

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1. Kesimpulan

Setelah melalui tahap analisis, maka dapat diperoleh beberapa kesimpulan sebagai berikut:

1. Hasil identifikasi parameter modal struktur frame adalah: (a) analisis simulasi numerik, yaitu  $\omega_{n1} = 12.730 \text{ rad/s}$ ,  $\omega_{n2} = 33.5074 \text{ rad/s}$  dan  $\xi_1 = 0.0234$ ,  $\xi_2 = 0.0475$  (b) analisis DD-SSI model simulasi yaitu  $\omega_{n1} = 12.743 \text{ rad/s}$ ,  $\omega_{n2} = 35.751 \text{ rad/s}$  dan  $\xi_1 = 0.00192$ ,  $\xi_2 = 0.0505$ ; (c) analisis DD-SSI model eksperimen  $\omega_{n1} = 12.359 \text{ rad/s}$ ,  $\omega_{n2} = 36.455 \text{ rad/s}$  dan  $\xi_1 = 0.0205$ ,  $\xi_2 = 0.0553$ .
2. Hasil identifikasi parameter modal struktur tower adalah: (a) analisis simulasi numerik, yaitu  $\omega_n = 46.0380 \text{ rad/s}$ ,  $\xi = 0.0046$ , (b) analisis DD-SSI model simulasi yaitu  $\omega_n = 47.964 \text{ rad/s}$ ,  $\xi = 0.00365$ ; (c) analisis DD-SSI model eksperimen  $\omega_n = 48.0741 \text{ rad/s}$ ,  $\xi_1 = 0.00394$ .
3. Hasil perbandingan parameter modal pada struktur model *frame* dan *tower* ditunjukkan pada Tabel 2.2 sebagai berikut;

Tabel 2.2. Perbandingan parameter modal pada struktur model *frame* dan *tower*

Metode	Frame		$\xi$	Tower	
	Mode	$\omega_n$ (rad/s)		$\omega_n$ (rad/s)	$\xi$
Simulasi Numerik/ DD-SSI Simulasi	1	0.1%	17.97%	4.18%	20.65%
	2	6.5%	6.32%		
Simulasi Numerik/ DD-SSI Model Experimen	1	3%	12%	4.42%	14.35%
	2	9%	16%		
DD-SSI Model Simulasi/ DD-SSI Model Experimen	1	3%	7%	0.23%	7.95%
	2	2%	10%		

4. Perbandingan rasio redaman struktur pada penelitian hanyalah sebuah asumsi bukan merupakan rasio redaman *real* struktur, sehingga kurang tepat dijadikan standar untuk dibandingkan dengan hasil analisis metode DD-SSI.
5. Identifikasi parameter dinamik struktur dengan metode DD-SSI menghasilkan frekuensi alami yang sangat akurat (dibawah 10%) pada kedua model.
6. Analisis Metode DD-SSI pada model *frame* simulasi pada mode pertama sulit terdeteksi. Sedangkan pada model tower sangat jelas dan akurat.
7. Penggunaan Metode DD-SSI membutuhkan RAM komputer yang besar karena harus melalui tahap *trial* pada *input* ordo maksimum sistem ( $n$ ) dan besaran blok baris ( $x$ ) sampai menemukan pola pola yang stabil pada diagram stabilisasi. Semakin besar jumlah data ( $N$ ), ordo ( $n$ ) dan ( $x$ ) semakin lama proses *running*-nya.

## 5.2. Saran

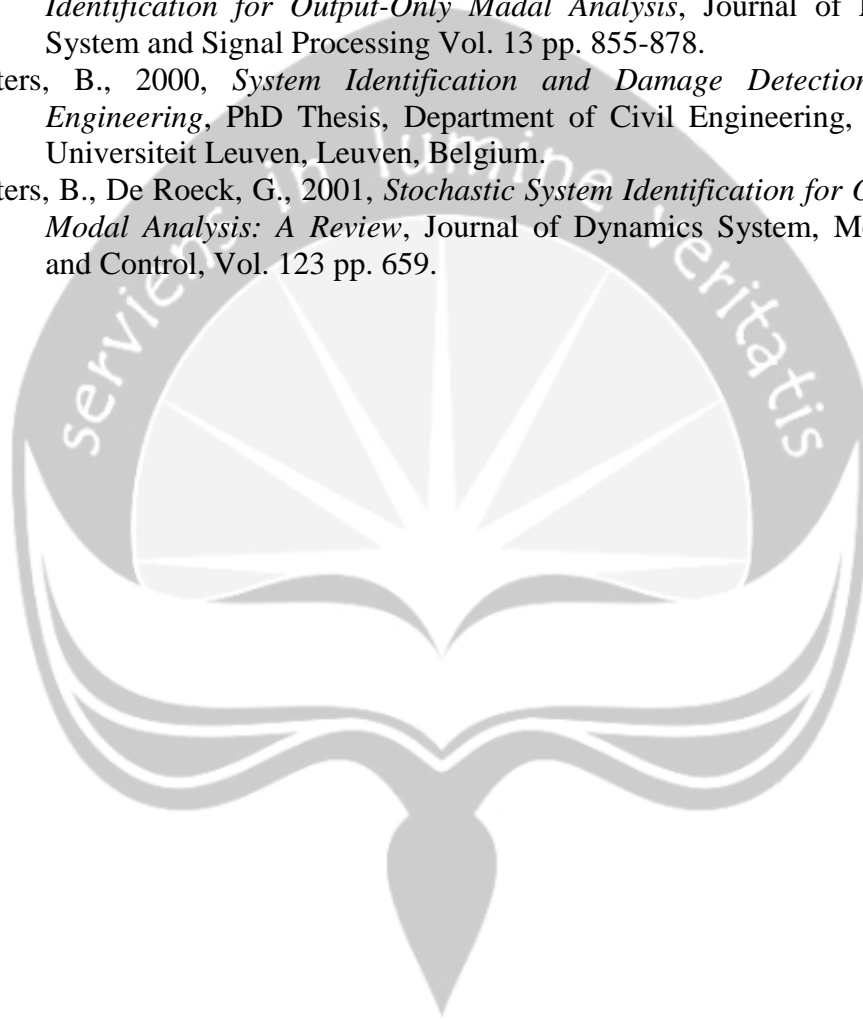
Berdasarkan pembahasan hasil dan kesimpulan dari penelitian ini maka penulis menyarankan beberapa hal, yaitu:

1. Perlu dilakukan penelitian terhadap model struktur dan material yang berbeda
2. Diperlukan metode identifikasi yang berbeda sebagai pembanding untuk membuktikan keakuratan hasil
3. Perlu dilakukan penelitian lanjutan dengan menggunakan sensor yang berbeda dan analisis menggunakan *software* selain Matlab sebagai pembanding
4. Perlu dilakukan penelitian pada struktur bangunan sipil seperti jembatan, tower, gedung atau bangunan lainnya.
5. Perlu dipelajari lebih lanjut proses pengolahan data pengukuran (*signal processing*) seperti *detrend*, *denosing alogartihm*, dan *filtering alogarithm* untuk mendapatkan hasil yang lebih akurat.

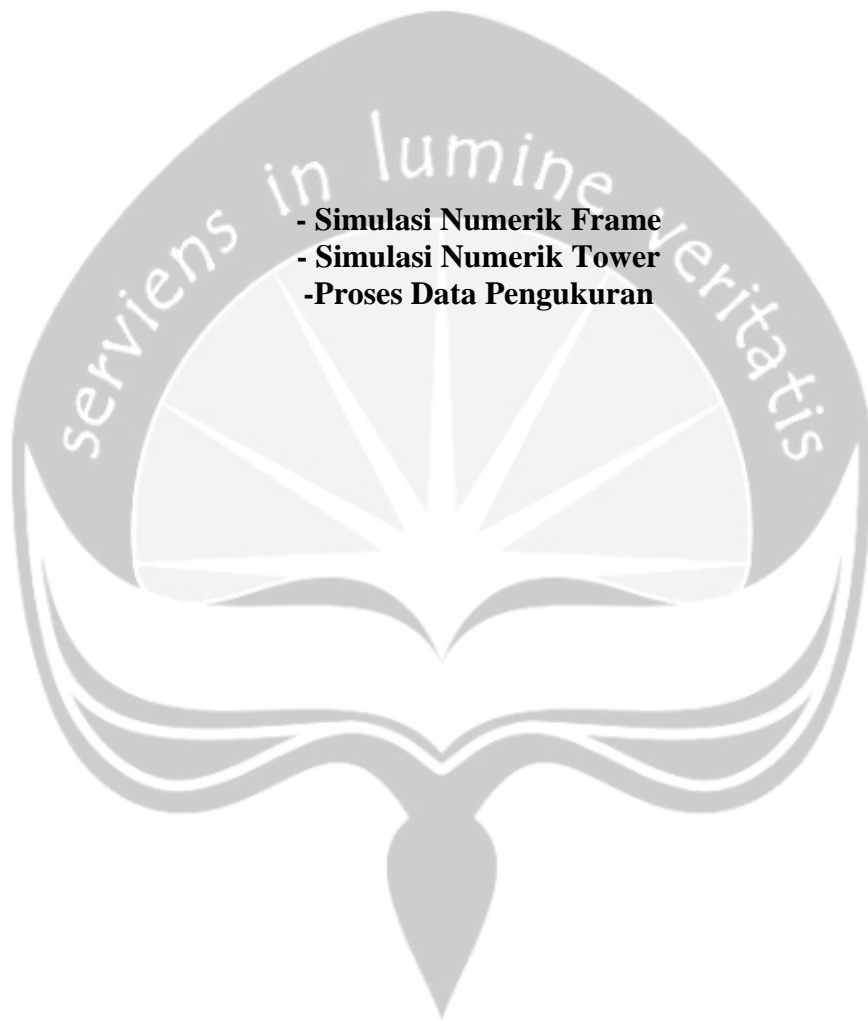
## DAFTAR PUSTAKA

- Arfiadi Y., 2016, Analisis Struktur dengan Program Matlab dan FreeMat, Cahaya Atma Pustaka, Kelompok Penerbit Universitas Atma Jaya Yogyakarta, Yogyakarta.
- Arfiadi Y., 2016, Bahan Kuliah Dinamika Struktur Lanjut, Universitas Atma Jaya Yogyakarta, Yogyakarta.
- Kurniawan A.H, Arfiadi Y, Lisantono A.M, 2017, *Identifikasi Parameter Dinamik Struktur Utuh dan Rusak Berdasarkan Operatrion Modal Analysis Sthocastic Subspcase Identification*, Tesis, Magister Teknik Sipil Universitas Atma Jaya Yogyakarta
- Rainieri, C., Fabbrocino, G., 2011, *Operational Modal Analysis for the Characterization of Heritage Structures*, Journal of Goefizika Vol. 28.
- Rainieri, C., Fabbrocino, G., 2014, *Operation Modal Analysis of Civil Engineering Structures*, Springer, 1<sup>st</sup> edition.
- Schanke, A.S., 2015, *Operational Modal Analysis of Large Bridges*, Master Thesis, Departement of Civil and Enviromental Engineering Norwegian University of science and Technology.
- Van Overshcee, P., De Moor, B., *Subspace Identification for Linear Systems: Theory – Implementation - Applications*. Kluwer Academic Publishers, 1<sup>st</sup> edition.
- Beskhyroun, S., Ma, Q., 2012, *Low-Cost Accelerometers for Experimental Modal Analysis*, Proceedings of 15<sup>th</sup> World Conference on Earthquake Engineering Lisbon, Portugal.
- Brincker, R., 2014, *Some Elements of Operational Modal Analysis*, Journal of Shock and Vibration Vol. 2014, Article ID 325839.
- Chopra, A.K., 2011, *Dynamics of Strcutres - Theory and Applications to Earthquake Engineering*, Pearson Prentice Hall, 4<sup>th</sup> edition.
- Ghalishooyan, M., Shooshtari, A., 2015, *Operational Modal Analysis Techniques and Their Theoretical and Practical Aspect: A Comprehensive Review and Introduction*, Proceedings of 6<sup>th</sup> International Operational Modal Analysis Conference Gijon, Spain.
- Golub, G., Van Loan, C., 1996, *Matrix Computations*, The Johns Hopkins University Press, 3<sup>rd</sup> edition.
- Newland, D. E., 1996, *An Introduction to Random Vibrations, Spectral & Wavelet Analysis*, Prentice Hall, 3<sup>rd</sup> edition.
- Peeters, B., De Roeck, G., Pollet, T., Schueremans, L., 1995, *Stochastic Subspace Techniques Applied to Parameter Identification of Civil Engineering Structures*, Proceedings of New Advances in Modal Synthesis of Large Structures: Nonlinear, Damped and Nondeterministic Cases, pp. 151-162, Lyon, France,
- Peeters, B., Wahab, A.M., De Roeck, G., De Visser, J., De Wilde, P.W., Ndambi, M., Vantomme, J., 1996, *Evaluation of Structural Damage by dynamic*

- system identification*, Proceedings of 21th International Seminar on Modal Analysis, pp. 1349-1361, Katholieke Universiteit Leuven, Belgium.
- Peeters, B., De Roeck, G., 1997, *The Performance of Time Domain System Identification Methods Applied to Operational Data*, Proceedings of Structural Damage Assessment Using Advanced Signal Processing Procedures, pp. 377-386, University of Sheffield, Sheffield, UK.
- Peeters, B., De Roeck, G., 1999b, *Experimental Dynamic Analysis of A Steel Mast Excited by Wind Load*, Proceedings of the 4<sup>th</sup> European Conference on Structural Dynamic, pp. 1075-1080, Prague, Czech Republic.
- Peeters, B., De Roeck, G., 1999c, *Referenced-Based Stochastic Subspace Identification for Output-Only Modal Analysis*, Journal of Mechanical System and Signal Processing Vol. 13 pp. 855-878.
- Peeters, B., 2000, *System Identification and Damage Detection in Civil Engineering*, PhD Thesis, Department of Civil Engineering, Katholieke Universiteit Leuven, Leuven, Belgium.
- Peeters, B., De Roeck, G., 2001, *Stochastic System Identification for Operational Modal Analysis: A Review*, Journal of Dynamics System, Measurement and Control, Vol. 123 pp. 659.



## Lampiran 1. Program Simulasi Numerik



### a. Algoritma Simulasi Numerik Frame \*ModelFrame.m

```
%ALOGRATIHM SIMULATION NUMERIK%
%=====
=%
% Model Frame 2 Lt : Script Analisis Numerik
% fungsi yang dibutuhkan:coor,klf,kg,memf,assff,kcon,eigval, rand
% Input: L,E,M,masse jenis(mbar)
% Output: wn,T,modeshape, acc
% Desain function (realind2D) by Arfiadi (2016)
% Modifikasi Alogarithm by Marianus J Moa 2018

clc
clear all
%coordinate system, menghasilkan koordinat titik nodal
n1=coor(0,0);
n2=coor(0.30,0);
n3=coor(0,0.4);
n4=coor(0.30,0.4);
n5=coor(0,0.8);
n6=coor(0.30,0.8);

E= 2e5 % kN/m^2
v=0.05;

Aa=2*0.003*0.018%ms
Ab=2*0.018*0.003
mbar= 0.2826 %kg/m' %7850 kg/m3*A
Ia=1/12*0.003*0.018^3
Ib=1/12*0.018*0.003^3

A1=Aa;
I1=Ia;
f1=6/5;
ri1=0.009;
rj1=0.45;
[L1,T1]=memf(n3,n4); %menghitung panjang batang dan matriks
transformasi
k1=klf(E,A1,I1,L1); % matriks kekakuan batang dalam koordinat
lokal
K1=kg(k1,T1); % Transformasi Kekakuan batang koordinat lokal ke
global
ID1=[9 1 2 9 3 4]; % id dof batang

A2=Aa;
I2=Ia;
f2=f1;
ri2=ri1;
rj2=rj1;
[L2,T2]=memf(n5,n6);
k2=klf(E,A2,I2,L2);
K2=kg(k2,T2);
ID2=[10 5 6 10 7 8];

A3=Ab;
I3=Ib;
ri3=0;
```

```

rj3=0.018;
[L3,T3]=memf(n1,n3);
k3=klf(E,A3,I3,L3);
K3=kg(k3,T3);
ID3=[ 0 0 0 9 1 2];

A4=Ab;
I4=Ib;
[L4,T4]=memf(n2,n4);
k4=klf(E,A4,I4,L4);
K4=kg(k3,T3);
ID4=[0 0 0 9 3 4];

A5=Ab;
I5=Ib;
[L5,T5]=memf(n3,n5);
k5=klf(E,A5,I5,L5);
K5=kg(k5,T5);
ID5=[9 1 2 10 5 6];

A6=Ab;
I6=Ib;
[L6,T6]=memf(n4,n6);
k6=klf(E,A6,I6,L6);
K6=kg(k6,T6);
ID6=[9 3 4 10 7 8];

dof=10;
K=assf(K1,ID1,dof);%menggabungkan matriks kekakuan global batang
K=K+assf(K2,ID2,dof); %kedalam matriks kekakuan struktur
K=K+assf(K3,ID3,dof);
K=K+assf(K4,ID4,dof);
K=K+assf(K5,ID5,dof);
K=K+assf(K6,ID6,dof);

Klat=kcon (K,8,2) % kondensasi

M=[0.007 0; 0 0.007]
[eigv,eigval]=eig(M\Klat)

[n1,n2]=size(M);
n=n1;

[wo,worder]=sort(sqrt(diag(eigval)))
mode=eigv(:,worder)

modeshape(:,1)=mode(:,1)/abs(mode(n,1))
modeshape(:,2)=mode(:,2)/abs(mode(n,2))

T1=2*pi/wo(1)

psill=0.05;
ak=2*psill/wo(1)
C=ak*Klat

eo=[-0.02;-0.02];

```



```

A= [zeros(n,n) eye(n); -inv(M)*Klat -inv(M)*C ]
E=[zeros(n,1); inv(M)*eo]

N=2*n
Cy=eye(N)
Dy=zeros(N,1)

%Random Acceleration
t=(0:0.01:10000);
a=-0.01 + (0.01+0.01)*rand(length(t),1);
%randn(length(t),1);

plot(t,a,'-.k')
acc=[t' a];
t=acc(:,1);
iul=9.81*acc(:,2);

sys1=ss(A,E,Cy,Dy);
[y1,t1,z1]=lsim(sys1,iul,t);
a=-inv(M)*[Klat C]*z1'; %-inv(M)*eo*iul';

subplot 211
plot(t,a(1,:),'-.k') %percepatan
legend('Lantai pertama')
xlabel('waktu (detik)');
ylabel('percepatan (m/s2)');
subplot 212
plot(t,a(2,:),'r-.') %percepatan
legend('lantai kedua')
xlabel('waktu (detik)');
ylabel('percepatan (m/s2)');

data=a';
data1=a(1,:);
data2=a(2,:);

test1=[data1;%zeros(size(data1,1),1)%
test2=[data2;%zeros(size(data1,1),1)%

u1max=max(abs(z1(:,1)));
u2max=max(abs(z1(:,2)));

v1max=max(abs(z1(:,3)));
v2max=max(abs(z1(:,4)));

a1max=max(abs(a(:,1)));
a2max=max(abs(a(:,2)));

save('testa.asc','test1','-ascii')
save('testb.asc','test2','-ascii')

f1=1/(2*pi()/wo(1,1))
f2=1/(2*pi()/wo(2,1))

A=a;

```

```

phi=modeshape;
zeta=[0.005 0.005];
wn=[wo(1,1) wo(2,1)];
t1=t';
save ('modelframe.mat', 'A', 'phi', 't1', 'zeta', 'wn')

```

## b. Algoritma Simulasi Numerik Tomer \*ModelTower.m

```

%ALOGRATIHM SIMULATION NUMERIK%
=====
=%
% Tower : Script Analisis Numerik
% fungsi yang dibutuhkan: coor, klf, kg, memf, assff, kcon, eigval, rand
% Input: L, E, M, massa jenis (mbar)
% Output: wn, T, modeshape, acc
% Desain function (realind2D) by Arfiadi (2016)
% Modifikasi Algoritma by Marianus J Moa 2018

clc
clear all

L=0.6

E= 2e5 % kN/m^2
v=0.02;
D=0.006 % diameter batang (m)
A= 0.25*pi*D^2 % Luas tampang batang (m2)
rho= 0.2826 %kg/m' %7850 kg/m3*A
m=rho*L; %--massa lokal
I=1/2*m*(D/2)^2

K= 3*E*I/L^3

M=[0.001]
[eigv, eigval]=eig(M\K)
[n1, n2]=size(M);
n=n1;

[wo, worder]=sort(sqrt(diag(eigval)))
mode=eigv(:, worder)
modeshape(:, 1)=mode(:, 1)/abs(mode(n, 1))

T=2*pi/wo(1)
psill=0.05;
ak=2*psill/wo(1)
C=ak*K

eo=[-0.02];
A= [zeros(n, n) eye(n); -inv(M)*K -inv(M)*C ]
E=[zeros(n, 1); inv(M)*eo]

N=2*n
Cy=eye(N)
Dy=zeros(N, 1)

```

```

%Random Acceleration
t=(0:0.01:10000);
a=-0.01 + (0.01+0.01)*rand(length(t),1);
%randn(length(t),1);
plot (t,a,'-.k')
acc=[t' a];
%

t=acc(:,1);
iul=9.81*acc(:,2);
sys1=ss(A,E,Cy,Dy);
[y1,t1,z1]=lsim(sys1,iul,t);
a=-inv(M)*[K C]*z1'; %-inv(M)*eo*iul';
%

plot (t,a(1,:), '-.k') %percepatan
xlabel('waktu (detik)');
ylabel('percepatan (m/s2)');

data=a';
data1=a(1,:)';
test1=[data1];%zeros(size (data1,1),1)%
ulmax=max(abs(z1(:,1)));
vlmax=max(abs(z1(:,2)));
almax=max(abs(a(:,1)));

save ('testa.asc','test1','-ascii')
f=1/(2*pi()/wo(1,1))

AT=a;
phi=modeshape;
zeta=[0.005];
wn=[wo(1,1)];
t1=t';
save ('modeltower.mat', 'AT', 'phi', 't1', 'zeta', 'wn')

```

### c. Proses data hasil pengukuran \*framedata.m

```

%Proses data hasil pengukuranK%
%=====

clear all;

n=1000000;
A=[1:n]';
data1=csvread('DATAFRAMELT1.CSV');
Amentah1=data1(A,2);
A1=Amentah1/2048;
Acc1=A1*9.81;
t1=data1(A,1);

data2=csvread('DATAFRAMELT2.CSV');
Amentah2=data2(A,2);

```

```

A2=Amentah2/2048;
Acc2=A2*9.81;
t2=data2(A,1);

subplot 211
plot (t1,Acc1,'-.k')
legend('Lantai pertama')
xlabel('waktu (detik)');
ylabel('percepatan (m/s2)');

subplot 212
plot (t2,Acc2,'r-.')
legend('lantai kedua')
xlabel('waktu (detik)');
ylabel('percepatan (m/s2)');
Acc=[Acc1,Acc2];
Az=Acc';
t=t1';
save ('frameexp.mat','Acc','Az','t')

```



## Lampiran 2. Program DD-SSI

- *Fungsi DDSSI.m*
- *Fungsi diagstab.m*
- *Fungsi \_solve.m*
- 



### a. Fungsi DDSSIm

```
function [f,c] = DDSSI(Y,dt,n,x)

% SSIData: Metode Data-Driven Stochastic Subspace Identification
%
%[f,c] = DDSSI(Y,dt,n,x)

% INPUTS      Y = Matriks Data dari riwayat waktu waktu
%              dt= Pembagian Waktu
%              n = ordo maksimum sistem
%              x = besaran baris blok i

% OUTPUT      f = frekuensi natural sistem
%              c = rasio redaman sistem
%
% FUNGSI YANG DIBUTUHKAN
%              diagstab.m

% Schanke 2015 dan Jefri Moa 2019

%-----Perhitungan Awal-----%
[l,N]=size(Y); %saluran dan jumlah pengukuran dari total
pengukuran
i=x*ceil(n/l); %jumlah blok baris >=n/l pers 6.4
j=N-2*i+1;    %jumlah kolom j dalam aplikasi praktik pers 6.5
%-----%

%-----Matriks Hankel dengan SVD-----%
%Matriks Hankel
H=zeros(1*2*i,j);
for k=1:2*i
    H((k-1)*l+1:k*l, :)=Y(:,k:k+j-1); %pers 6.6
end

%Faktorisasi LQ
[Q,L]=qr(H',0); %pers 6.7
L=L'; %Matriks Triangular yang lebih kecil L pers 6.8
Q=Q'; %Matriks Ortonormal Q pers 6.9

%Nilai dari Matriks L dan Q yang akan digunakan selanjutnya
L21=L(1*i+1:l*i+1,1:l*i);
L22=L(1*i+1:l*i+1,l*i+1:l*i+1);
L31=L(1*i+1+l:2*l*i,1:l*i);
L32=L(1*i+1+l:2*l*i,l*i+1:l*i+1);
Q1=Q(1:l*i,:);
Q2=Q(1*i+1:l*i+1,:);

%Proyeksi
Pi=[L21;L31]*Q1; %pers 7.0
Pim=[L31 L32]*[Q1;Q2]; %pers 7.1

%Output Sequence
Yi=[L21 L22]*[Q1;Q2]; %pers 7.3
```

```

%SVD
[U,S]=svd(Pi); %pers 7.4
s=diag(S);

clear L Q L21 L22 L31 L32 Q1 Q2 H S N Y
%-----%

%--Perhitungan Matriks State A dan Parameter Modal untuk Setiap
Ordo-----%
f=zeros(n,n); %Menentukan Efisiensi Perhitungan
c=zeros(n,n); %Menentukan Efisiensi Perhitungan
for k=1:n
    Uk=U(:,1:k); %Menentukan untuk Perhitungan selanjutnya
    ss=diag(sqrt(s(1:k))); %Menentukan untuk Perhitungan selanjutnya
    Oi=Uk*ss; %Matriks Observability Oi pers 7.6
    Si=pinv(Oi)*Pi; %Kalman Filter State Sequence pers 7.7

    Oim=Uk(1:l*(i-1),:)*ss; %Matriks Obsevability Oi dengan l baris
    terakhir dihapus
    Sip=pinv(Oim)*Pim; %Kalman State Sequence pers 7.9

    RHS=[Sip;Yi]*pinv(Si);
    A=RHS(1:k,:); %Matriks State A pers 2.98
    %Ci=RHS(k+1:k+1,:); %Ouput Matriks Pengaruh C pers 8.0

    [phi,M] = eig(A); %Solusi Eigenvalue matriks A pers 8.1
    my=diag(M); %Eigenvalue dari Matriks Diagonal M

    %mode{k}=[Ci]*phi; %Bentuk Ragam/Mode shapes pers 8.2

lambda= log(my)/dt; %Dari Waktu Diskrit ke Waktu Kontinyu pers 8.3

f(1:k,k)=abs(lambda); %frekuensi alami(rad/s) pers 8.4
%fd(1:k,k)=imag(lambda); %Frekuensi Modal Teredam (rad/s) 6.5
c(1:k,k)=-real(lambda)./abs(lambda); %rasio redaman pers 8.6
end
%-----%
%-----%
diagstab(f,c,n,1); %Diagram Stabilisasi
end

```

\*untuk run rasio redaman (c) , tukar posisi c ke f menjadi diagstab(c, f, n, 1)

## b. Fungsi diagstab.m

```
function [] = diagstab(c,f,n,l)

% diagstab: Plot diagram stabilisasi
%
% [] = diagstab(f,c,n,l)
%
% INPUTS f: frekuensi alami sistem
%         c: rasio redaman sistem
%         n: ordo maksimum sistem
%         l: untuk PLSCF: jumlah saluran pengukuran
%
% Shancke 2015 & Modifikasi by Jefri Moa 2019

%-----pemlotan data dalam diagram stabilisasi-----%
%Memilih frekuensi dan rasio redaman
f(~f) = nan;
[f,ix]=sort(f);
c(~c) = nan;
for p=1:size(c,l)
    if p<n+1
        c(:,p)=c(ix(:,p),p);
    end
end
%Pemeriksaan persyaratan stabilisasi dari frekuensi dan rasio
redaman
fu(:,1)=f(:,1); %Frekuensi tidak stabil
fs(:,1)=nan; %frekuensi stabil
fcu(:,1)=f(:,1); %rasio redaman tidak stabil
fcs(:,1)=nan; %Rasio redaman stabil
for k=2:n
    for j=1:n*1
        ch=0;
        chd=0;
        for m=1:n*1
            if ch == 0
                sf=abs(f(m,k-1)-f(j,k))/f(m,k-1); %pemeriksaan
frekuensi
                if sf < 0.001 %Batas nilai frekuensi
                    fs(j,k)=f(j,k);
                    fu(j,k)=nan;
                    ch=1;
                else
                    fu(j,k)=f(j,k);
                    fs(j,k)=nan;
                    ch=0;
                end
            end
        end
        if chd == 0
            sc=abs(c(m,k-1)-c(j,k))/c(m,k-1); %Pemeriksaan
redaman
            if sc < 0.05 %nilai batas redaman
                fcs(j,k)=f(j,k);
                fcu(j,k)=nan;
                chd=1;
            else
                fcu(j,k)=f(j,k);
            end
        end
    end
end
```



```

                fcs(j,k)=nan;
                chd=0;
            end
        end
    end
end
%Plotting
figure(1);
for k=1:n
    scatter(fs(:,k),ones(1,n*1)*k,50,'blue');
    scatter(fu(:,k),ones(1,n*1)*k,50,'red');
    scatter(fcs(:,k),ones(1,n*1)*k,50,'blue','+');
    scatter(fcu(:,k),ones(1,n*1)*k,50,'red','+');
    hold on
end
hold off

xlabel('Frekuensi Alami (rad/s)');
ylabel('Ordo');
title('Diagram Stabilisasi Frame');
%-----%
end
c.

```

### c.Fungsi Program \_solve

```

% Framesolve: Script untuk mencoba (trial) metode DD-SSI pada
sframe 2 lantai
% fungsi yang dibutuhkan:DDSSI.m
% Input: acceleration,n,x,dt
% Output: diagram stabilisasi ( frekuensi alami(f),rasio redaman
(c))
% Design By Schanke 2015 & Modifikasi by Marianus J Moa 2018
% 17 Desember 2018

clc
clear all
close all

%INPUT
%-----memuat dan menyusun data-----%
load modelframe.mat %Load data
%-----memuat dan menyusun data-----%

%Pengumpulan data dari simulasi numerik/model eksperimen ke dalam
satu matriks
Y=[A];
dt=0.01; %Pembagian waktu
size(Y) %Ukuran matriks data
% Trial DD-SSI
n=80; %Ordo maksimum sistem
x=2; %Besaran blok baris
[f,c]=DDSSI(Y,dt,n,x) %Penyelesaian dengan DD-SSI

```

```
% Plotting untuk Diagaram Stabilisasi DD-SSI  
figure(1)  
axis([0,50,0,n])
```

**\*program \_solve digunakan untuk menginput data percepatan dan trial nilai ordo maskimun (N) dan blok baris (x)**

