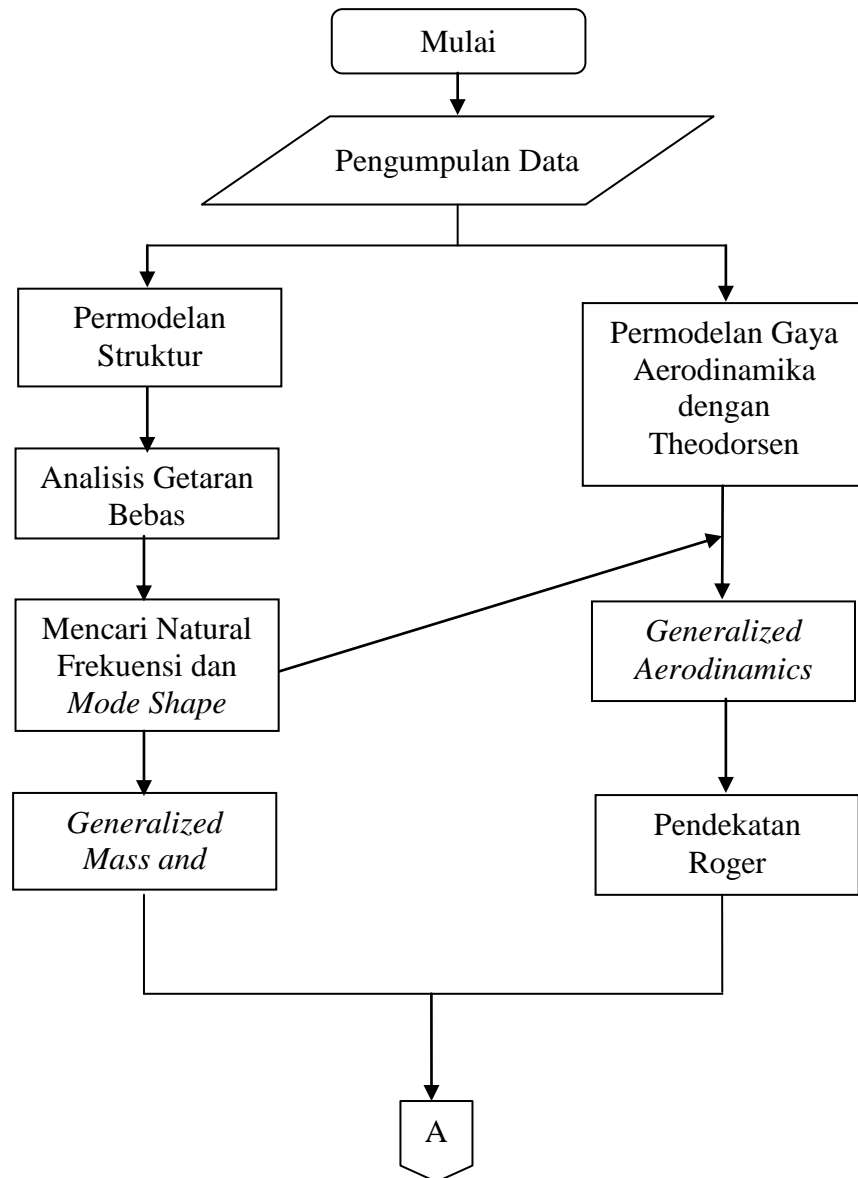
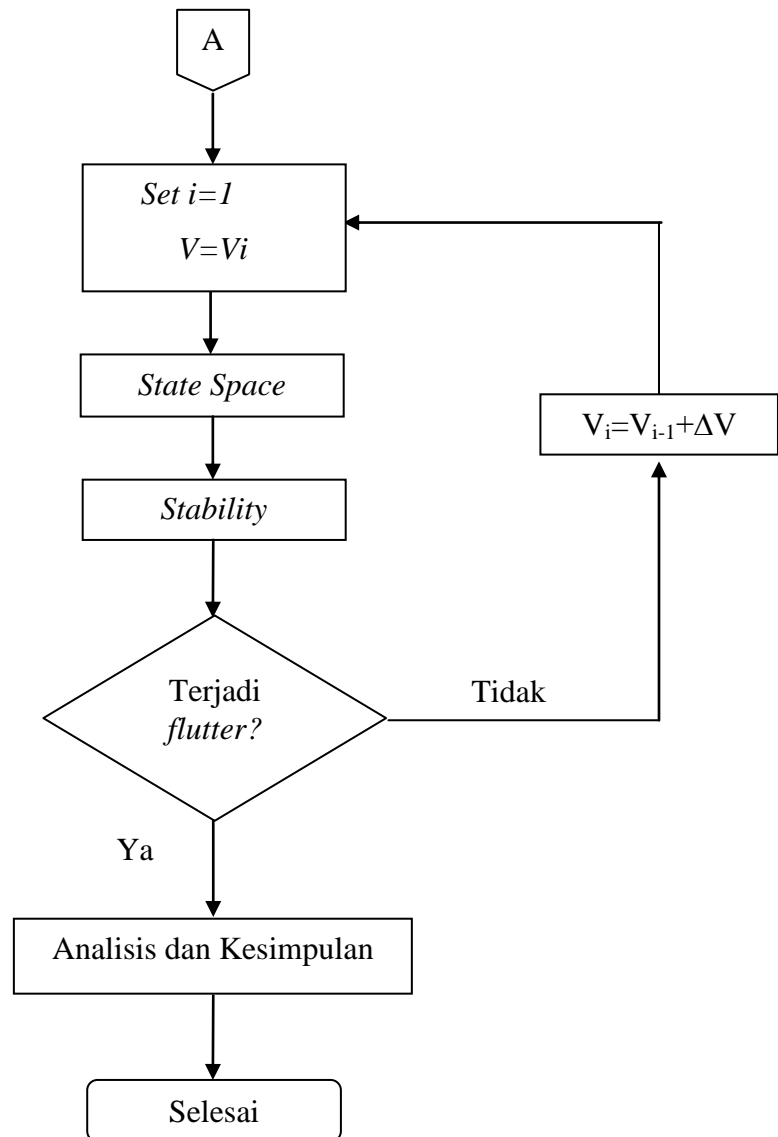


BAB III
METODOLOGI PENELITIAN

3.1 Metodologi Pencarian *Flutter*

3.1.1. Diagram Alir





Gambar 3.1 Diagram alir penelitian

3.1.2. Asumsi Model

Untuk mempermudah analisis menentukan kecepatan flutter maka perlu dilakukan penyederhanaan dalam system. Asumsi membuat model yang sederhana memungkinkan evaluasi masing-masing komponen model dan pembelajaran tentang persamaan dan perbedaan antara model dan sistem fisiologis yang sebenarnya. Secara keseluruhan

sistem blade dimodelkan sebagai kondisi unsteady-state. Berikut beberapa asumsi pada penelitian ini karena dianggap sesuai dengan model referensi, yaitu:

1. Spesifikasi teknis blade menggunakan blade UH-60
2. Blade dianggap batang kantilever dimana salah satu ujungnya dijepit dan salah satunya bebas.
3. Blade dibagi menjadi 100 elemen terdeskritisasi dimana tiap-tiap elemennya terbagi menjadi dua variasi kebebasan yaitu gerak vertical dan torsional sehingga ada 200 derajat kebebasan.
4. Guna mengetahui kecepatan flutter, maka kecepatan flutter di variasikan dari 0-1000 ft/s
5. Pemodelan dan simulasi menggunakan bantuan *software* Matlab R2009A

3.2 Formulasi Matematis Analisis Getaran Bebas

3.2.1. Bentuk Modus dan Frekuensi Pribadi

Frekuensi bending dan *mode shapes* digunakan dalam tugas akhir ini ditentukan dengan metode sederhana yaitu dengan mengasumsikan kekakuan batang dan distribusi massa adalah seragam (*uniform*) sepanjang *blade*. Metode eksak dan analisis digunakan untuk mencari bentuk modus. Adapun kondisi batas yang digunakan disini adalah kondisi batas untuk batang jepit-bebas.

Bentuk modus yang digunakan dalam analisis ini ditentukan dengan metode yang disederhanakan dengan asumsi kekakuan seragam dan distribusi massa di sepanjang *blade*. Sebuah analisis yang lebih tepat dan rinci akan diperlukan untuk menjelaskan rincian seperti perubahan lokal dalam distribusi kekakuan dan massa karena fitur-fitur seperti *blade* doublers dekat akar *blade* dan berat keseimbangan tempel *blade* dekat ujung. Sementara ini rincian akan diinginkan untuk desain *blade* yang sebenarnya, mereka dapat dilihat sebagai urutan kedua efek dan tidak diperlukan untuk analisis *flutter* urutan pertama. Untuk model disederhanakan, properti geometrik dan inersia dari *blade* dirata-ratakan antara 20% dan 90% dari panjangnya. Solusi berbasis Fourier dari disematkan bebas seragam balok dari Young dan Felgar diterapkan untuk menentukan bentuk modus non-rotating

$$f_n(x) = C_n (\sin(\beta_n x) - \sinh(\beta_n x) - \alpha_n (\cos(\beta_n x) - \cosh(\beta_n x))) \quad (3.1)$$

Dimana

$$\alpha_n = \frac{\sin(\beta_n L) + \sinh(\beta_n L)}{\cos(\beta_n L) - \cosh(\beta_n L)} \text{ untuk modulus tiga pertama dari bending.} \quad (3.2)$$

$$\beta_n L \text{ modulus ke-1} = 1.875104$$

$$\beta_n L \text{ modulus ke-2} = 4.694091$$

$$\beta_n L \text{ modulus ke-3} = 7.854757$$

kemudian untuk menentukan frekuensi pribadi non-rotating maka:

$$\omega_{nb} = (\beta_n L)^2 \sqrt{\frac{EI_{xx}}{\mu R^4}} \quad (3.3)$$

dimana

EI_{xx} = Elastisitas Inersia

μ = massa tiap elemen panjang

R = panjang blade

$\beta_n L$ = konstanta bentuk modulus

Adapun untuk kondisi blade yang memutar maka rumus untuk mencari frekuensi pribadi pada sistem berubah yaitu:

$$\omega_{Rb} = \omega_{nb} + K_n \Omega^2 \quad (3.4)$$

dimana

K_n = koefisien southwell

Ω = kecepatan rotor

Masalah berikutnya adalah menentukan bentuk modulus dan frekuensi pribadi model blade beam yang uniform dalam kasus torsional. Untuk kasus ini, Den Hartog mempresentasikan solusi yang lebih sederhana. Dengan menganggap kekakuan sepanjang beam adalah konstan maka persamaan bentuk modulus nya

$$F_n = \sin \left[\frac{\left(n - \frac{1}{2} \right) \pi r}{R} \right] \quad (3.5)$$

dimana

r = titik tengah tiap elemen beam

n = konstanta 1, 2, 3, dst

sedangkan frekuensi pribadi nya adalah

$$\omega_{nt} = \left(n - \frac{1}{2} \right) \pi \sqrt{\frac{GJ}{I_a R^2}} \quad (3.6)$$

$$\omega_{Rt} = \omega_{Rt} + \Omega^2$$

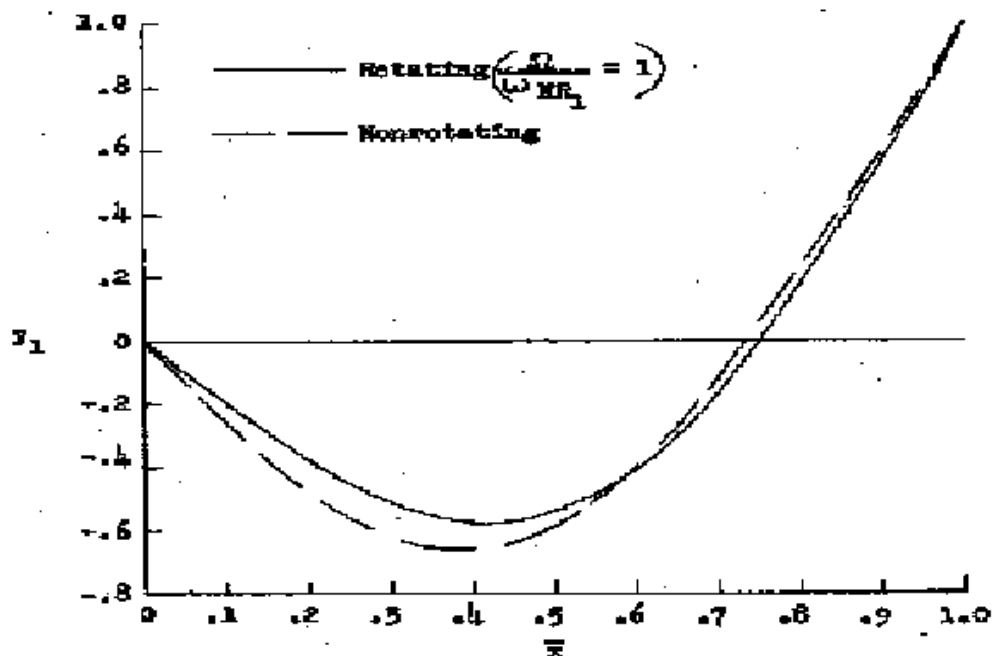
dimana

ω_{nt} = frekuensi pribadi non-rotating torsi

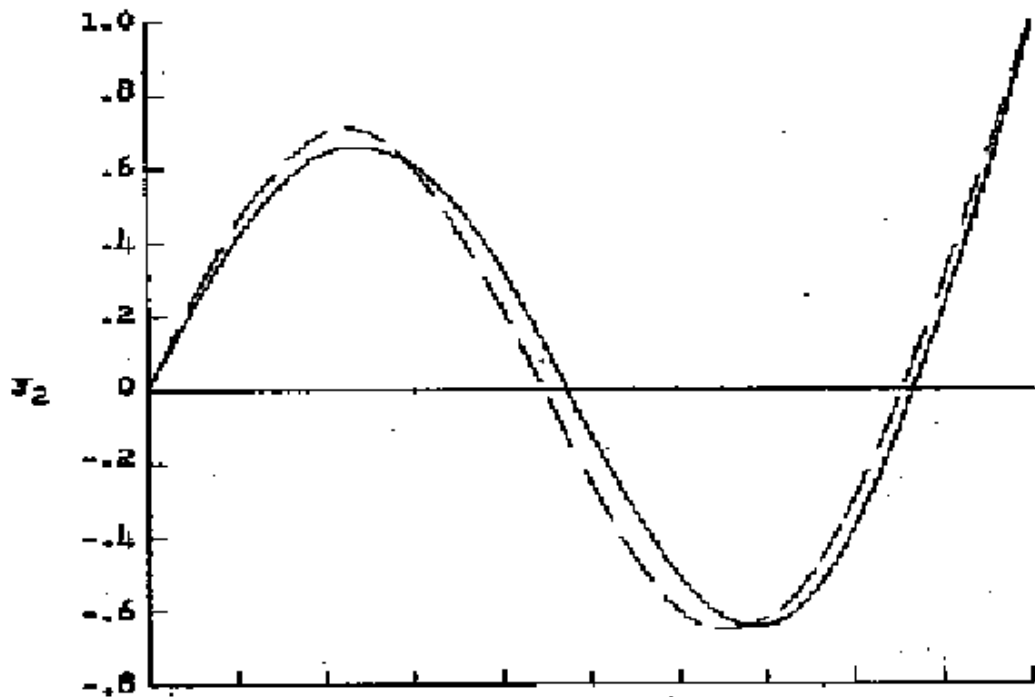
ω_{Rt} = frekuensi pribadi rotating torsi

GJ = inersia polar

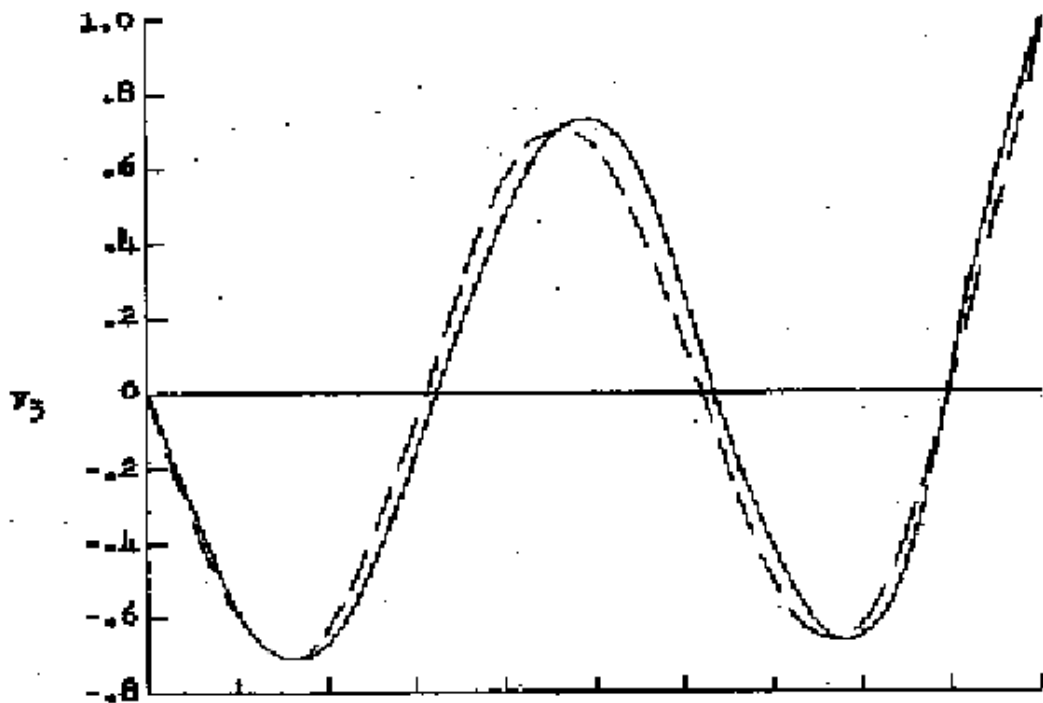
Dalam penelitian ini menggunakan enam bentuk modulus, yaitu tiga bentuk modulus bending dan tiga bentuk modulus torsi. Adapun bentuk modulus digambarkan sebagai berikut:



Gambar 3.2 Bentuk modulus untuk modulus bending pertama



Gambar 3.3 Bentuk modus untuk modus bending kedua



Gambar 3.4 Bentuk modus untuk modus bending ketiga

3.2.2. Matriks Massa, Redaman dan Kekakuan

Pada tahapan ini, akan membentuk matriks yang digunakan untuk menghitung model sistem blade. Adapun rumus persamaan getaran untuk flutter telah di presentasikan pada persamaan (2.4) dan persamaan (2.5) untuk Matriks penyusunnya. Di dalam sub bab ini dibahas lebih detail tentang matriks pembangun dari matriks massa, redaman dan kekakuan.

- Matriks Massa

$$[M_s] = \begin{bmatrix} [m] & [S_\alpha] \\ [S_\alpha] & [I_\alpha] \end{bmatrix} \quad (3.7)$$

$$m = \sum (\mu \cdot d \cdot \omega_{Rb} \omega_{Rb}) \quad (3.8)$$

$$S_\alpha = \sum \left(\frac{S_{al} \cdot \omega_{Rb} \omega_{Rb}}{G_m} \right) \quad (3.9)$$

$$\text{Dimana} \quad (3.10)$$

$$S_{al} = \mu \cdot d \cdot c_{gea} \cdot b \quad (3.11)$$

$$G_m = \sum (\mu \cdot d \cdot f^* f) \quad (3.12)$$

$$I_\alpha = \sum (I_{al} \cdot d \cdot \omega_{Rb} \omega_{Rb})$$

dimana

S_{al} = momen statik tiap elemen

μ = massa tiap elemen panjang

d = panjang tiap elemen

c_{gea} = jarak antara CG (pusat gravitasi) dan EA (sumbu elastik)

b = setengah chord

I_{al} = inersia massa

- Matriks Redaman

Dalam penelitian tugas akhir ini, sistem dianggap sebagai batang cantilever tak teredam sehingga getaran awal sistem adalah 0.

- Matriks Kekakuan

$$[K_s] = \begin{bmatrix} K_h & 0 \\ 0 & K_h \end{bmatrix} \text{ dimana } K_h \text{ adalah input blade UH-60.}$$

- Matriks Gaya Aerodinamik

Adapun matriks gaya aerodinamik sesuai dengan persamaan (2.5) secara lebih lanjut dijelaskan pada sub bab 3.3 untuk penentuan aerodinamik tak tunaknya.

3.3 Pendekatan Roger

Analisis *flutter* dengan metode-p dasarnya adalah suatu analisis yang menggunakan pendekatan rasional untuk mendapatkan koefisien aerodinamika tak tunak, sehingga masukanannya tidak lagi frekuensi melainkan kecepatan. Permasalahan aerodinamika tak tunak merupakan permasalahan yang cukup sulit diselesaikan akan tetapi dapat dicari dengan formulasi *Theodorsen*. Selanjutnya diterapkan pendekatan roger guna mendapatkan nilai koefisien aerodinamika tak tunak tersebut dalam domain waktu.

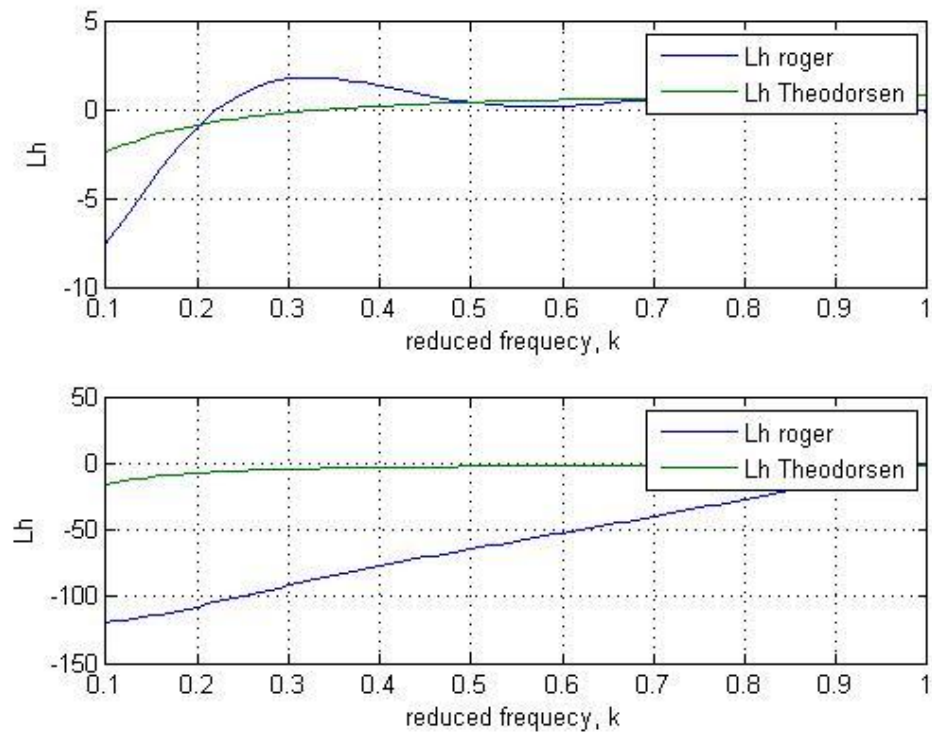
Pada analisis *flutter* dengan metode-p ada beberapa prosedur yang dilakukan, yaitu:

1. Selesaikan masalah aerodinamika tak tunak dengan pendekatan roger dengan mencari nilai koefisien-koefisien (A0, A1, A2, A3, A4, A5, A6) dengan prosedur seperti yang telah dijelaskan pada bab 2
2. Aplikasikan solusi pendekatan roger pada langkah 1 terhadap persamaan aeroelastik *flutter*
3. Transformasikan persamaan aeroelastik menjadi domain waktu kemudian nyatakan persamaan dalam bentuk state-space
4. Tentukan kecepatan udara V
5. Cari nilai eigen dari matriks dinamik [A]
6. Ambil nilai eigen yang kompleks tersebut dan pisahkan antara bagian riil dan imajiner. Yang mana bagian imajiner menunjukkan frekuensi dan bagian riil menunjukkan artificial damping.

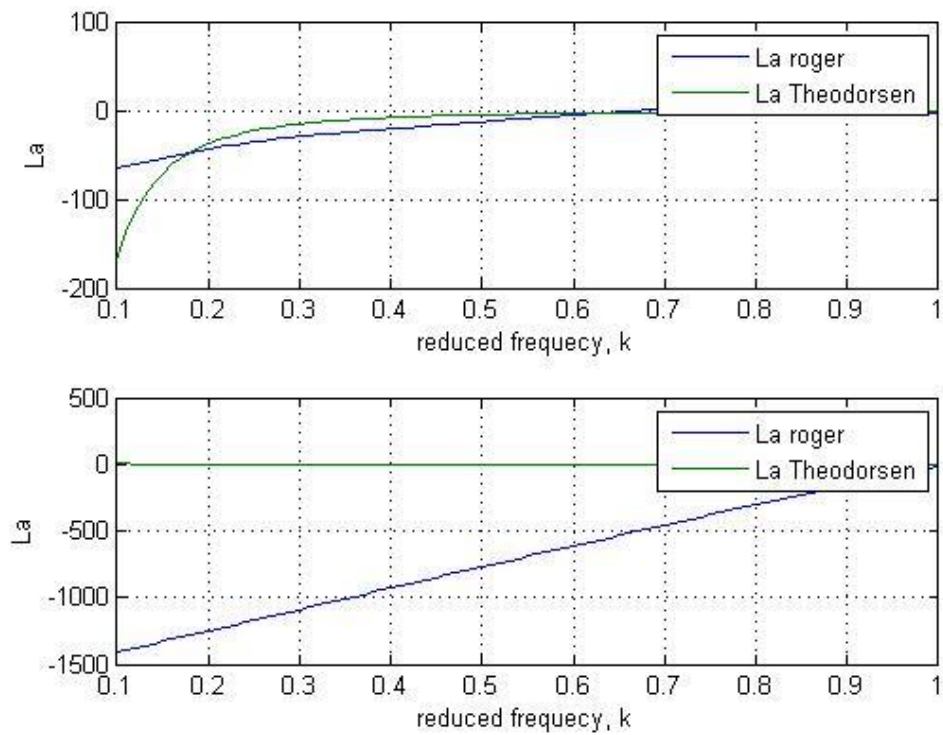
7. Plot nilai eigen tersebut pada diagram *flutter*
8. Jika nilai eigen bagian riil yang menunjukkan redaman berharga nol maka kecepatan ditambah dan kembali ke langkah 4

Dengan menggunakan metode-p ini keadaan *flutter* dapat juga disajikan dalam bentuk root locus yaitu dengan cara menggambarkan nilai eigen riil dan imajiner untuk tiap masukan kecepatan.

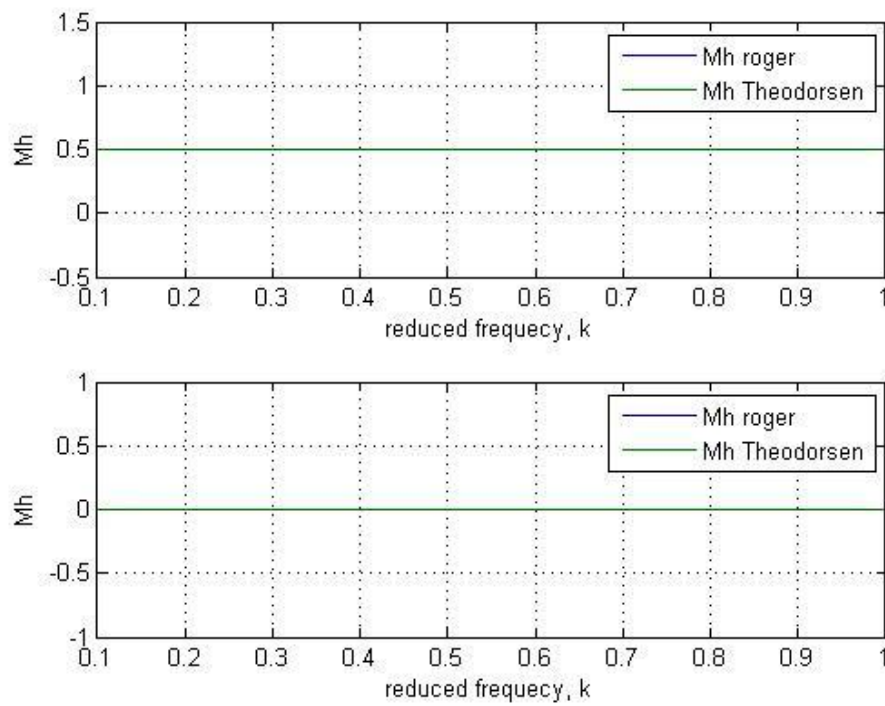
Validasi hasil menggunakan pendekatan roger dilakukan dengan memperhitungkan nilai lag kemudian di buat suatu diagram V-f. (kecepatan-frekuensi) dan V-g (kecepatan-redaman).



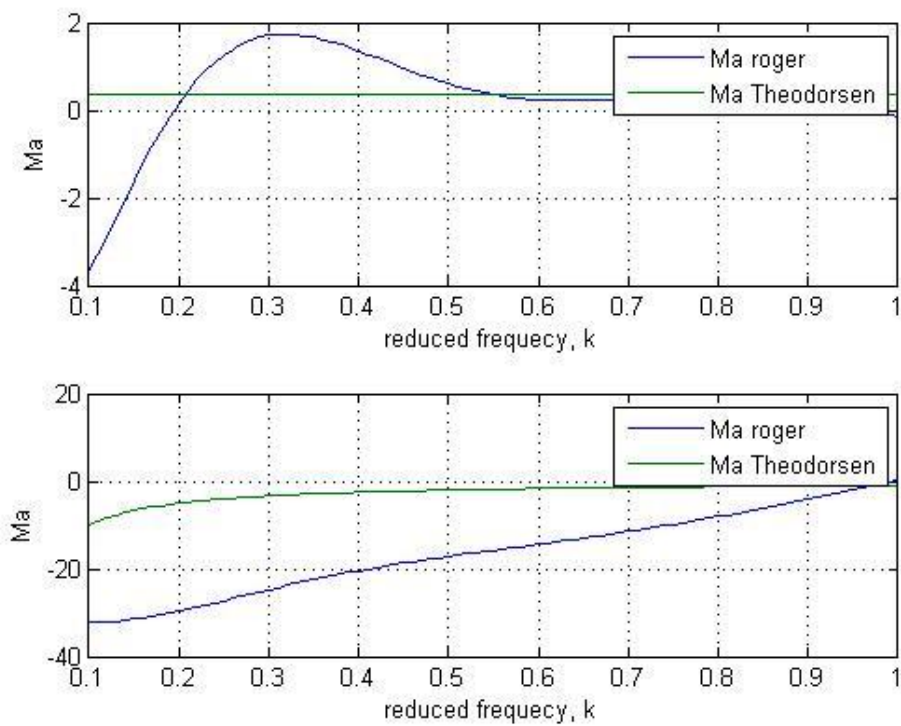
Gambar 3.5 nilai *real* dan *imaginer* untuk Lh



Gambar 3.2 nilai *real* dan *imaginer* untuk La



Gambar 3.3 nilai *real* dan *imaginer* untuk Mh



Gambar 3.4 nilai *real* dan *imaginer* untuk Ma

3.4 Menentukan Nilai Eigen

Setelah melakukan validasi pendekatan roger maka langkah berikutnya adalah membuat matriks untuk menentukan nilai eigen pada struktur. Sebagaimana yang telah dibahas pada sub bab 2.3.2 tentang pendekatan roger berikut dengan persamaan (2.26) sampai dengan (2.40), setelah didapatkan matriks baru pada persamaan 2.35 maka mencari nilai eigen nya. Adapun pentransformasikan persamaan tersebut ke M-file Matlab adalah sebagai berikut.

```

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);

```

Gambar 3.5 Kode MATLAB menghitung nilai eigen

Adapun nilai eigen pada kode matlab gambar 3.5 menghasilkan 6 macam variasi dalam menentukan titik flutter. Setelah itu menggambarkannya dalam bentuk root locus dan analisa stabilitas dengan menggambarkannya ke dalam diagram V-f dan V-g. kode MATLAB nya sebagai berikut.

```

%=====
%plot v-g dan v-f
%=====
%
%for tr=1:1 :length(Kec)
%   Kec(tr), freq(tr,⊙)
%end
ktip=b*wntp(1)./(Vfwd+0*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','tors1 1','tors1
2','tors1 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','tors1 1','tors1
2','tors1 3')
%axis([0,50,-400,0])

figure(9)
plot(-ZR,ZI)

```

```

grid on
xlabel('real'); ylabel('imag');
legend('bending 1','bending 2','bending 3','torsion 1','torsion 2','torsion 3')
axis([-100,100,-900,900])
%
vel=150;
%
for j=1:p
    vf(j)=(vel+(0*R*mid(p+1-t)));
end
for n=1:3
    for m=1:3
        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,[]);
        [y,t,x] = initial(sys,xsp0,t)
        figure(10)
        plot(t,y)
        grid on

```

Gambar 3.6 kode program untuk plot root locus dan diagram V-f dan V-g