LAMPIRAN

Kode Program Matlab

```
clear
clc
%
%  Program untuk menentukan natural static & rotating mode shapes
%  frequencies pada balok yang seragam/uniform menggunakan model UH-60
%
mode=3;
p=100;                                              % banyak elemen
diskritisasi;
alt=0;                                              % bisa 0, 5k, 10k
cgea=0.25; %input(5);
mn=1;                                               % input('Vfwd
minimum? ');
mx=2000;                                            % input('Vfwd
maximum? ');
%
R=322; offset=15; chord=20.76;                      % in
L=R-offset; e=offset/L;                             % in
O=27.02;                                            % kecepatan rotor
rad/s
mu=0.00164167;                                      % lbs^2/in^2
E=10000000;                                         % lb/in^2
EIxx=2.27859e7; EIyy=7.71261e8;                     % lbin^2
I=(EIxx+EIyy)/E;                                    % in^4
Ia=0.037006386;                                     % lbins^2/in
polar mass MOI trhadap e.a.
GJ=2.467909e7;                                      % lbin^2
%
a=-0.5;                                             % non-dim elastic
axis location measured from midchord
b=chord/2;                                          % panjang
setenghah chord dlm m
if alt==10
    rho=0.0017556/(12^4);                           % 10000 ft %
lbs^2/in^4 density of air
elseif alt==5
    rho=0.0020482/(12^4);                           % 5000 ft
else
    rho=0.0023769/(12^4);                           % Sea Level
end
%
d=R/p;                                              % panjang tiap
satu elemen
%
%natural frequency
asubn=[15.41820562 49.96486209 104.2476966];        % Hartog
K0=[6.38 17.63 35.05]; K1=[9.18 26.02 52.2];        % Yntema
BL=[3.926602 7.068583 10.21018];                    % bending fixed-
free
```

```
an=[(sin(BL(:,1))-sinh(BL(:,1)))/(cos(BL(:,1))-cosh(BL(:,1)))
(sin(BL(:,2))-sinh(BL(:,2)))/(cos(BL(:,2))-cosh(BL(:,2)))
(sin(BL(:,3))-sinh(BL(:,3)))/(cos(BL(:,3))-cosh(BL(:,3)))];
pf=cos(BL(:,1))*cosh(BL(:,1));                       %persamaan
frekuensi
%
%natural frekuensi tidak memutar
%
wnb=[(BL(:,1))^2*(sqrt((EIxx)/(mu*R^4)))
(BL(:,2))^2*(sqrt((EIxx)/(mu*R^4)))
(BL(:,3))^2*(sqrt((EIxx)/(mu*R^4)))];    %natural frekuensi bending
tidak memutar
wnt=[0.5 1.5 2.5]*pi*sqrt(GJ/(Ia*R^2));             %natural frekuensi
torsi tidak memutar
%
%natural frekuensi memutar
%
wnbp=sqrt(wnb.*wnb+(K0+K1*e).*O^2) ;                 %bending
wntp=(O.^2+wnt.^2).^(1/2)          ;                 %torsi
%
r(1)=0;                                              % r is the vector
of element edges in percent R
for q=1:p
    r(q+1)=q/p;
end
for n=1:p
    mid(n)=(r(n+1)+r(n))/2;                           % titik tengah
tiap elemen
end
%
%menentukan modeshape bending fixed-engsel
%
u=mid'*BL;
for n=1:3
    for m=1:p
        h=u(m,n);
        f(p+1-m,n)=(cosh(h)+cos(h)-an(n)*(sinh(h)+sin(h)))/2;
    end
end
%
for n=1:3                       % This section normalizes mode shape
    if f(1,n)<0                 % to all begin with a positive slope
        h=-1*f(:,n);            % for display purposes
        f(:,n)=h;               % (does not affect calculations)
    end
end
for n=1:3                       % This section calculates the
torsional mode shapes
    for m=1:p                   % According to Hartog (already
normalized to 1)
        F(m,n)=sin((n-.5)*pi*mid(m));
    end
end
%
%======================================================================
=====
```

```
%Plot mode shapes
%========================================================================
=====

figure(1)     % Plot of mode shapes
title('Mode Shapes untuk Bending dan Torsi')
plot(mid,f(:,1),mid,f(:,2),mid,f(:,3),mid,F(:,1),mid,F(:,2),mid,F(:,3)
)
grid on
axis([min(mid),max(mid),-1,1])
legend('f(1)','f(2)','f(3)','F(1)','F(2)','F(3)')
xlabel('normalized panjang beam');ylabel('Normalized Defleksi');

figure(2)   %mode shapes bending
title('Mode Shapes untuk Bending')
plot(mid,f(:,1),mid,f(:,2),mid,f(:,3))
grid on
axis([min(mid),max(mid),-1,1])
legend('f(1)','f(2)','f(3)')
xlabel('normalized panjang beam');ylabel('Normalized Defleksi
Bending');

figure(3)   %mode shapes torsi
title('Mode Shapes untuk Torsi')
plot(mid,F(:,1),mid,F(:,2),mid,F(:,3))
grid on
axis([min(mid),max(mid),-1,1])
legend('F(1)','F(2)','F(3)')
xlabel('normalized panjang beam');ylabel('Normalized Defleksi Torsi');
%
%
%normalize bending
fact1=sqrt(f(:,1)'*f(:,1)*mu*d);
fact2=sqrt(f(:,2)'*f(:,2)*mu*d);
fact3=sqrt(f(:,3)'*f(:,3)*mu*d);
wb(:,:)=f(:,:); %/fact1;
mb=[wb(:,1) wb(:,2) wb(:,3)]'*[wb(:,1) wb(:,2) wb(:,3)]*mu*d; %cek
apakah identitas
%
%normalized torsi
fact4=sqrt(F(:,1)'*F(:,1)*Ia*d);
fact5=sqrt(F(:,2)'*F(:,2)*Ia*d);
fact6=sqrt(F(:,3)'*F(:,3)*Ia*d);
wt(:,:)=F(:,:); %/fact4;
mt=[wt(:,1) wt(:,2) wt(:,3)]'*[wt(:,1) wt(:,2) wt(:,3)]*Ia*d; %cek
apakah identitas
%
Gmass=sum(mu*d*f'*f);
GTmass=sum(Ia*d*F'*F);
%generalized
z=zeros(p,3);
%modal
modal=[wb(:,1) wb(:,2) wb(:,3) z(:,1) z(:,2) z(:,3);z(:,1) z(:,2)
z(:,3) wt(:,1) wt(:,2) wt(:,3)];
%
Salf=mu*d*cgea*b;               % momen statik tiap elemen
```

```
%
%VALIDASI ROGER


y1=-0.3;
y2=-0.6;
y3=-1.1;
y4=-1.6;

w=[1 1;1 1];      %weighting factor
Ac=[zeros(7,7)];
b1=[zeros(7,1)];
b2=[zeros(7,1)];
b3=[zeros(7,1)];
b4=[zeros(7,1)];

%kr=0.068693:0.00055:0.123198;
%kr=[0.0001 0.0005 0.001 0.004 0.008 0.01 0.03 0.05 0.08 0.1 0.2 0.3
0.4 0.6 0.8 1 2 3 4 5]
kr=0.1:0.01:1
for m=1:length(kr);
    %m
    k=kr(m);
    H1=besselh(1,2,k);
    H0=besselh(0,2,k);
    Ck=H1/(H1+i*H0);      % Theodorsen’s function
    Lh(m)=1-2*i*Ck/k;    % Lift - plunge
    La(m)=0.5-2*i*(0.5+(1-i/k)*Ck)/k;  % Lift - pitch
    Mh(m)=0.5;          % Moment - plunge
    Ma(m)=3/8-i/k;        % Moment - pitch



    Ac=[1 0 -1*(k^2) ((k^2)./(k^2+y1^2)) (k^2)./(k^2+y2^2)
k^2./(k^2+y3^2) k^2./(k^2+y4^2); 0 k 0 k*y1./(k^2+y1^2)
k*y2./(k^2+y2^2) k*y3./(k^2+y3^2) k*y4./(k^2+y4.^2)]'*[1 0 -1*(k^2)
((k^2)./(k^2+y1^2)) (k^2)./(k^2+y2^2) k^2./(k^2+y3^2) k^2./(k^2+y4^2);
0 k 0 k*y1./(k^2+y1^2) k*y2./(k^2+y2^2) k*y3./(k^2+y3^2)
k*y4./(k^2+y4.^2)]+Ac;
    b1=[1 0 -1*(k^2) ((k^2)./(k^2+y1^2)) (k^2)./(k^2+y2^2)
k^2./(k^2+y3^2) k^2./(k^2+y4^2); 0 k 0 k*y1./(k^2+y1^2)
k*y2./(k^2+y2^2) k*y3./(k^2+y3^2)
k*y4./(k^2+y4.^2)]'*[real(Lh(m));imag(Lh(m))]+b1;
    b2=[1 0 -1*(k^2) ((k^2)./(k^2+y1^2)) (k^2)./(k^2+y2^2)
k^2./(k^2+y3^2) k^2./(k^2+y4^2); 0 k 0 k*y1./(k^2+y1^2)
k*y2./(k^2+y2^2) k*y3./(k^2+y3^2)
k*y4./(k^2+y4.^2)]'*[real(La(m));imag(La(m))]+b2;
    b3=[1 0 -1*(k^2) ((k^2)./(k^2+y1^2)) (k^2)./(k^2+y2^2)
k^2./(k^2+y3^2) k^2./(k^2+y4^2); 0 k 0 k*y1./(k^2+y1^2)
k*y2./(k^2+y2^2) k*y3./(k^2+y3^2)
k*y4./(k^2+y4.^2)]'*[real(Mh(m));imag(Mh(m))]+b3;
    b4=[1 0 -1*(k^2) ((k^2)./(k^2+y1^2)) (k^2)./(k^2+y2^2)
k^2./(k^2+y3^2) k^2./(k^2+y4^2); 0 k 0 k*y1./(k^2+y1^2)
k*y2./(k^2+y2^2) k*y3./(k^2+y3^2)
k*y4./(k^2+y4.^2)]'*[real(Ma(m));imag(Ma(m))]+b4;
```

```
end
%stop
%koefisien nilai roger
    A_Lhr=Ac\b1;
    dA_Lhr1=A_Lhr(1,:);
    dA_Lhr2=A_Lhr(2,:);
    dA_Lhr3=A_Lhr(3,:);
    dA_Lhr4=A_Lhr(4,:);
    dA_Lhr5=A_Lhr(5,:);
    dA_Lhr6=A_Lhr(6,:);
    dA_Lhr7=A_Lhr(7,:);
    A_Lar=Ac\b2;
    dA_Lar1=A_Lar(1,:);
    dA_Lar2=A_Lar(2,:);
    dA_Lar3=A_Lar(3,:);
    dA_Lar4=A_Lar(4,:);
    dA_Lar5=A_Lar(5,:);
    dA_Lar6=A_Lar(6,:);
    dA_Lar7=A_Lar(7,:);
    A_Mhr=Ac\b3;
    dA_Mhr1=A_Mhr(1,:);
    dA_Mhr2=A_Mhr(2,:);
    dA_Mhr3=A_Mhr(3,:);
    dA_Mhr4=A_Mhr(4,:);
    dA_Mhr5=A_Mhr(5,:);
    dA_Mhr6=A_Mhr(6,:);
    dA_Mhr7=A_Mhr(7,:);
    A_Mar=Ac\b4;
    dA_Mar1=A_Mar(1,:);
    dA_Mar2=A_Mar(2,:);
    dA_Mar3=A_Mar(3,:);
    dA_Mar4=A_Mar(4,:);
    dA_Mar5=A_Mar(5,:);
    dA_Mar6=A_Mar(6,:);
    dA_Mar7=A_Mar(7,:);

%nilai roger 4 lag
%for k=1:length(m)     % Run entire blade for range of values k at
blade tip
    %real roger
    Lh_roger_real=1*dA_Lhr1-
(kr.^2)*dA_Lhr3+((kr.^2)./(kr.^2+y1^2))*dA_Lhr4+(kr.^2)./(kr.^2+y2^2)*
dA_Lhr5+kr.^2./(kr.^2+y3^2)*dA_Lhr6+kr.^2./(kr.^2+y4^2)*dA_Lhr7;
    La_roger_real=1*dA_Lar1-
(kr.^2)*dA_Lar3+((kr.^2)./(kr.^2+y1^2))*dA_Lar4+(kr.^2)./(kr.^2+y2^2)*
dA_Lar5+kr.^2./(kr.^2+y3^2)*dA_Lar6+kr.^2./(kr.^2+y4^2)*dA_Lar7;
    Mh_roger_real=1*dA_Mhr1-
(kr.^2)*dA_Mhr3+((kr.^2)./(kr.^2+y1^2))*dA_Mhr4+(kr.^2)./(kr.^2+y2^2)*
dA_Mhr5+kr.^2./(kr.^2+y3^2)*dA_Mhr6+kr.^2./(kr.^2+y4^2)*dA_Mhr7;
    Ma_roger_real=1*dA_Mar1-
(kr.^2)*dA_Mar3+((kr.^2)./(kr.^2+y1^2))*dA_Mar4+(kr.^2)./(kr.^2+y2^2)*
dA_Mar5+kr.^2./(kr.^2+y3^2)*dA_Mar6+kr.^2./(kr.^2+y4^2)*dA_Mar7;

    %imaginer roger

Lh_roger_imag=0*dA_Lhr1+k*dA_Lhr2+0*dA_Lhr3+kr.*y1./(kr.^2+y1^2)*dA_Lh
```

```
r4+kr.*y2./(kr.^2+y2^2)*dA_Lhr5+kr.*y3./(kr.^2+y3^2)*dA_Lhr6+kr.*y4./(
kr.^2+y4.^2)*dA_Lhr7;

La_roger_imag=0*dA_Lar1+k*dA_Lar2+0*dA_Lar3+kr.*y1./(kr.^2+y1^2)*dA_La
r4+kr.*y2./(kr.^2+y2^2)*dA_Lar5+kr.*y3./(kr.^2+y3^2)*dA_Lar6+kr.*y4./(
kr.^2+y4.^2)*dA_Lar7;

Mh_roger_imag=0*dA_Mhr1+k*dA_Mhr2+0*dA_Mhr3+kr.*y1./(kr.^2+y1^2)*dA_Mh
r4+kr.*y2./(kr.^2+y2^2)*dA_Mhr5+kr.*y3./(kr.^2+y3^2)*dA_Mhr6+kr.*y4./(
kr.^2+y4.^2)*dA_Mhr7;

Ma_roger_imag=0*dA_Mar1+k*dA_Mar2+0*dA_Mar3+kr.*y1./(kr.^2+y1^2)*dA_Ma
r4+kr.*y2./(kr.^2+y2^2)*dA_Mar5+kr.*y3./(kr.^2+y3^2)*dA_Mar6+kr.*y4./(
kr.^2+y4.^2)*dA_Mar7;
%end

%======================================================================
=====
%PLOT ROGER
%======================================================================
=====

figure(4)
subplot(2,1,1)
title('Real Lh')
plot(kr,Lh_roger_real,kr,real(Lh))
legend('Lh roger','Lh Theodorsen')
%axis([min(kr),max(kr),0.75,1.05])
xlabel('reduced frequecy, k');ylabel('Lh');
grid on
%stop
    subplot(2,1,2)
    title('imaginer Lh')
    plot(kr,Lh_roger_imag,kr,imag(Lh))
    %plot(kr,imag(Lh))
    %axis([min(kr),max(kr),-1.2,0])
    legend('Lh roger','Lh Theodorsen')
    xlabel('reduced frequecy, k');ylabel('Lh');
    grid on
%stop
figure(5)
subplot(2,1,1)
title('Real La')
plot(kr,La_roger_real,kr,real(La))
%axis([min(kr),max(kr),-0.8,0.6])
legend('La roger','La Theodorsen')
xlabel('reduced frequecy, k');ylabel('La');
grid on
    subplot(2,1,2)
    title('Imaginer La')
    plot(kr,La_roger_imag,kr,imag(La))
    %axis([min(kr),max(kr),-2,0])
    legend('La roger','La Theodorsen')
    xlabel('reduced frequecy, k');ylabel('La');
    grid on
```

```
figure(6)
subplot(2,1,1)
title('Real Mh')
plot(kr,Mh_roger_real,kr,real(Mh))
%axis([min(kr),max(kr),0,1])
legend('Mh roger','Mh Theodorsen')
xlabel('reduced frequecy, k');ylabel('Mh');
grid on
    subplot(2,1,2)
    title('Imaginer Mh')
    plot(kr,Mh_roger_imag,kr,imag(Mh))
    %axis([min(kr),max(kr),0,1])
    legend('Mh roger','Mh Theodorsen')
    xlabel('reduced frequecy, k');ylabel('Mh');
    grid on

figure(7)
subplot(2,1,1)
title('Real Ma')
plot(kr,Ma_roger_real,kr,real(Ma))
legend('Ma roger','Ma Theodorsen')
%axis([min(kr),max(kr),0,0.5])
xlabel('reduced frequecy, k');ylabel('Ma');
grid on
    subplot(2,1,2)
    title('Imaginer Ma')
    plot(kr,Ma_roger_imag,kr,imag(Ma))
    %axis([min(kr),max(kr),-1,0])
    legend('Ma roger','Ma Theodorsen')
    xlabel('reduced frequecy, k');ylabel('Ma');
    grid on

for n=1:3                            % Calculate static moments for
mode coupling
    for m=1:3
        Sar(n,m)=sum(Salf*wb(:,n).*wt(:,m))/Gmass(1,n);
        Iaf(n,m)=sum(Ia*d*wt(:,n).*wt(:,m));
        Mma(n,m)=sum(mu*d*wb(:,n).*wb(:,m));
        Saf(n,m)=sum(Salf*wt(:,n).*wb(:,m))/GTmass(1,n);
    end
end
%
Identi=[1 0 0;
        0 1 0;
        0 0 1];
genmass=[Identi -Sar; -Saf Identi];
%stop
%genmass=[Mma Sar; Saf Iaf];
%
nlag=3;
nstate=3+nlag;
y1=0.2;
y2=0.4;
y3=0.6;
y4=0.8;
%
```

```
%mencari theodorsen
%
m=0.068693:0.00055:0.123198;
%
%roger untuk run program Aer
%
Vfwd=linspace(mn,mx,300);    % Set range of airspeeds for helicopter
%
ww=ones(7*p,7*p);
coefi=zeros(p,7,7);
rhlas=zeros(p,7);
rhlhs=zeros(p,7);
rhmas=zeros(p,7);
rhmhs=zeros(p,7);
%
for q=1:length(Vfwd)
    for t=1:p
        Vbaru=(Vfwd(q)+O*R*mid(p+1-t));
        kt=wntp(1)*b/Vbaru;
        H1=besselh(1,2,kt);
        H0=besselh(0,2,kt);
        Ck=H1/(H1+i*H0);                       % Theodorsen’s function
        Lh=(1-2*i*Ck/kt);                      % Lift - plunge
        La=(0.5-2*i*(0.5+(1-i/kt)*Ck)/kt);  % Lift - pitch
        Mh=(0.5);                              % Moment - plunge
        Ma=(3/8-i/kt);                         % Moment - pitch
        %
        Lhh= 2*pi*kt^2*d*Lh;
        Laa= 2*pi*kt^2*d*(La-(0.5+a)*Lh)*b;
        Mhh= 2*pi*kt^2*d*(Mh-(0.5+a)*Lh)*b;
        Maa= 2*pi*kt^2*d*(Ma-(0.5+a)*(La+Mh)+(0.5+a)^2*Lh)*b^2;
        %
        b11=[1  0  -1*(kt^2) ((kt^2)/(kt^2+y1^2)) (kt^2)/(kt^2+y2^2)
kt^2/(kt^2+y3^2)   kt^2./(kt^2+y4^2);
             0  kt    0      kt*y1/(kt^2+y1^2)     kt*y2/(kt^2+y2^2)
kt*y3/(kt^2+y3^2)  kt*y4/(kt^2+y4^2)];
        %
        bb=b11'*b11;
        %
        coef(t,:,:)=bb(:,:);
        %
        btlh=b11'*[real(Lhh) imag(Lhh)]';
        btla=b11'*[real(Laa) imag(Laa)]';
        btmh=b11'*[real(Mhh) imag(Mhh)]';
        btma=b11'*[real(Maa) imag(Maa)]';
        %
        rhlh(t,:)=btlh(:);
        rhla(t,:)=btla(:);
        rhma(t,:)=btma(:);
        rhmh(t,:)=btmh(:);
    end
    %pause
    coefi=coefi+coef;
    rhlhs=rhlhs+rhlh;
    rhlas=rhlas+rhla;
    rhmhs=rhmhs+rhmh;
```

```
    rhmas=rhmas+rhma;
end
%
for j=1:p
    bb(:,:)=coefi(j,:,:);
    rlh(:,1)=rhlhs(j,:);
    rla(:,1)=rhlas(j,:);
    rma(:,1)=rhmhs(j,:);
    rmh(:,1)=rhmhs(j,:);
%
    ALh=inv(bb)*rlh;
    ALa=inv(bb)*rla;
    AMh=inv(bb)*rmh;
    AMa=inv(bb)*rma;
%
    AALh(j,:)=ALh(:);
    AALa(j,:)=ALa(:);
    AAMh(j,:)=AMh(:);
    AAMa(j,:)=AMa(:);
end
%
%state space
%
Identity=[1 0 0 0 0 0;
          0 1 0 0 0 0;
          0 0 1 0 0 0;
          0 0 0 1 0 0;
          0 0 0 0 1 0;
          0 0 0 0 0 1];
%
for n=1:3
    for m=1:3
        A2(n,m)=0.5*rho*sum(AALh(:,3).*wb(:,n).*wb(:,m))/Gmass(1,n);
        A2(n,3+m)=0.5*rho*sum(AALa(:,3).*wb(:,n).*wt(:,m))/Gmass(1,n);

A2(3+n,m)=0.5*rho*sum(AAMh(:,3).*wt(:,n).*wb(:,m))/GTmass(1,n);

A2(3+n,3+m)=0.5*rho*sum(AAMh(:,3).*wt(:,n).*wt(:,m))/GTmass(1,n);
    end
end
%
Msp=genmass-b^2*A2;
%
Kec=linspace(1,1000);
%
fb=wb'*wb;
ft=wt'*wt;
%
for tr=1:length(Kec)
    vel=Kec(tr);
    for j=1:p
        Vf(j)=(vel+(O*R*mid(p+1-t)));
    end
    for n=1:3
        for m=1:3
```

```
A0(n,m)=0.5*rho*sum(Vf(:)'.^2*AALh(:,1).*wb(:,n).*wb(:,m))/Gmass(1,n);

A0(n,3+m)=0.5*rho*sum(Vf(:)'.^2*AALa(:,1).*wb(:,n).*wt(:,m))/Gmass(1,n
);

A0(3+n,m)=0.5*rho*sum(Vf(:)'.^2*AAMh(:,1).*wt(:,n).*wb(:,m))/GTmass(1,
n);

A0(3+n,3+m)=0.5*rho*sum(Vf(:)'.^2*AAMh(:,1).*wt(:,n).*wt(:,m))/GTmass(
1,n);
            %

A1(n,m)=0.5*rho*sum(Vf(:).*AALh(:,2).*wb(:,n).*wb(:,m))/Gmass(1,n);

A1(n,3+m)=0.5*rho*sum(Vf(:).*AALa(:,2).*wb(:,n).*wt(:,m))/Gmass(1,n);

A1(3+n,m)=0.5*rho*sum(Vf(:).*AAMh(:,2).*wt(:,n).*wb(:,m))/GTmass(1,n);

A1(3+n,3+m)=0.5*rho*sum(Vf(:).*AAMh(:,2).*wt(:,n).*wt(:,m))/GTmass(1,n
);
            %

A3(n,m)=0.5*rho*sum(Vf(:)'.^2*AALh(:,4).*wb(:,n).*wb(:,m))/Gmass(1,n);

A3(n,3+m)=0.5*rho*sum(Vf(:)'.^2*AALa(:,4).*wb(:,n).*wt(:,m))/Gmass(1,n
);

A3(3+n,m)=0.5*rho*sum(Vf(:)'.^2*AAMh(:,4).*wt(:,n).*wb(:,m))/GTmass(1,
n);

A3(3+n,3+m)=0.5*rho*sum(Vf(:)'.^2*AAMh(:,4).*wt(:,n).*wt(:,m))/GTmass(
1,n);
            %

A4(n,m)=0.5*rho*sum(Vf(:)'.^2*AALh(:,5).*wb(:,n).*wb(:,m))/Gmass(1,n);

A4(n,3+m)=0.5*rho*sum(Vf(:)'.^2*AALa(:,5).*wb(:,n).*wt(:,m))/Gmass(1,n
);

A4(3+n,m)=0.5*rho*sum(Vf(:)'.^2*AAMh(:,5).*wt(:,n).*wb(:,m))/GTmass(1,
n);

A4(3+n,3+m)=0.5*rho*sum(Vf(:)'.^2*AAMh(:,5).*wt(:,n).*wt(:,m))/GTmass(
1,n);
            %

A5(n,m)=0.5*rho*sum(Vf(:)'.^2*AALh(:,6).*wb(:,n).*wb(:,m))/Gmass(1,n);

A5(n,3+m)=0.5*rho*sum(Vf(:)'.^2*AALa(:,6).*wb(:,n).*wt(:,m))/Gmass(1,n
);

A5(3+n,m)=0.5*rho*sum(Vf(:)'.^2*AAMh(:,6).*wt(:,n).*wb(:,m))/GTmass(1,
n);
```

```
A5(3+n,3+m)=0.5*rho*sum(Vf(:)'.^2*AAMh(:,6).*wt(:,n).*wt(:,m))/GTmass(
1,n);
            %

A6(n,m)=0.5*rho*sum(Vf(:)'.^2*AALh(:,7).*wb(:,n).*wb(:,m))/Gmass(1,n);

A6(n,3+m)=0.5*rho*sum(Vf(:)'.^2*AALa(:,7).*wb(:,n).*wt(:,m))/Gmass(1,n
);

A6(3+n,m)=0.5*rho*sum(Vf(:)'.^2*AAMh(:,7).*wt(:,n).*wb(:,m))/GTmass(1,
n);

A6(3+n,3+m)=0.5*rho*sum(Vf(:)'.^2*AAMh(:,7).*wt(:,n).*wt(:,m))/GTmass(
1,n);
        end
    end
%
    Dsp=-b*A1;
    Ksp=[wnbp(1).^2 0 0 0 0 0;
         0 wnbp(2).^2 0 0 0 0;
         0 0 wnbp(3).^2 0 0 0;
         0 0 0 wntp(1).^2 0 0;
         0 0 0 0 wntp(2).^2 0;
         0 0 0 0 0 wntp(3).^2]-(A0);
%
%matriks baru
%
    Ivev=zeros(6,6);
    for n=1:3                            % Calculate static moments
for mode coupling
        Ivev(n,n)=sum(Vf(:).*wb(:,n).*wb(:,n))/fb(n,n);
        Ivev(3+n,3+n)=sum(Vf(:).*wt(:,n).*wt(:,n))/ft(n,n);
    end
    %
    MatrikSP=[zeros(6,6)            Identity      zeros(6,6)
zeros(6,6)  zeros(6,6)  zeros(6,6);
             -1*(inv(Msp)*Ksp) -1*(inv(Msp)*Dsp) inv(Msp)*A3
inv(Msp)*A4 inv(Msp)*A5 inv(Msp)*A6;
              zeros(6,6)            Identity      -Ivev*y1/b
zeros(6,6)  zeros(6,6)  zeros(6,6);
              zeros(6,6)            Identity      zeros(6,6)  -
Ivev*y2/b  zeros(6,6)  zeros(6,6);
              zeros(6,6)            Identity      zeros(6,6)
zeros(6,6)  -Ivev*y3/b  zeros(6,6);
              zeros(6,6)            Identity      zeros(6,6)
zeros(6,6)  zeros(6,6)  -Ivev*y4/b];
%
% menghitung eigenvalues
%
    Z(tr,:)=eig(MatrikSP(:,:),('nobalance'));
    [Zz(tr,:),index]=sort(imag(Z(tr,:)));
    Z1(tr,:)=Z(tr,index);
    ZR(tr,:)=real(Z1(tr,:));
    ZI(tr,:)=imag(Z1(tr,:));
    for jj=1:6
```

```
        freqi(tr,jj)=ZI(tr,30+jj);
        dampi(tr,jj)=-
ZR(tr,30+jj)/sqrt(ZI(tr,30+jj)^2+ZR(tr,30+jj)^2);
    end
    freq(tr,1)=freqi(tr,1);
    freq(tr,2)=freqi(tr,2);
    freq(tr,3)=freqi(tr,4);
    damp(tr,1)=dampi(tr,1);
    damp(tr,2)=dampi(tr,2);
    damp(tr,3)=dampi(tr,4);
    %Z1'
    %ZI'
    %
    %pause
end
%

%stop
%ZR
%pause
Vf'
Kec'
freq
damp
%pause
%
for tr=1:1 :length(Kec)
        Vef(tr)=(Kec(tr)+(O*R))./12;
end
%size(MatrikSP)
%size(eig(MatrikSP))
%
%======================================================================
=====
%plot Vg dan Vf
%======================================================================
=====
%
%for tr=1:1 :length(Kec)
%    Kec(tr), freq(tr,:)
%end
ktip=b*wntp(1)./(Vfwd+O*R);         % Define reduced frequency for tip
x=1./ktip;
if mode==3
    V2=(freq(tr,2)*b/12).*x; % velocity corresponding to actual oTR(1)
in fwd flight
else
    V2=(freq(tr,4)*b/12).*x; % velocity corresponding to actual oTR(1)
in fwd flight
end

figure(8)
subplot(2,1,1)
plot(Vef,freq)
grid on
xlabel('Velocity(ft/s)'); ylabel('freq');
```

```
%legend('bending 1','bending 2','bending 3','torsi 1','torsi 2','torsi
3')
%axis([0,50,50,400])
subplot(2,1,2)
plot(Vef,damp)
grid on
xlabel('Velocity (ft/s)'); ylabel('damp');
%legend('bending 1','bending 2','bending 3','torsi 1','torsi 2','torsi
3')
%axis([0,50,-400,0])

figure(9)
plot(-ZR,ZI)
grid on
xlabel('real'); ylabel('imag');
legend('bending 1','bending 2','bending 3','torsi 1','torsi 2','torsi
3')
axis([-100,100,-900,900])
%
vel=0;
%
for j=1:p
    Vf(j)=(vel+(O*R*mid(p+1-t)));
end
for n=1:3
    for m=1:3

A0(n,m)=0.5*rho*sum(Vf(:)'.^2*AALh(:,1).*wb(:,n).*wb(:,m))/Gmass(1,n);

A0(n,3+m)=0.5*rho*sum(Vf(:)'.^2*AALa(:,1).*wb(:,n).*wt(:,m))/Gmass(1,n
);

A0(3+n,m)=0.5*rho*sum(Vf(:)'.^2*AAMh(:,1).*wt(:,n).*wb(:,m))/GTmass(1,
n);

A0(3+n,3+m)=0.5*rho*sum(Vf(:)'.^2*AAMh(:,1).*wt(:,n).*wt(:,m))/GTmass(
1,n);
            %

A1(n,m)=0.5*rho*sum(Vf(:).*AALh(:,2).*wb(:,n).*wb(:,m))/Gmass(1,n);

A1(n,3+m)=0.5*rho*sum(Vf(:).*AALa(:,2).*wb(:,n).*wt(:,m))/Gmass(1,n);

A1(3+n,m)=0.5*rho*sum(Vf(:).*AAMh(:,2).*wt(:,n).*wb(:,m))/GTmass(1,n);

A1(3+n,3+m)=0.5*rho*sum(Vf(:).*AAMh(:,2).*wt(:,n).*wt(:,m))/GTmass(1,n
);
            %

A3(n,m)=0.5*rho*sum(Vf(:)'.^2*AALh(:,4).*wb(:,n).*wb(:,m))/Gmass(1,n);

A3(n,3+m)=0.5*rho*sum(Vf(:)'.^2*AALa(:,4).*wb(:,n).*wt(:,m))/Gmass(1,n
);

A3(3+n,m)=0.5*rho*sum(Vf(:)'.^2*AAMh(:,4).*wt(:,n).*wb(:,m))/GTmass(1,
n);
```

```
A3(3+n,3+m)=0.5*rho*sum(Vf(:)'.^2*AAMh(:,4).*wt(:,n).*wt(:,m))/GTmass(
1,n);
            %

A4(n,m)=0.5*rho*sum(Vf(:)'.^2*AALh(:,5).*wb(:,n).*wb(:,m))/Gmass(1,n);

A4(n,3+m)=0.5*rho*sum(Vf(:)'.^2*AALa(:,5).*wb(:,n).*wt(:,m))/Gmass(1,n
);

A4(3+n,m)=0.5*rho*sum(Vf(:)'.^2*AAMh(:,5).*wt(:,n).*wb(:,m))/GTmass(1,
n);

A4(3+n,3+m)=0.5*rho*sum(Vf(:)'.^2*AAMh(:,5).*wt(:,n).*wt(:,m))/GTmass(
1,n);
            %

A5(n,m)=0.5*rho*sum(Vf(:)'.^2*AALh(:,6).*wb(:,n).*wb(:,m))/Gmass(1,n);

A5(n,3+m)=0.5*rho*sum(Vf(:)'.^2*AALa(:,6).*wb(:,n).*wt(:,m))/Gmass(1,n
);

A5(3+n,m)=0.5*rho*sum(Vf(:)'.^2*AAMh(:,6).*wt(:,n).*wb(:,m))/GTmass(1,
n);

A5(3+n,3+m)=0.5*rho*sum(Vf(:)'.^2*AAMh(:,6).*wt(:,n).*wt(:,m))/GTmass(
1,n);
            %

A6(n,m)=0.5*rho*sum(Vf(:)'.^2*AALh(:,7).*wb(:,n).*wb(:,m))/Gmass(1,n);

A6(n,3+m)=0.5*rho*sum(Vf(:)'.^2*AALa(:,7).*wb(:,n).*wt(:,m))/Gmass(1,n
);

A6(3+n,m)=0.5*rho*sum(Vf(:)'.^2*AAMh(:,7).*wt(:,n).*wb(:,m))/GTmass(1,
n);

A6(3+n,3+m)=0.5*rho*sum(Vf(:)'.^2*AAMh(:,7).*wt(:,n).*wt(:,m))/GTmass(
1,n);
   end
end
%
    Dsp=-b*A1;
    Ksp=[wnbp(1).^2 0 0 0 0 0;
         0 wnbp(2).^2 0 0 0 0;
         0 0 wnbp(3).^2 0 0 0;
         0 0 0 wntp(1).^2 0 0;
         0 0 0 0 wntp(2).^2 0;
         0 0 0 0 0 wntp(3).^2]-(A0);
%
%matriks baru
%
Ivev=zeros(6,6);
for n=1:3                            % Calculate static moments for
mode coupling
        Ivev(n,n)=sum(Vf(:).*wb(:,n).*wb(:,n))/fb(n,n);
```

```
        Ivev(3+n,3+n)=sum(Vf(:).*wt(:,n).*wt(:,n))/ft(n,n);
end
%
MatrikSP=[zeros(6,6)            Identity      zeros(6,6)  zeros(6,6)
zeros(6,6)  zeros(6,6);
          -1*(inv(Msp)*Ksp) -1*(inv(Msp)*Dsp) inv(Msp)*A3  inv(Msp)*A4
inv(Msp)*A5 inv(Msp)*A6;
          zeros(6,6)            Identity      -Ivev*y1/b  zeros(6,6)
zeros(6,6)  zeros(6,6);
          zeros(6,6)            Identity      zeros(6,6)  -Ivev*y2/b
zeros(6,6)  zeros(6,6);
          zeros(6,6)            Identity      zeros(6,6)  zeros(6,6)
-Ivev*y3/b  zeros(6,6);
          zeros(6,6)            Identity      zeros(6,6)  zeros(6,6)
zeros(6,6)  -Ivev*y4/b];
%
for k=1:36
    xsp0(k)=0;
end
xsp0(1)=0.5;
t=0:0.01:2;
C=[1 zeros(1,35)];
sys = ss(MatrikSP,[],C,[]);
%initial(sys,xsp0)
[y,t,x] = initial(sys,xsp0,t)
%initialplot(sys,xsp0,t)
%hh(1)=0;
%hh
figure(10)
%subplot(2,1,1)
plot(t,y)
grid on
%subplot(2,1,2)
%plot(t,dispp(50,t))
%grid on
```