LAMPIRAN A

KODE PROGRAM

KODE PROGRAM
UNTUK SIMULASI POLA RADIASI 31 ELEMEN SUSUNAN ANTENA DENGAN PENGKODEAN JARAK ANTAR ELEMEN 3 BIT

```
clc;
clear all;
pjg_gel=1;
B=2*pi/pjg_gel;
teta= 0:1:180;
phi=0:1:180;
bnyk_phi=length(phi);
bnyk_teta=length(teta);
jum_pop=15;
jum_bit=3;
jum_elemen=30;
jum_gen=jum_bit*jum_elemen;
jum_generasi=50;
ra=1;
rb=0;
psilang=0.8;
pmut=0.01;

for i=1:jum_pop
    for j=1:jum_gen
        a=rand;
        if(a<0.5)
            populasi(i,j)=0;
        else
            populasi(i,j)=1;
        end;
    end;
end;

xx=0;
ss=0;

gen=1;
simpan_kromosom=zeros(jum_generasi*jum_pop,jum_gen+jum_
elemen+3);

while gen<=jum_generasi
    close all;
        k=1;
    while k<=jum_pop
        kromosom=populasi(k,:);
        for l=1:jum_elemen
            xx=l*jum_bit;
            ss=xx-(jum_bit-1);
            jarak(l)=(kromosom(ss)*2^-
1+kromosom(ss+1)*2^-2+kromosom(ss+2)*2^-3)+0.125 ;
        end;
        jarak_ahkir=cumsum(jarak);
        for m=1:jum_elemen

jarak_kali_cos_teta(m,:)=B*jarak_ahkir(m).*cos(teta*pi/
180);
        end;
        for n=1:bnyk_teta

jum_jarak_cos(:,n)=cumsum(jarak_kali_cos_teta(:,n));
        end;
        hasil_cos=cos(jum_jarak_cos*pi/180);
        for o=1:bnyk_teta
            jum_hasil_cos(o)=sum(hasil_cos(:,o));
        end;

        for p=1:bnyk_phi
            for q=1:bnyk_teta

sin_hasil_cos(p,q)=2*sin(phi(p)*pi/180)*jum_hasil_cos(q
);
            end;
        end;
        for r=1:bnyk_teta
            jum_sin_hasil(r)=sum(sin_hasil_cos(:,r));
        end;
        log_jum_sin=10*log10(jum_sin_hasil);

        figure(k)

        plot(teta,log_jum_sin)
        xlabel('derajat')
        ylabel('Medan Jauh Relatif (dB)')
        grid on
        xlim([0 180])
        ylim([25 45])

        aa=log_jum_sin;
        ujung=length(aa);
        indeks_maks_1=find(aa==max(aa))
        for mm=indeks_maks_1:1:ujung-1
            mm;
            selisih=aa(mm)-aa(mm+1);
            if selisih <0
                indeks_min_1=mm
                break;
            end
        end
        mm;
        nn=mm;
        for nn=indeks_min_1:1:ujung-1
            nn;
            selisih=aa(nn)-aa(nn+1);
            if selisih >0
                indeks_maks_2=nn
                break;
            end
        end
        main_lobe=aa(indeks_maks_1)
        side_lobe=aa(indeks_maks_2)
        fitness=main_lobe-side_lobe
        temp(k,:)=[kromosom jarak main_lobe side_lobe
fitness];
        pause
        k=k+1;
    end;
    simpan_kromosom((gen-
1)*jum_pop+1:gen*jum_pop,:)=temp;
    sort_fitness=sortrows(temp,jum_gen+jum_elemen+3);
    LFR=flipud(sort_fitness);
    induk1=LFR(1,1:jum_gen);
    induk2=LFR(2,1:jum_gen);
    induk3=LFR(3,1:jum_gen);
    induk4=LFR(4,1:jum_gen);
    induk5=LFR(5,1:jum_gen);
    induk6=LFR(6,1:jum_gen);
    induk7=LFR(7,1:jum_gen);
    TP=1+fix(rand*(jum_gen-1));
    anak(1,:)=[induk1(1:TP) induk2(TP+1:jum_gen)];
    anak(2,:)=[induk2(1:TP) induk1(TP+1:jum_gen)];
    anak(3,:)=[induk3(1:TP) induk4(TP+1:jum_gen)];
    anak(4,:)=[induk4(1:TP) induk3(TP+1:jum_gen)];
    mut_induk1=induk1;
    for ii=1:jum_gen
        if (rand<pmut),
            if induk1(ii)==0,
                mut_induk1(ii)=1;
            else
                mut_induk1(ii)=0;
            end;
        end;
    end;
    mut_induk2=induk2;
    for ii=1:jum_gen
        if (rand<pmut),
            if induk2(ii)==0,
                mut_induk2(ii)=1;
            else
                mut_induk2(ii)=0;
            end;
        end;
    end;
    mut_induk3=induk3;
    for ii=1:jum_gen
        if (rand<pmut),
            if induk3(ii)==0,
                mut_induk3(ii)=1;
            else
                mut_induk3(ii)=0;
            end;
        end;
    end;
    mut_induk4=induk4;
    for ii=1:jum_gen
        if (rand<pmut),
            if induk4(ii)==0,
                mut_induk4(ii)=1;
            else
                mut_induk4(ii)=0;
            end;
        end;
    end;
    mut_induk5=induk5;
    for ii=1:jum_gen
        if (rand<pmut),
            if induk5(ii)==0,
                mut_induk5(ii)=1;
            else
                mut_induk5(ii)=0;
            end;
        end;
    end;
    mut_induk6=induk6;
    for ii=1:jum_gen
        if (rand<pmut),
            if induk6(ii)==0,
                mut_induk6(ii)=1;
            else
                mut_induk6(ii)=0;
            end;
        end;
    end;
```

```
    mut_induk7=induk4;
    for ii=1:jum_gen
        if (rand<pmut),
            if induk7(ii)==0,
                mut_induk7(ii)=1;
            else
                mut_induk7(ii)=0;
            end;
        end;
    end;
    temp_populasi(1,:)=induk1;
    temp_populasi(2,:)=induk2;
    temp_populasi(3,:)=induk3;
    temp_populasi(4,:)=induk4;
    temp_populasi(5,:)=anak(1,:);
    temp_populasi(6,:)=anak(2,:);
    temp_populasi(7,:)=anak(3,:);
    temp_populasi(8,:)=anak(4,:);
    temp_populasi(9,:)=mut_induk1;
    temp_populasi(10,:)=mut_induk2;
    temp_populasi(11,:)=mut_induk3;
    temp_populasi(12,:)=mut_induk4;
    temp_populasi(13,:)=mut_induk5;
    temp_populasi(14,:)=mut_induk6;
    temp_populasi(15,:)=mut_induk7;

    for ii=16:jum_pop
        jj=ii-11;
        temp_populasi(ii,:)=LFR(jj,1:jum_gen);
    end;

    populasi=temp_populasi;
    gen=gen+1;
end;

close all;
```

KODE PROGRAM
UNTUK SIMULASI POLA RADIASI 31 ELEMEN SUSUNAN ANTENA DENGAN PENGKODEAN JARAK ANTAR ELEMEN 4 BIT

```
clc;
clear all;
pjg_gel=1;
B=2*pi/pjg_gel;
teta= 0:1:180;
phi=0:1:180;
bnyk_phi=length(phi);
bnyk_teta=length(teta);
jum_pop=15;
jum_bit=4;
jum_elemen=30;
jum_gen=jum_bit*jum_elemen;
jum_generasi=50;
ra=1;
rb=0;
psilang=0.8;
pmut=0.01;

for i=1:jum_pop
    for j=1:jum_gen
        a=rand;
        if(a<0.5)
            populasi(i,j)=0;
        else
            populasi(i,j)=1;
        end;
    end;
end;

xx=0;
ss=0;

gen=1;
simpan_kromosom=zeros(jum_generasi*jum_pop,jum_gen+jum_
elemen+3);

while gen<=jum_generasi
    close all;
        k=1;
    while k<=jum_pop
        kromosom=populasi(k,:);
        for l=1:jum_elemen
            xx=l*jum_bit;
            ss=xx-(jum_bit-1);
```

```
      jarak(l)=(kromosom(ss)*2^-
1+kromosom(ss+1)*2^-2+kromosom(ss+2)*2^-
3)+kromosom(ss+3)*2^-4+0.0625 ;
    end;
    jarak_ahkir=cumsum(jarak);
    for m=1:jum_elemen

jarak_kali_cos_teta(m,:)=B*jarak_ahkir(m).*cos(teta*pi/
180);
    end;
    for n=1:bnyk_teta

jum_jarak_cos(:,n)=cumsum(jarak_kali_cos_teta(:,n));
    end;
    hasil_cos=cos(jum_jarak_cos*pi/180);
    for o=1:bnyk_teta
      jum_hasil_cos(o)=sum(hasil_cos(:,o));
    end;

    for p=1:bnyk_phi
      for q=1:bnyk_teta

sin_hasil_cos(p,q)=2*sin(phi(p)*pi/180)*jum_hasil_cos(q
);
      end;
    end;
    for r=1:bnyk_teta
      jum_sin_hasil(r)=sum(sin_hasil_cos(:,r));
    end;
    log_jum_sin=10*log10(jum_sin_hasil);

    figure(k)

    plot(teta,log_jum_sin)
    xlabel('derajat')
    ylabel('Medan Jauh Relatif (dB)')
    grid on
    xlim([0 180])
    ylim([25 45])

    aa=log_jum_sin;
    ujung=length(aa);
    indeks_maks_1=find(aa==max(aa))
    for mm=indeks_maks_1:1:ujung-1
      mm;
      selisih=aa(mm)-aa(mm+1);
      if selisih <0
        indeks_min_1=mm
```

```
                break;
            end
        end
        mm;
        nn=mm;
        for nn=indeks_min_1:1:ujung-1
            nn;
            selisih=aa(nn)-aa(nn+1);
            if selisih >0
                indeks_maks_2=nn
                break;
            end
        end
        main_lobe=aa(indeks_maks_1)
        side_lobe=aa(indeks_maks_2)
        fitness=main_lobe-side_lobe
        temp(k,:)=[kromosom jarak main_lobe side_lobe
fitness];
        pause
        k=k+1;
    end;
    simpan_kromosom((gen-
1)*jum_pop+1:gen*jum_pop,:)=temp;
    sort_fitness=sortrows(temp,jum_gen+jum_elemen+3);
    LFR=flipud(sort_fitness);
    induk1=LFR(1,1:jum_gen);
    induk2=LFR(2,1:jum_gen);
    induk3=LFR(3,1:jum_gen);
    induk4=LFR(4,1:jum_gen);
    induk5=LFR(5,1:jum_gen);
    induk6=LFR(6,1:jum_gen);
    induk7=LFR(7,1:jum_gen);
    TP=1+fix(rand*(jum_gen-1));
    anak(1,:)=[induk1(1:TP) induk2(TP+1:jum_gen)];
    anak(2,:)=[induk2(1:TP) induk1(TP+1:jum_gen)];
    anak(3,:)=[induk3(1:TP) induk4(TP+1:jum_gen)];
    anak(4,:)=[induk4(1:TP) induk3(TP+1:jum_gen)];
    mut_induk1=induk1;
    for ii=1:jum_gen
        if (rand<pmut),
            if induk1(ii)==0,
                mut_induk1(ii)=1;
            else
                mut_induk1(ii)=0;
            end;
        end;
    end;
    mut_induk2=induk2;
```

```
for ii=1:jum_gen
    if (rand<pmut),
        if induk2(ii)==0,
            mut_induk2(ii)=1;
        else
            mut_induk2(ii)=0;
        end;
    end;
end;
mut_induk3=induk3;
for ii=1:jum_gen
    if (rand<pmut),
        if induk3(ii)==0,
            mut_induk3(ii)=1;
        else
            mut_induk3(ii)=0;
        end;
    end;
end;
mut_induk4=induk4;
for ii=1:jum_gen
    if (rand<pmut),
        if induk4(ii)==0,
            mut_induk4(ii)=1;
        else
            mut_induk4(ii)=0;
        end;
    end;
end;
mut_induk5=induk5;
for ii=1:jum_gen
    if (rand<pmut),
        if induk5(ii)==0,
            mut_induk5(ii)=1;
        else
            mut_induk5(ii)=0;
        end;
    end;
end;
mut_induk6=induk6;
for ii=1:jum_gen
    if (rand<pmut),
        if induk6(ii)==0,
            mut_induk6(ii)=1;
        else
            mut_induk6(ii)=0;
        end;
    end;
    end;
    mut_induk7=induk4;
    for ii=1:jum_gen
        if (rand<pmut),
            if induk7(ii)==0,
                mut_induk7(ii)=1;
            else
                mut_induk7(ii)=0;
            end;
        end;
    end;
    temp_populasi(1,:)=induk1;
    temp_populasi(2,:)=induk2;
    temp_populasi(3,:)=induk3;
    temp_populasi(4,:)=induk4;
    temp_populasi(5,:)=anak(1,:);
    temp_populasi(6,:)=anak(2,:);
    temp_populasi(7,:)=anak(3,:);
    temp_populasi(8,:)=anak(4,:);
    temp_populasi(9,:)=mut_induk1;
    temp_populasi(10,:)=mut_induk2;
    temp_populasi(11,:)=mut_induk3;
    temp_populasi(12,:)=mut_induk4;
    temp_populasi(13,:)=mut_induk5;
    temp_populasi(14,:)=mut_induk6;
    temp_populasi(15,:)=mut_induk7;

    for ii=16:jum_pop
        jj=ii-11;
        temp_populasi(ii,:)=LFR(jj,1:jum_gen);
    end;

    populasi=temp_populasi;
    gen=gen+1;
end;

close all;
```

KODE PROGRAM
UNTUK SIMULASI POLA RADIASI 49 ELEMEN SUSUNAN ANTENA
DENGAN PENGKODEAN JARAK ANTAR ELEMEN 3 BIT

```
clc;
clear all;
pjg_gel=1;
B=2*pi/pjg_gel;
teta= 0:1:180;
phi=0:1:180;
bnyk_phi=length(phi);
bnyk_teta=length(teta);
jum_pop=15;
jum_bit=3;
jum_elemen=48;
jum_gen=jum_bit*jum_elemen;
jum_generasi=50;
ra=1;
rb=0;
psilang=0.8;
pmut=0.01;

for i=1:jum_pop
    for j=1:jum_gen
        a=rand;
        if(a<0.5)
            populasi(i,j)=0;
        else
            populasi(i,j)=1;
        end;
    end;
end;

xx=0;
ss=0;

gen=1;
simpan_kromosom=zeros(jum_generasi*jum_pop,jum_gen+jum_elemen+3);

while gen<=jum_generasi
    close all;
        k=1;
    while k<=jum_pop
        kromosom=populasi(k,:);
        for l=1:jum_elemen
            xx=l*jum_bit;
            ss=xx-(jum_bit-1);
            jarak(l)=(kromosom(ss)*2^-
1+kromosom(ss+1)*2^-2+kromosom(ss+2)*2^-3)+0.125 ;
        end;
        jarak_ahkir=cumsum(jarak);
        for m=1:jum_elemen

jarak_kali_cos_teta(m,:)=B*jarak_ahkir(m).*cos(teta*pi/
180);
        end;
        for n=1:bnyk_teta

jum_jarak_cos(:,n)=cumsum(jarak_kali_cos_teta(:,n));
        end;
        hasil_cos=cos(jum_jarak_cos*pi/180);
        for o=1:bnyk_teta
            jum_hasil_cos(o)=sum(hasil_cos(:,o));
        end;

        for p=1:bnyk_phi
            for q=1:bnyk_teta

sin_hasil_cos(p,q)=2*sin(phi(p)*pi/180)*jum_hasil_cos(q
);
            end;
        end;
        for r=1:bnyk_teta
            jum_sin_hasil(r)=sum(sin_hasil_cos(:,r));
        end;
        log_jum_sin=10*log10(jum_sin_hasil);

        figure(k)

        plot(teta,log_jum_sin)
        xlabel('derajat')
        ylabel('Medan Jauh Relatif (dB)')
        grid on
        xlim([0 180])
        ylim([25 45])

        aa=log_jum_sin;
        ujung=length(aa);
        indeks_maks_1=find(aa==max(aa))
        for mm=indeks_maks_1:1:ujung-1
            mm;
            selisih=aa(mm)-aa(mm+1);
            if selisih <0
                indeks_min_1=mm
                break;
            end
        end
        mm;
        nn=mm;
        for nn=indeks_min_1:1:ujung-1
            nn;
            selisih=aa(nn)-aa(nn+1);
            if selisih >0
                indeks_maks_2=nn
                break;
            end
        end
        main_lobe=aa(indeks_maks_1)
        side_lobe=aa(indeks_maks_2)
        fitness=main_lobe-side_lobe
        temp(k,:)=[kromosom jarak main_lobe side_lobe
fitness];
        pause
        k=k+1;
    end;
    simpan_kromosom((gen-
1)*jum_pop+1:gen*jum_pop,:)=temp;
    sort_fitness=sortrows(temp,jum_gen+jum_elemen+3);
    LFR=flipud(sort_fitness);
    induk1=LFR(1,1:jum_gen);
    induk2=LFR(2,1:jum_gen);
    induk3=LFR(3,1:jum_gen);
    induk4=LFR(4,1:jum_gen);
    induk5=LFR(5,1:jum_gen);
    induk6=LFR(6,1:jum_gen);
    induk7=LFR(7,1:jum_gen);
    TP=1+fix(rand*(jum_gen-1));
    anak(1,:)=[induk1(1:TP) induk2(TP+1:jum_gen)];
    anak(2,:)=[induk2(1:TP) induk1(TP+1:jum_gen)];
    anak(3,:)=[induk3(1:TP) induk4(TP+1:jum_gen)];
    anak(4,:)=[induk4(1:TP) induk3(TP+1:jum_gen)];
    mut_induk1=induk1;
    for ii=1:jum_gen
        if (rand<pmut),
            if induk1(ii)==0,
                mut_induk1(ii)=1;
            else
                mut_induk1(ii)=0;
            end;
        end;
    end;
    mut_induk2=induk2;
    for ii=1:jum_gen
        if (rand<pmut),
            if induk2(ii)==0,
                mut_induk2(ii)=1;
            else
                mut_induk2(ii)=0;
            end;
        end;
    end;
    mut_induk3=induk3;
    for ii=1:jum_gen
        if (rand<pmut),
            if induk3(ii)==0,
                mut_induk3(ii)=1;
            else
                mut_induk3(ii)=0;
            end;
        end;
    end;
    mut_induk4=induk4;
    for ii=1:jum_gen
        if (rand<pmut),
            if induk4(ii)==0,
                mut_induk4(ii)=1;
            else
                mut_induk4(ii)=0;
            end;
        end;
    end;
    mut_induk5=induk5;
    for ii=1:jum_gen
        if (rand<pmut),
            if induk5(ii)==0,
                mut_induk5(ii)=1;
            else
                mut_induk5(ii)=0;
            end;
        end;
    end;
    mut_induk6=induk6;
    for ii=1:jum_gen
        if (rand<pmut),
            if induk6(ii)==0,
                mut_induk6(ii)=1;
            else
                mut_induk6(ii)=0;
            end;
        end;
    end;
```

```
    mut_induk7=induk4;
    for ii=1:jum_gen
        if (rand<pmut),
            if induk7(ii)==0,
                mut_induk7(ii)=1;
            else
                mut_induk7(ii)=0;
            end;
        end;
    end;
    temp_populasi(1,:)=induk1;
    temp_populasi(2,:)=induk2;
    temp_populasi(3,:)=induk3;
    temp_populasi(4,:)=induk4;
    temp_populasi(5,:)=anak(1,:);
    temp_populasi(6,:)=anak(2,:);
    temp_populasi(7,:)=anak(3,:);
    temp_populasi(8,:)=anak(4,:);
    temp_populasi(9,:)=mut_induk1;
    temp_populasi(10,:)=mut_induk2;
    temp_populasi(11,:)=mut_induk3;
    temp_populasi(12,:)=mut_induk4;
    temp_populasi(13,:)=mut_induk5;
    temp_populasi(14,:)=mut_induk6;
    temp_populasi(15,:)=mut_induk7;

    for ii=16:jum_pop
        jj=ii-11;
        temp_populasi(ii,:)=LFR(jj,1:jum_gen);
    end;

    populasi=temp_populasi;
    gen=gen+1;
end;

close all;
```