii);  
        end  
    end  
end
```

## VII. Main Program

Berikut adalah program utama Algoritma Genetika untuk mengoptimalkan perencanaan perluasan jaringan transmisi daya

```
clear;  
clc;  
close all;  
  
%Parameter AG  
UkPop=100;  
Psilang=0.8;  
Pmutasi=0.005;  
MaxG=50;  
BilKecil    = 0.07;           % Digunakan untuk menghindari pembagian dengan  
0  
Fthreshold  = 1/BilKecil;     % Threshold untuk nilai Fitness  
Bgraf       = Fthreshold;    % Untuk menangani tampilan grafis  
  
%Parameter Object Solusi  
Nbranch=30;  
Nbus=25;  
NYE=20;  
Cmwh=36.1;  
define_constants;  
gmax=[4025;3251;1008;700;2640;2239;764.5;4714];  
  
%inisialisasi gambar  
hfig=figure;  
hold on  
title('Optimasi TEP Jawa Bali 500 kV Berbasis AG dengan Mempertimbangkan  
Rugi-rugi daya')  
set(hfig, 'position', [50,50,600,400]);  
set(hfig, 'doublebuffer', 'on');  
axis([1 MaxG 0 Fthreshold]);  
hbestplot=plot(1:MaxG, zeros(1,MaxG));  
htext1=text(0.6*MaxG,0.15*10,sprintf('Fitness Terbaik:%7.4f',0.0));  
htext2=text(0.6*MaxG,0.10*10,sprintf('Obj. func. Terbaik (Ct)',0.0));
```



```

xlabel('Generasi');
ylabel('Fitness Terbaik');
hold off
drawnow
% Inisialisasi populasi
Populasi = PopulasiJawaBalibr(UkPop,Nbranch);

% Loop evolusi
for generasi=1:MaxG,
    [Fitness(1),Ct(1)] = EvaluasiIndividuJawaBali2(Populasi(1,:),Nbranch);
    MaxF = Fitness(1);
    MinF = Fitness(1);
    MaxCt=Ct(1);
    MinCt=Ct(1);
    IndeksIndividuTerbaik = 1;
    for ii=2:UkPop,
        Kromosom = Populasi(ii,:);
        [Fitness(ii),Ct(ii)] = EvaluasiIndividuJawaBali2(Kromosom,Nbranch);
        if (Fitness(ii) >= MaxF),
            MaxF = Fitness(ii);
            MaxCt = Ct(ii);
            IndeksIndividuTerbaik = ii;
            BestX = Kromosom;
        end
        if (Fitness(ii) < MinF),
            MinF = Fitness(ii);
            MinCt= Ct(ii);
        end
    end
end

%buat grafik
plotvector=get(hbestplot,'ydata');
plotvector(generasi)=MaxF;
set(hbestplot,'ydata',plotvector);
set(htext1,'string',sprintf('Fitness Terbaik:%7.4f',MaxF));
set(htext2,'string',sprintf('Obj. func. Terbaik (Ct):%7.4f',MaxCt));
drawnow

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);

% Roulette-wheel selection dan pindah silang
for jj=IterasiMulai:2:UkPop,

```

```

IP1 = RouletteWheel(UkPop,LinearFitness);
IP2 = RouletteWheel(UkPop,LinearFitness);
if (rand < Psilang),
    Anak = PindahSilang(Populasi(IP1,:),Populasi(IP2,:),Nbranch);
    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
for kk=IterasiMulai:UkPop,
    TempPopulasi(kk,:) = Mutasi(TempPopulasi(kk,:),Nbranch,Pmutasi);
end

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

End

```

# MANUAL PENGGUNAAN PROGRAM

## 1. Input Data Sistem

Data sistem diinputkan pada program bantu Matpower. Data sistem yang diperlukan adalah data pembangkit, data beban, dan data saluran transmisi Jawa-Bali 500 kV. Data dibuat dalam bentuk matriks dan disesuaikan dengan format Matpower kemudian disimpan dalam bentuk m-file. Hasil dari perhitungan *power flow* digunakan untuk menghitung fungsi obyektif dan evaluasi *fitness*.

```
function mpc = caseJawaBali500kVPDP
% PENELITIAN IKRIMA ALFI
% Data dari PLN P3B JB

%% MATPOWER Case Format : Version 2
mpc.version = '2';

%%----- Power Flow Data -----%%
%% system MVA base
mpc.baseMVA = 10000;

%% bus data
% bus_i type Pd Qd Gs Bs area Vm Va baseKV zone Vmax
Vmin
mpc.bus = [
    1 3 422 206 0 0 1 1.02 -0.00 500 1 1.05
0.95; %Suralaya
    2 1 616 464 0 0 1 1.01 -3.49 500 1 1.05
0.95; %Cilegon Baru
    3 1 677 158 0 0 1 0.98 -11.43 500 1 1.05
0.95; %Cawang
    4 1 1311 -29 0 0 1 0.99 -7.62 500 1 1.05
0.95; %Balaraja
    5 1 1097 263 0 0 1 0.99 -9.99 500 1 1.05
0.95; %Kembangan
    6 1 2349 555 0 0 1 0.98 -11.28 500 1 1.05
0.95; %Bekasi
    7 1 1599 277 0 0 1 0.99 -9.3 500 1 1.05
0.95; %Gandul
    8 1 1294 693 0 0 1 0.99 -9.59 500 1 1.05
0.95; %Cibinong
    9 1 1347 429 0 0 1 0.99 -9.38 500 1 1.05
0.95; %Depok
    10 2 0 0 0 0 1 0.99 -9.23 500 1 1.05
0.95; %Muara Tawar
    11 1 460 174 0 0 2 1.03 -6.93 500 1 1.05
0.95; %Tasikmalaya Baru
    12 1 1446 981 0 0 2 0.99 -10.02 500 1 1.05
0.95; %Cibatu
    13 2 1231 0 0 0 2 0.99 -9.48 500 1 1.05
0.95; %Cirata
    14 2 0 0 0 0 2 0.99 -9.09 500 1 1.05
0.95; %Saguling
```

```

15 1 733 423 0 0 2 1.00 0.00 500 1 1.05
0.95; %Bandung Selatan
16 1 1540 895 0 0 2 1.00 -7.11 500 1 1.05
0.95; %Mandirancan
17 2 500 23 0 0 3 1.00 5.83 500 1 1.05
0.95; %Tanjung Jati
18 1 876 983 0 0 3 1.01 0.07 500 1 1.05
0.95; %Ungaran
19 1 1278 481 0 0 3 1.01 -2.17 500 1 1.05
0.95; %Pedan
20 1 635 147 0 0 4 1.02 1.81 500 1 1.05
0.95; %Ngimbang
21 2 366 134 0 0 4 1.03 6.03 500 1 1.05
0.95; %Gresik Baru
22 1 1889 1076 0 0 4 1.03 5.47 500 1 1.05
0.95; %Surabaya Barat
23 2 1072 401 0 0 4 1.03 7.18 500 1 1.05
0.95; %Grati
24 1 1317 395 0 0 4 1.01 1.62 500 1 1.05
0.95; %Kediri
25 2 1366 307 0 0 4 1.03 9.05 500 1 1.05
0.95; %Paiton
26 1 445 100 0 0 4 1.01 1 500 1 1.05
0.95 %Kapal
];

```

```
%% generator data
```

```

% bus Pg Qg Qmax Qmin Vg mBase status
Pmax Pmin Pc1 Pc2 Qc1min Qc1max Qc2min Qc2max ramp_agc ramp_10
ramp_30 ramp_q apf
mpc.gen = [
1 5747 2635 3000 -1478.6 1.02 10000 1
6000 1610 0 0 0 0 0 0 0 0; %PLTU
Suralaya
10 3751 1809 2500 -1859.14 0.99 10000 1
4500 1300 0 0 0 0 0 0 0 0; %PLTG Muara
Tawar
13 1248 439 1200 -663.744 0.99 10000 1
1300 280 0 0 0 0 0 0 0 0; %PLTA
Cirata
14 1391 263 525 -460.936 0.99 10000 1
1500 403 0 0 0 0 0 0 0 0; %PLTA
Saguling
17 4141 122 1636.124 -969.812 1.00 10000 1
5000 1056 0 0 0 0 0 0 0 0; %PLTU
Tanjung Jati
21 2881 601 1679.292 -1474.37 1.03 10000 1
3500 895 0 0 0 0 0 0 0 0; %PLTU
Gresik
23 1000 122 573.372 -341.301 1.03 10000 1
1200 305 0 0 0 0 0 0 0 0; %PLTG Grati
25 6000 1284 3028.304 -1750.78 1.03 10000 1
6500 1886 0 0 0 0 0 0 0 0; %PLTU
Paiton
];

```

```
%% branch data
```

```

% fbus tbus r x b rateA rateB rateC ratio angle
status angmin angmax
mpc.branch = [
1 1 4 0.07355355 0.70666634 0 1990 0 0 0 0
1 -360 360; %Suralaya-New Balaraja S1
1 1 2 0.01252992 0.14017536 0 2200 0 0 0 0
1 -360 360; %Suralaya-Cilegon S1
1 2 8 0.13133324 1.46925792 0 2200 0 0 0 0
1 -360 360; %Cilegon-Cibinong
1 4 7 0.05958448 0.5724584 0 1990 0 0 0 0
1 -360 360; %New Balaraja-Gandul
1 6 3 0.01973648 0.1896184 0 1990 0 0 0 0
1 -360 360; %Bekasi-Cawang
1 5 7 0.03026357 0.33856618 0 2200 0 0 0 0
1 -360 360; %Gandul-Kembangan S1
1 7 9 0.00694176 0.06669298 0 1990 0 0 0 0
1 -360 360; %Gandul-Depok S1
1 8 6 0.0444188 0.426754 0 1990 0 0 0 0
1 -360 360; %Cibinong-Bekasi
1 9 8 0.01824804 0.1753182 0 1990 0 0 0 0
1 -360 360; %Depok-Cibinong S1
1 10 12 0.05644118 0.54225908 0 1990 0 0 0 0
1 -360 360; %Muara Tawar-Cibatu S1
1 10 3 0.056256 0.54048 0 1990 0 0 0 0
1 -360 360; %Muara Tawar-Cawang
1 10 8 0.062116 0.59678 0 1990 0 0 0 0
1 -360 360; %Muara Tawar-Cibinong
1 11 9 0.280618 3.139344 0.0283037811 2200 0 0 0 0
1 -360 360; %Tasikmalaya-Depok S1
1 13 12 0.0547992 0.52648382 0 1990 0 0 0 0
1 -360 360; %Cirata-Cibatu S1
1 14 8 0.0822276 0.9199008 0.0082936661 2200 0 0 0 0
1 -360 360; %Saguling-Cibinong S1
1 14 13 0.02949455 0.28336916 0 1990 0 0 0 0
1 -360 360; %Saguling-Cirata S1
1 14 15 0.039156 0.438048 0 2200 0 0 0 0
1 -360 360; %Saguling-Bandung Selatan S1
1 16 15 0.1398196 1.343318 0.0120302759 1990 0 0 0 0
1 -360 360; %Mandirancan-Bandung Selatan S1
1 17 18 0.1353392 1.5140736 0 2200 0 0 0 0
1 -360 360; %Tanjung Jati-Ungaran S1
1 18 16 0.26956 2.5898 0.0231933232 1990 0 0 0 0
1 -360 360; %Ungaran-Mandirancan S1
1 18 19 0.0903612 2.5898 0 1990 0 0 0 0
1 -360 360; %Ungaran-Pedan
1 19 11 0.305718 3.420144 0.0308354252 2200 0 0 0 0
1 -360 360; %Pedan-Tasikmalaya S1
1 22 18 0.2979224 2.862292 0.0256336642 1990 0 0 0 0
1 -360 360; %Surabaya Barat-Ungaran
1 20 18 0.086142 0.82761 0.0074117794 1990 0 0 0 0
1 -360 360; %Ngimbang-Ungaran
1 21 22 0.0280108 0.269114 0 1990 0 0 0 0
1 -360 360; %Gresik Baru-Surabaya Barat S1
1 22 20 0.054498 0.52359 0.0046890849 1990 0 0 0 0
1 -360 360; %Surabaya Barat-Ngimbang
1 23 22 0.07972764 0.89193312 0 2200 0 0 0 0
1 -360 360; %Grati-Surabaya Barat S1
1 24 19 0.205318 2.296944 0.0207088488 2200 0 0 0 0
1 -360 360; %Kediri-Pedan S1
1 25 24 0.205318 2.296944 0.0207088488 2200 0 0 0 0
1 -360 360; %Paiton-Kediri S1

```

```

25 23 0.08821445 0.98687722 0.0088975138 2200 0 0 0 0
1 -360 360; %Paiton-Grati S1
25 26 0.49224 4.71744 0 1990 0 0 0 0
0 -360 360; %Paiton-Kapal S1
1 4 0.07355355 0.70666634 0 1990 0 0 0 0
1 -360 360; %Suralaya-New Balaraja S2
1 2 0.01252992 0.14017536 0 2200 0 0 0 0
1 -360 360; %Suralaya-Cilegon S2
7 5 0.03026357 0.33856618 0 2200 0 0 0 0
1 -360 360; %Gandul-Kembangan S2
7 9 0.00694176 0.06669298 0 1990 0 0 0 0
1 -360 360; %Gandul-Depok S2
9 8 0.01824804 0.1753182 0 1990 0 0 0 0
1 -360 360; %Depok-Cibinong S2
10 12 0.05644118 0.54225908 0 1990 0 0 0 0
1 -360 360; %Muara Tawar-Cibatu S2
11 9 0.280618 3.139344 0.0283037811 2200 0 0 0 0
1 -360 360; %Tasikmalaya-Depok S2
13 12 0.0547992 0.52648382 0 1990 0 0 0 0
1 -360 360; %Cirata-Cibatu S2
14 8 0.0822276 0.9199008 0.0082936661 2200 0 0 0 0
1 -360 360; %Saguling-Cibinong S2
14 13 0.02949455 0.28336916 0 1990 0 0 0 0
1 -360 360; %Saguling-Cirata S2
14 15 0.039156 0.438048 0 2200 0 0 0 0
1 -360 360; %Saguling-Bandung Selatan S2
16 15 0.1398196 1.343318 0.0120302759 1990 0 0 0 0
1 -360 360; %Mandirancan-Bandung Selatan S2
17 18 0.1353392 1.5140736 0 2200 0 0 0 0
1 -360 360; %Tanjung Jati-Ungaran S2
18 16 0.26956 2.5898 0.0231933232 1990 0 0 0 0
1 -360 360; %Ungaran-Mandirancan S2
19 11 0.305718 3.420144 0.0308354252 2200 0 0 0 0
1 -360 360; %Pedan-Tasikmalaya S2
21 22 0.0280108 0.269114 0 1990 0 0 0 0
1 -360 360; %Gresik Baru-Surabaya Barat S2
23 22 0.07972764 0.89193312 0 2200 0 0 0 0
1 -360 360; %Grati-Surabaya Barat S2
24 19 0.205318 2.296944 0.0207088488 2200 0 0 0 0
1 -360 360; %Kediri-Pedan S2
25 24 0.205318 2.296944 0.0207088488 2200 0 0 0 0
1 -360 360; %Paiton-Kediri S2
25 23 0.08821445 0.98687722 0.0088975138 2200 0 0 0 0
1 -360 360; %Paiton-Grati S2
];

```

2. Program Optimasi Perencanaan Perluasan Jaringan Transmisi Daya Listrik menggunakan Algoritma Genetika ditulis menggunakan Matlab
3. Pada menu matlab, set-path ke file tempat data m-file disimpan
4. Run main program