

LAMPIRAN 1

PROGRAM ALGORITMA GENETIKA

```

%=====
% Program Utama Algoritma Genetika
%=====
clc                                     % Me-refresh command window
clear all                               % Menghapus semua semua variabel yang sedang aktif

Nvar   = 3;                             % Jumlah variabel pada fungsi yang dioptimasi
Nbit   = 17;                             % Jumlah bit yang mengkodekan satu variabel
JumGen = Nbit*Nvar;                       % Jumlah gen dalam kromosom
Rb     = 0;                               % Batas bawah interval
Ra     = 100;                             % Batas atas interval

UkPop  = 50;                             % Jumlah kromosom dalam populasi
Psilang = 0.3;                           % Probabilitas pindah silang
Pmutasi = 0.03;                           % Probabilitas mutasi
MaxG    = 1500;                           % Jumlah generasi

BilKecil = 10^-1;                         % Digunakan untuk menghindari pembagian dengan 0
Fthreshold = 5;                           % Threshold untuk nilai Fitness
Bgraf    = Fthreshold;                    % Untuk menangani tampilan grafis

% Inisialisasi grafis 2D
hfig = figure;
hold on
title('Optimasi fungsi menggunakan Algoritma Genetika')
set(hfig, 'position', [50,50,600,400]);
set(hfig, 'DoubleBuffer', 'on');
axis([1 MaxG 0 Bgraf]);
hbestplot = plot(1:MaxG,zeros(1,MaxG));
htext1 = text(0.6*MaxG,0.40*Bgraf,sprintf('Fitness terbaik: %7.4f', 0.0));
htext2 = text(0.6*MaxG,0.35*Bgraf,sprintf('Nilai Kp: %5.4f', 0.0));
htext3 = text(0.6*MaxG,0.30*Bgraf,sprintf('Nilai Ki: %5.4f', 0.0));
htext4 = text(0.6*MaxG,0.25*Bgraf,sprintf('Nilai Kd: %5.4f', 0.0));
htext5 = text(0.6*MaxG,0.20*Bgraf,sprintf('Nilai minimum: %5.4f', 0.0));
htext6 = text(0.6*MaxG,0.15*Bgraf,sprintf('Ukuran populasi: %3.0f', 0.0));
htext7 = text(0.6*MaxG,0.10*Bgraf,sprintf('Probabilitas mutasi: %5.4f', 0.0));
htext8 = text(0.6*MaxG,0.05*Bgraf,sprintf('Probabilitas crossover: %5.4f', 0.0));
xlabel('Generasi');
ylabel('Fitness terbaik');
hold off
drawnow;

% Inisialisasi populasi
Populasi = InisialisasiPopulasi(UkPop,JumGen);

% Loop evolusi
for generasi=1:MaxG,
    for ii=1,

```

```

x = DekodekanKromosom(Populasi(1,:),Nvar,Nbit,Ra,Rb);
Fitness(1) = EvaluasiIndividu(x,BilKecil);
MaxF = Fitness(1);
MinF = Fitness(1);
IndeksIndividuTerbaik = 1;
end

for ii=2,
x = DekodekanKromosom(Populasi(2,:),Nvar,Nbit,Ra,Rb);
Fitness(2) = EvaluasiIndividu(x,BilKecil);
MaxF = Fitness(2);
MinF = Fitness(2);
IndeksIndividuTerbaik = 2;
end

for ii=3:UkPop,
Kromosom = Populasi(ii,:);
x = DekodekanKromosom(Kromosom,Nvar,Nbit,Ra,Rb);
Fitness(ii) = EvaluasiIndividu(x,BilKecil);
if (Fitness(ii) > MaxF),
    MaxF = Fitness(ii);
    IndeksIndividuTerbaik = ii;
    BestX = x;
end
if (Fitness(ii) < MinF),
    MinF = Fitness(ii);
end
end

% Penanganan grafis 2D
plotvector = get(hbestplot,'YData');
plotvector(generasi) = MaxF;
set(hbestplot,'YData',plotvector);
set(htext1,'String',sprintf('Fitness terbaik: %7.4f', MaxF));
set(htext2,'String',sprintf('Nilai Kp: %5.4f', BestX(1)));
set(htext3,'String',sprintf('Nilai Ki: %5.4f', BestX(2)));
set(htext4,'String',sprintf('Nilai Kd: %5.4f', BestX(3)));
set(htext5,'String',sprintf('Nilai minimum: %5.4f', (1/MaxF)-BilKecil));
set(htext6,'String',sprintf('Ukuran populasi: %3.0f', UkPop));
set(htext7,'String',sprintf('Probabilitas mutasi: %5.4f', Pmutasi));
set(htext8,'String',sprintf('Probabilitas crossover: %5.4f', Psilang));
drawnow

if MaxF >= Fthreshold,
    break;
end

TempPopulasi = Populasi;

% Elitisme:
% - Buat dua kopi kromosom terbaik jika ukuran populasi genap

```

```

IterasiMulai = 3;
TempPopulasi(1,:) = Populasi(IndeksIndividuTerbaik,:);
TempPopulasi(2,:) = Populasi(IndeksIndividuTerbaik,:);

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,:),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
for kk=IterasiMulai:UkPop,
    TempPopulasi(kk,:) = Mutasi(TempPopulasi(kk,:),JumGen,Pmutasi);
end

```

```

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

```

```

%=====
% Mendekodekan kromosom yang berisi bilangan biner menjadi individu x yang
% bernilai real dalam interval yang ditentukan [Ra,Rb]
%=====

```

```

function x = DekodekanKromosom(Kromosom,Nvar,Nbit,Ra,Rb)

```

```

for ii=1:Nvar,
    x(ii) = 0;
    for jj=1:Nbit,
        x(ii) = x(ii) + Kromosom((ii-1)*Nbit+jj)*2^(-jj);
    end
    x(ii) = Rb + (Ra-Rb)*x(ii);
end

```

```

%=====
% Mengevaluasi individu sehingga didapatkan nilai fitness-nya
%=====

```

```

function fitness = EvaluasiIndividu(x,BilKecil)
R=5;

```

```
T=0.1;
UkPop=50;
```

```
for ii=1,
y(ii)=0;
e(ii)=R-y(ii);
m(ii)=(x(1)+T*x(2)+x(3)/T)*e(ii);
itae=T*abs(e(ii));
fitness = 10 / (itae+BilKecil);
end
```

```
for ii=2,
y(ii)=0;
e(ii)=R-y(ii);
m(ii)=m(ii-1)+(x(1)+T*x(2)+x(3)/T)*e(ii)-(x(1)+2*x(3)/T)*e(1);
itae = itae + ii*T*abs(e(ii));
fitness = 10 / (itae+BilKecil);
end
```

```
for ii=3:UkPop,
y(ii)=0.8635*y(ii-1) - 0.06015*y(ii-2) + 0.0345*m(ii-1) + 0.0345*m(ii-2);
e(ii)=R-y(ii);
m(ii)=m(ii-1)+(x(1)+ T*x(2)+x(3)/T)*e(ii)-(x(1)+2*x(3)/T)*e(ii-1)+(x(3)*e(ii-2)/T);
itae = itae + ii*T*abs(e(ii));
fitness = 10 / (itae+BilKecil);
end
```

```
%=====
% Membangkitkan sejumlah UkPop kromosom, masing-masing kromosom berisi bilangan
% biner (0 dan 1) sejumlah JumGen
%=====
```

```
function Populasi = InisialisasiPopulasi(UkPop,JumGen)
```

```
Populasi = fix(2*rand(UkPop,JumGen));
```

```
%=====
% Men-skala-kan nilai fitness ke dalam ranking sehingga diperoleh
% nilai-nilai fitness baru yang berada dalam rentang [MaxF,MinF]
%=====
```

```
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

```
%=====
```

```
% Mutasi gen dengan probabilitas sebesar Pmutasi
```

```
% Gen-gen yang terpilih diubah nilainya: 0 menjadi 1, dan 1 menjadi 0
```

```
%=====
```

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

```
MutKrom = Kromosom;
```

```
for ii=1:JumGen,
```

```
    if (rand < Pmutasi),
```

```
        if Kromosom(ii)==0,
```

```
            MutKrom(ii) = 1;
```

```
        else
```

```
            MutKrom(ii) = 0;
```

```
        end
```

```
    end
```

```
end
```

```
%=====
```

```
% Memindah-silangkan bagian kromosom Bapak dan Ibu yang dipotong
```

```
% secara random, sehingga dihasilkan dua buah kromosom Anak
```

```
%=====
```

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

```
% 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)];
```

```
%=====
```

```
% Memilih orang tua menggunakan LinearFitness, yaitu nilai fitness hasil
```

```
% pen-skala-an. Pemilihan dilakukan secara proporsional sesuai dengan
```

```
% nilai fitness-nya.
```

```
%=====
```

```
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
```