

## Aplikasi Berbasis *MATLAB* untuk Simulasi Fenomena *Steady State Stability*

Arief Goeritno<sup>1)</sup>, Tapip Hendrawan<sup>2)</sup>

1) Program Studi Teknik Elektro, Universitas Ibn Khaldun Bogor  
Jl. K.H. Sholeh Iskandar km.2 Kedung Badak, Tanah Sareal, Kota Bogor 16164, JB, INDONESIA  
2) PT (Persero) Telekomunikasi Indonesia (Telkom), Tbk.  
Jl. Jenderal Gatot Subroto Kav.52, Jakarta Selatan 12710  
Email: arief.goeritno@uika-bogor.ac.id

**Abstrak.** Fenomena *steady state stability* dimaknai sebagai kemampuan untuk bertahan dalam kondisi sinkronisasi antar mesin dalam system, setelah terjadi gangguan kecil berupa perubahan beban. Keberadaan fenomena tersebut dapat dilakukan melalui simulasi berbasis aplikasi komputer, seperti *MATLAB*. Aplikasi berbasis *MATLAB* buntut keperluan simulasi berupa struktur sintaks program untuk peng-input-an data nilai-nilai asumsi, aliran daya, dan *steady state stability*. Sintaks program peng-input-an data nilai-nilai asumsi berisi data bus, generator, dan impedans saluran antar bus. Sintaks program untuk simulasi fenomena aliran daya digunakan untuk penentuan daya mekanis setiap mesin pembangkitan daya akibat perubahan beban. Sintaks program untuk fenomena *steady state stability* berisi nilai daya mekanis dari hasil penghitungan program aliran daya, data tiap-tiap mesin, dan persamaan-persamaan untuk pencarian sejumlah parameter. Mesin pembangkit dengan pembangkitan daya terbesar butuh waktu lebih lama untuk pencapaian kondisi kestabilan kembali setelah terdapat perubahan beban.

**Kata kunci:** aplikasi berbasis *MATLAB*, simulasi di sistem daya listrik, fenomena *steady state stability*.

### 1. Pendahuluan

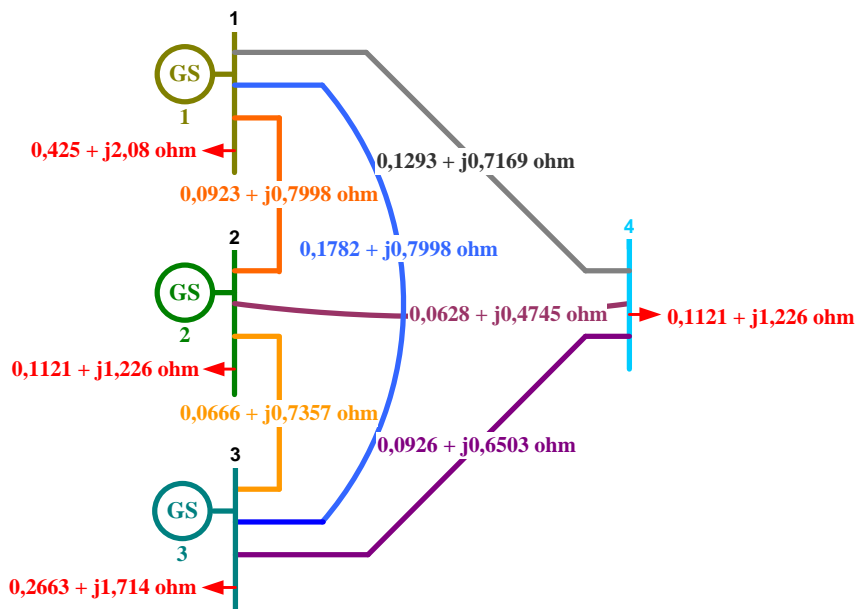
*MATLAB* merupakan sistem interaktif dan sebuah aplikasi dengan bahasa pemrograman. Elemen data dasar merupakan sebuah matriks yang tidak perlu pendeklarasian ukuran atau jenis data, sehingga banyak masalah perhitungan dapat diselesaikan pada waktu singkat dan perhitungan diambil untuk dituliskan ke dalam bahasa *FORTRAN* atau C [1,2]. Penyelesaian untuk kasus perhitungan yang kompleks, seperti studi aliran daya, perintah yang telah dimasukkan pada *command windows* sulit untuk dilakukan pengubahan terhadap satu nilai atau beberapa variabel atau perintah-perintah guna pengevaluasian kembali. Cara pemecahan kasus yang kompleks dengan mudah dan efisien terhadap problematika pada *command windows MATLAB*, dilakukan melalui pemberian masukan ke dalam sebuah *text file* yang biasa disebut *script files* atau *M-files* [1]. Untuk pembuatan sebuah *script file* dengan pemilihan *New M-file* dari menu *File*, sebuah jendela *text editor* ditampilkan [3].

Stabilitas (kestabilan) pada sistem daya listrik (*electrical power system, EPS*) telah diakui sebagai problematika untuk pengoperasian sistem daya yang aman sejak dekade 1920-an [4], setelah penggunaan listrik secara komersial pada akhir 1870-an dengan pemasangan lampu-lampu untuk penerangan mercusuar (*lighthouse*) dan penerangan jalan [5]. Stabilitas dimaknai sebagai kemampuan sistem daya listrik untuk kembali ke kondisi normal atau stabil setelah terganggu [6-8]. Stabilitas sistem terutama tergantung pada perilaku mesin sinkron setelah gangguan [9,6], karena batas-batas stabilitas sistem ditandai oleh keberadaan sejumlah daya maksimum teralirkan melalui suatu titik dalam sistem tanpa berakibat kehilangan stabilitas [9,10]. Kehilangan kestabilan pada sistem daya listrik sangat dipengaruhi oleh keberadaan sejumlah gangguan [11], sehingga keberlangsungan sistem daya listrik harus dipertahankan pada kondisi sinkronisasi sesempurna mungkin dalam keadaan tunak (*steady state*) [4,10]. Gangguan pada sistem dapat berupa berbagai jenis [11-13], seperti perubahan beban yang tiba-tiba, hubung singkat yang tiba-tiba antara saluran dan tanah [14], gangguan antar saluran, gangguan simetris saluran ke tanah, fenomena peralihan, dan lain-lain.

*Steady state stability* sangat penting untuk (i) perencanaan dan perancangan sistem daya, (ii) pengembangan perangkat kontrol otomatis khusus, (iii) penempatan elemen baru ke dalam operasi sistem, atau (iv) modifikasi kondisi pengoperasian [15]. Pemilihan stabilitas dibuat berdasarkan persyaratan batas stabilitas atau kualitas daya listrik dalam *steady state* atau selama transien. Batas

*steady state* berpedoman kepada aliran daya maksimum melalui titik tertentu tanpa berakibat kehilangan stabilitas, ketika daya dinaikkan atau diturunkan secara bertahap [16,17]. Estimasi batas *steady state* penting untuk analisis sistem daya listrik dengan cakupan pada pengecekan sistem daya listrik dalam kondisi stabil, penentuan batas stabilitas, dan estimasi kualitatif kondisi transien [9,15]. Keberadaan stabilitas dan estimasi batas pada *steady state* diperlukan untuk perkiraan pilihan jenis sistem eksitasi dan pengontrol, mode pengontrolan, parameter sistem kontrol eksitasi, dan otomatisasi, termasuk kinerja pemutus tenaga (PMT) [18]. Hal itu berkaitan dengan keberadaan sistem daya listrik yang terdiri atas beberapa mesin sinkron yang beroperasi dalam sinkronisasi [9,19]. Untuk kondisi dimana gangguan terjadi di sistem, sehingga dalam sistem terjadi pengembangan gaya untuk kembali ke keadaan normal atau stabil [20,6,10].

Berdasarkan kepada kemampuan sistem untuk perolehan kembali sinkronisasi frekuensi di semua jaringan setelah perubahan daya secara bertahap, maka perumusan masalah ditetapkan dengan pelaksanaan simulasi fenomena *steady state stability*. Simulasi tersebut dilakukan melalui penentuan analogi struktur jaringan daya listrik dengan empat *bus* yang saling terhubung dengan nilai asumsi terhadap sejumlah parameter, kemudian dilanjutkan dengan pembuatan algoritma dan sintaks berbasis *MATLAB*. Analogi struktur sistem daya listrik untuk simulasi *steady state stability* [21], seperti ditunjukkan pada Gambar 1.



Gambar 1. Analogi struktur sistem daya listrik untuk simulasi *steady state stability*

Berdasarkan Gambar 1 ditunjukkan, bahwa beban dan saluran transmisi penghubung dalam bentuk nilai impedans, yaitu nilai  $R + jX$  (ohm). Tanda lingkaran dengan notasi "GS" merupakan simbol generator sinkron pada sistem pembangkitan yang berputar, sedangkan sistem dengan 4 bus, yaitu bus-1, bus-2, bus-3, dan bus-4.

Tipe bus untuk simulasi terdiri atas (a) bus ayun (*bus penadah, slack bus, reference bus, swing bus*), (b) bus pembangkitan (*generating bus, voltage control bus*), dan (c) bus beban (*load bus*). Bus ayun ( $V - \phi$  bus) dinotasikan terhadap bus-1 yang berfungsi sebagai penanggung kekurangan daya pembangkitan setelah solusi aliran daya diperoleh. Bus pembangkitan ( $P - V$ ) dinotasikan terhadap bus-2 dan bus-3 yang berfungsi sebagai bus dengan parameter injeksi daya aktif dan besar tegangan bus diketahui. Bus beban ( $P - Q$  bus) untuk bus-4. Berfungsi sebagai bus dengan parameter injeksi daya aktif dan daya reaktif bus diketahui [22]. Penjabaran lebih lanjut berdasarkan rumusan masalah, maka ditetapkan tujuan penelitian yang berkaitan dengan perolehan struktur sintaks program berbasis *MATLAB* dan hasil pelaksanaan simulasi. Sasaran penelitian, meliputi (1) perolehan struktur sintaks program peng-input-an data nilai-nilai asumsi, (2) struktur sintaks program untuk simulasi fenomena aliran daya, (3) struktur sintaks program untuk simulasi fenomena *steady state stability* dengan data

nilai-nilai asumsi, dan (4) hasil pelaksanaan simulasi *steady state stability* dengan data nilai-nilai asumsi.

Parameter-parameter dalam permodelan matematis diberi nilai-nilai asumsi untuk pensimulasian parameter-parameter *steady state stability* pada analogi sistem daya listrik empat bus tiga pusat pembangkitan dengan pengubahan variabel nilai daya beban (*demand*) dan pengaruhnya terhadap masing-masing mesin. Berpedoman kepada Gambar 1, diketahui nilai-nilai asumsi awal pada permodelan sistem daya listrik empat bus dengan tiga pusat pembangkitan, berupa nilai saluran transmisi dengan basis pada 100 MVA, sehingga diperoleh satuan *per unit*. Nilai komponen saluran transmisi penghubung, seperti ditunjukkan pada Tabel 1.

Tabel 1. Nilai komponen saluran transmisi penghubung

Dari bus	Ke bus	$R$ (p.u.)	$X$ (p.u.)
1	2	9,23	79,98
1	3	12,93	71,69
1	4	17,82	79,98
2	3	6,28	47,45
2	4	6,66	73,57
3	4	9,26	65,03

Nilai daya beban (*demand*) pada tiap bus, seperti ditunjukkan pada Tabel 2.

Tabel 2. Nilai daya beban (*demand*) pada tiap bus

Bus	$P_d$ (MW)	$Q_d$ (MVar)	$V_{maks.}$ (p.u.)	$V_{min.}$ (p.u.)
1	42,50	203,8	1,1	0,9
2	11,21	122,6	1,1	0,9
3	26,63	171,4	1,1	0,9
4	42,18	147,5	1,1	0,9

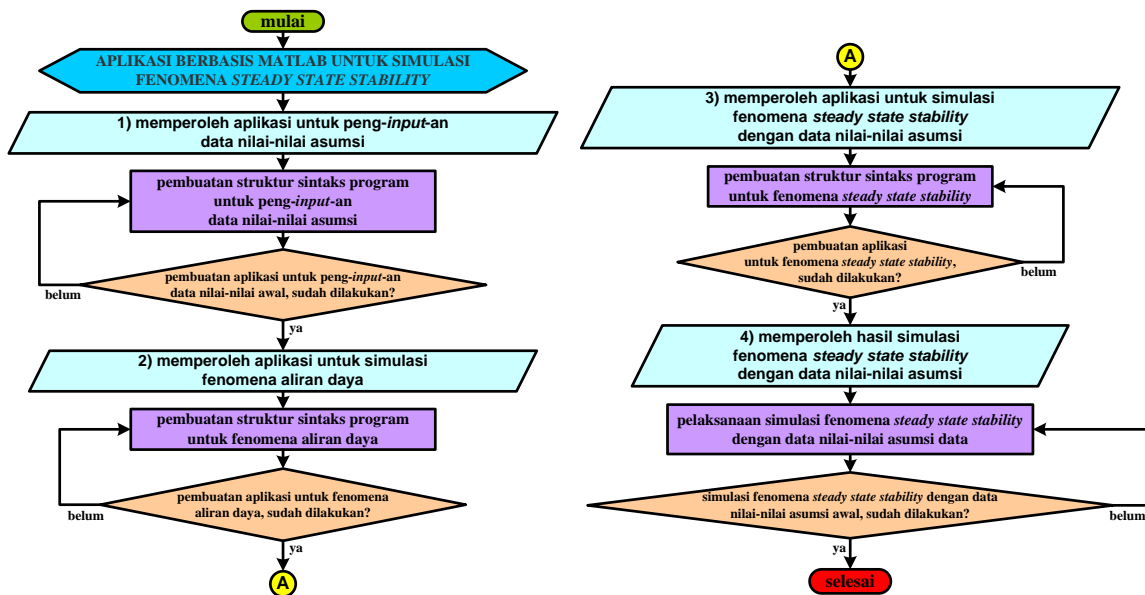
Nilai keadaan awal masing-masing generator sistem pembangkitan sebelum proses perhitungan aliran daya, seperti ditunjukkan pada Tabel 3.

Tabel 3. Nilai keadaan awal masing-masing generator sistem pembangkitan sebelum proses perhitungan aliran daya

Bus	$P_m$ (MW)	$Q_m$ (MVar)	$V$ (p.u.)	$\delta$ (derajat listrik)	$H$ (MJA/MVA)	$f_0$ (Hz.)
1	0	0	1,03	0	2,32	60
2	45	0	1,025	0	3,4	60
3	50	0	1,03	0	4,63	60

Berdasarkan Tabel 3 telah ditetapkan terhadap bus ke-1 sebagai bus ayun, maka nilai pembangkitan disesuaikan setelah sistem beroperasi.

Pelaksanaan simulasi terhadap fenomena *steady state stability* pada sistem daya listrik dengan penetapan nilai-nilai asumsi untuk proses simulasi, algoritma dan sintaks pada pembuatan struktur program digunakan aplikasi *MATLAB*, dan pelaksanaan simulasi melalui eksekusi terhadap program. Diagram alir pelaksanaan simulasi terhadap keberadaan fenomena *steady state stability* pada sistem daya listrik berbantuan aplikasi *MATLAB*, seperti ditunjukkan pada Gambar 2.



Gambar 2. Diagram alir pelaksanaan simulasi terhadap keberadaan fenomena *steady state stability* pada sistem daya listrik berbantuan aplikasi *MATLAB*

## 2. Pembahasan

### 2.1. Struktur Sintaks Program untuk Peng-input-an Data Nilai-nilai Asumsi

Untuk peng-input-an (pemasukan) data nilai-nilai asumsi diperlukan keberadaan struktur sintaks berbasis aplikasi *MATLAB*. Struktur sintaks program untuk peng-input-an data nilai-nilai asumsi, yaitu:

```

%Listing Program Matlab 5.0
%FOR MULTI-MACHINE SYSTEM
clc
disp(' EKSEKUSI PROGRAM ')
disp('*****')
disp('*****APLIKASI BERBASIS MATLAB*****')
disp('*UNTUK SIMULASI FENOMENA STEADY STATE STABILITY*')
disp('*****ARIEF GOERITNO, TAPIP HENDRAWAN*****')
disp('*****PRODI TEKNIK ELEKTRO, UIKA BOGOR*****')
disp('*****')
function [baseMVA, bus, gen, branch] = case3gs
% build function for data input of bus system
%%----- Data for Power Flow Iteration -----%%
%% system MVA base
baseMVA = 100;
%% bus data
% bus_i type   Pd Qd Gs Bs area   Vm Va baseKV zone   Vmax   Vmin
bus = [
    1   3   42.50       203.8  0   0   1   1   0   230 1   1.1 0.9;
    2   2   11.21       122.6  0   0   1   1   0   230 1   1.1 0.9;
    3   2   26.63       171.4  0   0   1   1   0   230 1   1.1 0.9;
    4   1   42.18       147.5  0   0   1   1   0   230 1   1.1 0.9;
];
%% generator data
%   bus   Pg   Qg   Qmax   Qmin   Vg   mBase   status   Pmax   Pmin
gen = [
    2  45   0   100 -100   1.025  100 1       0       0;
    3  50   0   100 -100   1.03   100 1       0       0;
    1   0   0   100 -100   1.03   100 1       0       0;
];
%% branch data
% fbus   tbus   r   x   b   rateA   rateB   rateC   ratio   angle   status
branch = [
    1   2   0.0923  0.7998  0.1025  250 250 250 0   0   1;
    1   4   0.1293  0.7169  0.0775  250 250 250 0   0   1;
    1   3   0.1782  0.7998  0.0775  250 250 250 0   0   1;
    2   4   0.0628  0.4745  0.0775  250 250 250 0   0   1;
    2   3   0.0666  0.7357  0.0775  250 250 250 0   0   1;
    3   4   0.0926  0.6503  0.0775  250 250 250 0   0   1;
];
  
```

```
1;  
return;
```

Berdasarkan uraian tersebut ditunjukkan, bahwa struktur sintaks program terdiri atas (i) data *bus* yang meliputi nomor *bus*, tipe *bus*, dan data daya beban (*demand*), (ii) data generator yang meliputi daya pembangkit, tegangan, dan data basis, dan (iii) data impedans saluran antar *bus*.

## 2.2. Struktur Sintaks Program untuk Simulasi Fenomena Aliran Daya

Struktur sintaks program untuk simulasi fenomena aliran daya, yaitu:

```
%Listing Program Matlab 5.0  
%FOR MULTI-MACHINE SYSTEM  
clc  
disp(' EKSEKUSI PROGRAM ')  
disp('*****APLIKASI BERBASIS MATLAB*****')  
disp('*****UNTUK SIMULASI FENOMENA STEADY STATE STABILITY*****')  
disp('*****ARIEF GOERITNO, TAPIP HENDRAWAN*****')  
disp('*****PRODI TEKNIK ELEKTRO, UIKA BOGOR*****')  
disp('*****')  
function [MVAbase, bus, gen, branch, success, et] = ...  
    runpf(casename, mpopt, fname, solvedcase)  
    [PQ, PV, REF, NONE, BUS_I, BUS_TYPE, PD, QD, GS, BS, BUS_AREA, VM, ...  
     VA, BASE_KV, ZONE, VMAX, VMIN, LAM_P, LAM_Q, MU_VMAX, MU_VMIN] = idx_bus;  
    [F_BUS, T_BUS, BR_R, BR_X, BR_B, RATE_A, RATE_B, ...  
     RATE_C, TAP, SHIFT, BR_STATUS, PF, QF, PT, QT, MU_SF, MU_ST] = idx_brch;  
    [GEN_BUS, PG, QG, QMAX, QMIN, VG, MBASE, ...  
     GEN_STATUS, PMAX, PMIN, MU_PMAX, MU_PMIN, MU_QMAX, MU_QMIN] = idx_gen;  
    %% default arguments  
    if nargin < 4  
        solvedcase = '';      %% don't save solved case  
        if nargin < 3  
            fname = '';      %% don't print results to a file  
            if nargin < 2  
                mpopt = mpopt;      %% use default options  
                if nargin < 1  
                    casename = 'case9'; %% default data file is 'case9.m'  
                end  
            end  
        end  
    end  
    end  
    %% options  
    verbose = mpopt(31);  
    qlim = mpopt(6);          %% enforce Q limits on gens?  
    dc = mpopt(10);          %% use DC formulation?  
    %% read data & convert to internal bus numbering  
    [baseMVA, bus, gen, branch] = loadcase(casename);  
    [i2e, bus, gen, branch] = ext2int(bus, gen, branch);  
    %% get bus index lists of each type of bus  
    [ref, pv, pq] = bustypes(bus, gen);  
    %% generator info  
    on = find(gen(:, GEN_STATUS) > 0);      %% which generators are on?  
    gbus = gen(on, GEN_BUS);                %% what buses are they at?  
    %%----- run the power flow -----  
    t0 = clock;  
  
    if dc    %% DC formulation  
        %% initial state  
        Va0 = bus(:, VA) * (pi/180);  
        %% build B matrices and phase shift injections  
        [B, Bf, Pbusinj, Pfinj] = makeBdc(baseMVA, bus, branch);  
        %% compute complex bus power injections (generation - load)  
        %% adjusted for phase shifters and real shunts  
        Pbus = real(makeSbus(baseMVA, bus, gen)) - Pbusinj - bus(:, GS) / baseMVA;  
        %% "run" the power flow  
        Va = dcpf(B, Pbus, Va0, ref, pv, pq);  
        %% update data matrices with solution  
        branch(:, [QF, QT]) = zeros(size(branch, 1), 2);  
        branch(:, PF) = (Bf * Va + Pfinj) * baseMVA;  
        branch(:, PT) = -branch(:, PF);  
        bus(:, VM) = ones(size(bus, 1), 1);  
        bus(:, VA) = Va * (180/pi);  
        %% update Pg for swing generator (note: other gens at ref bus are accounted for in Pbus)  
        %%      Pg = Pinj + Pload + Gs
```

```

%%      newPg = oldPg + newPinj - oldPinj
refgen = find(gbus == ref);          %% which is(are) the reference gen(s)?
gen(on(refgen(1)), PG) = gen(on(refgen(1)), PG) + (B(ref, :) * Va - Pbus(ref)) * baseMVA;
    success = 1;
else
    %% AC formulation
    %% initial state
    % V0 = ones(size(bus, 1), 1);      %% flat start
    V0 = bus(:, VM) .* exp(sqrt(-1) * pi/180 * bus(:, VA));
    V0(gbus) = gen(on, VG) ./ abs(V0(gbus)) .* V0(gbus);
    if qlim
        ref0 = ref;                    %% save index and angle of
        Varef0 = bus(ref0, VA);        %% original reference bus
        limited = [];                  %% list of indices of gens @ Q lims
        fixedQg = zeros(size(gen, 1), 1); %% Qg of gens at Q limits
    end
repeat = 1;
while (repeat)
    %% build admittance matrices
    [Ybus, Yf, Yt] = makeYbus(baseMVA, bus, branch);
    %% compute complex bus power injections (generation - load)
    Sbus = makeSbus(baseMVA, bus, gen);
    %% run the power flow
    alg = mpopt(1);
    if alg == 1
        [V, success, iterations] = newtonpf(Ybus, Sbus, V0, ref, pv, pq, mpopt);
    elseif alg == 2 | alg == 3
        [Bp, Bpp] = makeB(baseMVA, bus, branch, alg);
        [V, success, iterations] = fdcpf(Ybus, Sbus, V0, Bp, Bpp, ref, pv, pq, mpopt);
    elseif alg == 4
        [V, success, iterations] = gausspf(Ybus, Sbus, V0, ref, pv, pq, mpopt);
    else
        error('Only Newton''s method, fast-decoupled, and Gauss-Seidel power flow algorithms
currently implemented.');
```

```

    end
    %% update data matrices with solution
    [bus, gen, branch] = pfsoln(baseMVA, bus, gen, branch, Ybus, Yf, Yt, V, ref, pv, pq);
    if qlim
        %% enforce generator Q limits
        %% find gens with violated Q constraints
        mx = find( gen(:, GEN_STATUS) > 0 & gen(:, QG) > gen(:, QMAX) );
        mn = find( gen(:, GEN_STATUS) > 0 & gen(:, QG) < gen(:, QMIN) );
        if ~isempty(mx) | ~isempty(mn) %% we have some Q limit violations
            if verbose & ~isempty(mx)
                fprintf('Gen %d at upper Q limit, converting to PQ bus\n', mx);
            end
            if verbose & ~isempty(mn)
                fprintf('Gen %d at lower Q limit, converting to PQ bus\n', mn);
            end
        end
        %% save corresponding limit values
        fixedQg(mx) = gen(mx, QMAX);
        fixedQg(mn) = gen(mn, QMIN);
        mx = [mx;mn];
        %% convert to PQ bus
        gen(mx, QG) = fixedQg(mx); %% set Qg to binding limit
        gen(mx, GEN_STATUS) = 0;    %% temporarily turn off gen,
        for i = 1:length(mx)
            %% (one at a time, since
            bi = gen(mx(i), GEN_BUS); %% they may be at same bus)
            bus(bi, [PD,QD]) = ...    %% adjust load accordingly,
            bus(bi, [PD,QD]) - gen(mx(i), [PG,QG]);
        end
        bus(gen(mx, GEN_BUS), BUS_TYPE) = PQ; %% & set bus type to PQ
    %% update bus index lists of each type of bus
    ref_temp = ref;
    [ref, pv, pq] = bustypes(bus, gen);
    if verbose & ref ~= ref_temp
        fprintf('Bus %d is new slack bus\n', ref);
    end
    limited = [limited; mx];
    else
        repeat = 0; %% no more generator Q limits violated
    end
    else
        repeat = 0; %% don't enforce generator Q limits, once is enough
    end
end
end
if qlim & ~isempty(limited)
    %% restore injections from limited gens (those at Q limits)

```

```

        gen(limited, QG) = fixedQg(limited);    %% restore Qg value,
        for i = 1:length(limited)              %% (one at a time, since
            bi = gen(limited(i), GEN_BUS);      %% they may be at same bus)
            bus(bi, [PD,QD]) = ...              %% re-adjust load,
                bus(bi, [PD,QD]) + gen(limited(i), [PG,QG]);
        end

    gen(limited, GEN_STATUS) = 1;              %% and turn gen back on
    if ref ~= ref0
        %% adjust voltage angles to make original ref bus correct
        bus(:, VA) = bus(:, VA) - bus(ref0, VA) + Varef0;
    end
end
end
et = etime(clock, t0);
%%----- output results -----
%% convert back to original bus numbering & print results
[bus, gen, branch] = int2ext(i2e, bus, gen, branch);
if fname
    [fd, msg] = fopen(fname, 'at');
    if fd == -1
        error(msg);
    else
        fprintf(baseMVA, bus, gen, branch, [], success, et, fd, mpopt);
        fclose(fd);
    end
end
fprintf(baseMVA, bus, gen, branch, [], success, et, 1, mpopt);

%% save solved case
if solvedcase
    savecase(solvedcase, baseMVA, bus, gen, branch);
end
%% this is just to prevent it from printing baseMVA
%% when called with no output arguments
if nargin, MVAbase = baseMVA; end
return;

```

Berdasarkan uraian tersebut ditunjukkan, bahwa struktur sintaks program untuk simulasi fenomena aliran daya digunakan untuk penentuan daya mekanis setiap pembangkit akibat perubahan beban, dengan pengoperasian melalui masukan pada program peng-*input*-an data nilai-nilai asumsi awal.

Nilai daya mekanis awal hasil penghitungan dengan struktur sintaks program untuk fenomena aliran daya, seperti ditunjukkan pada Tabel 4.

Tabel 4. Nilai daya mekanis awal hasil perhitungan dengan struktur sintaks program untuk fenomena aliran daya

Bus	Tegangan		Daya dari Generator		Daya ke Beban	
	Mag. (p.u.)	Ang. (deg.)	P (MW)	Q (MVA <sub>r</sub> )	P (MW)	Q (MVA <sub>r</sub> )
1	1,030	0,00	35,43	258,06	42,50	203,8
2	1,025	5,464	45,00	176,20	11,21	122,6
3	1,030	3,912	50,00	239,05	26,63	171,4
4	0,514	-0,723	-	-	42,18	147,5
Total:			130,43	673,31	122,52	645,30

Berdasarkan Tabel 4 ditunjukkan, bahwa nilai daya mekanis awal yang di-*input*-kan ke struktur sintaks program untuk fenomena *steady state stability* sebesar 35,43 MW untuk mesin-1, 45 MW untuk mesin-2, dan 50 MW untuk mesin-3.

## 2.3. Struktur Sintaks Program untuk Simulasi Fenomena *Steady State Stability*

Struktur sintaks program untuk fenomena *steady state stability*, yaitu:

```

%Listing Program Matlab 5.0
%FOR MULTI-MACHINE SYSTEM
clc
disp(' EKSEKUSI PROGRAM ')
disp('*****')
disp('*****APLIKASI BERBASIS MATLAB*****')

```



```

disp('*UNTUK SIMULASI FENOMENA STEADY STATE STABILITY*')
disp('*****ARIEF GOERITNO, TAPIP HENDRAWAN*****')
disp('*****PRODI TEKNIK ELEKTRO, UIKA BOGOR*****')
disp('*****')
disp('* diketahui data sebagai berikut:          *')
disp('* S1=35,43+j258,3; S2=45,00+j176,20; S3=50,00-j239,05 *')
disp('*****')
%disp('*****')
% Data input
% Vbus H X D Fo
V1=1.030; H1=2.32; X1=0.32; D1=0.4; f01=60; % Mesin 1
V2=1.025; H2=3.40; X2=0.33; D2=0.4; f02=60; % Mesin 2
V3=1.030; H3=4.63; X3=0.20; D3=0.4; f03=60; % Mesin 3
% Hasil perhitungan daya mekanis setelah estimasi aliran daya
S1=input ('Masukkan nilai Daya Mekanis Mesin 1 = ')/100;
S2=input ('Masukkan nilai Daya Mekanis Mesin 2 = ')/100;
S3=input ('Masukkan nilai Daya Mekanis Mesin 3 = ')/100;
% nilai real dari daya mekanis
Pm1=abs(S1);
Pm2=abs(S2);
Pm3=abs(S3);
% Perhitungan Arus Listrik yang di hasilkan oleh masing-masing mesin
I1=conj(S1)/conj(V1) % pada mesin 1
I2=conj(S2)/conj(V2) % pada mesin 2
I3=conj(S2)/conj(V2) % pada mesin 3
% Perhitungan tegangan keluaran dari masing-masing mesin
E1=V1+I1*X1 % Perhitungan tegangan pada Mesin 1
E2=V2+I2*X2 % Perhitungan tegangan pada Mesin 2
E3=V3+I3*X3 % Perhitungan tegangan pada Mesin 3
% Perhitungan perubahan putaran, dan frekuensi pada mesin 1
disp(' Hasil Eksekusi')
Pmax1=abs(E1)*V1/X1, d01=asin(Pm1/Pmax1) %Daya maksimum
Ps1=Pmax1*cos(d01) %Koefisien Daya Sinkronisasi
wn1=sqrt(pi*60/H1*Ps1) %Undamped frequency of oscillation
z1=D1/2*sqrt(pi*60/(H1*Ps1)) %Damping ratio
wd1=wn1*sqrt(1-z1^2), fd1=wd1/(2*pi) %Damped frequency oscillation
tau1=1/(z1*wn1) %Waktu tetap
th1=acos(z1) %Sudut fase Tetap
Dd01=10*pi/180 %Initial angle in radian
t1=0:0.01:1; % time interval
Dd1=Dd01/sqrt(1-z1^2)*exp(-z1*wn1*t1).*sin(wd1*t1+th1); % perubahan sudut rotor
dl=(d01+Dd1)*180/pi ; %Power angle in degree
Dw1=-wn1*Dd01/sqrt(1-z1^2)*exp(-z1*wn1*t1).*sin(wd1*t1); % perubahan kecepatan putar sudut
rotor
f1=f01+Dw1/(2*pi); %Frekuensi (Hz)
% Perhitungan perubahan putaran, dan frekuensi pada mesin 2
Pmax2=abs(E2)*V2/X2, d02=asin(Pm2/Pmax2) %Daya maksimum
Ps2=Pmax2*cos(d02) %Koefisien Daya Sinkron
wn2=sqrt(pi*60/H2*Ps2) %Undamped frequency of oscillation
z2=D2/2*sqrt(pi*60/(H2*Ps2)) %Damping ratio
wd2=wn2*sqrt(1-z2^2), fd2=wd2/(2*pi) %Damped frequency oscill
tau2=1/(z2*wn2) %Waktu tetap
th2=acos(z2) %Sudut fase Teta
Dd02=10*pi/180 %Initial angle in radian
Dd2=Dd02/sqrt(1-z2^2)*exp(-z2*wn2*t1).*sin(wd2*t1+th2); % perubahan sudut rotor
d2=(d02+Dd2)*180/pi ; %Power angle in degree
Dw2=-wn2*Dd02/sqrt(1-z2^2)*exp(-z2*wn2*t1).*sin(wd2*t1); %perubahan kecepatan putar sudut rotor
f2=f02+Dw2/(2*pi); %Frekuensi (Hz)
% Perhitungan perubahan putaran, dan frekuensi pada mesin 3
Pmax3=abs(E3)*V3/X3, d03=asin(Pm3/Pmax3) %Daya maksimum
Ps3=Pmax3*cos(d03) %Koefisien Daya Sinkron
wn3=sqrt(pi*60/H3*Ps3) %Undamped frequency of oscillation
z3=D3/2*sqrt(pi*60/(H3*Ps3)) %Damping ratio
wd3=wn3*sqrt(1-z3^2), fd3=wd3/(2*pi) %Damped frequency oscill
tau3=1/(z3*wn3) %Waktu tetap
th3=acos(z3) %Sudut fase Teta
Dd03=10*pi/180 %Initial angle in radian
Dd3=Dd03/sqrt(1-z3^2)*exp(-z3*wn3*t1).*sin(wd3*t1+th3);
d3=(d03+Dd3)*180/pi ; %Power angle in degree
Dw3=-wn3*Dd03/sqrt(1-z3^2)*exp(-z3*wn3*t1).*sin(wd3*t1);
f3=f03+Dw3/(2*pi); %Frekuensi (Hz)
% pencetakan grafik perubahan sudut, perubahan kecepatan sudut dan perubahan frekuensi
subplot(4,1,2), plot(t1,Dd1,t1,Dd2,t1,Dd3), grid % Grafik Perubahan sudut rotor mesin 1
xlabel('t (sec)'), ylabel('Delta Rotor Angle`in (degree)')
subplot(4,1,4), plot(t1,f1,t1,f2,t1,f3), grid % Grafik perubahan sudut rotor mesin 2
xlabel('t (sec)'), ylabel('Frekuensi in (Hz)')
subplot(4,1,3), plot(t1,Dw1,t1,Dw2,t1,Dw3), grid % Grafik perubahan sudut rotor mesin 3

```

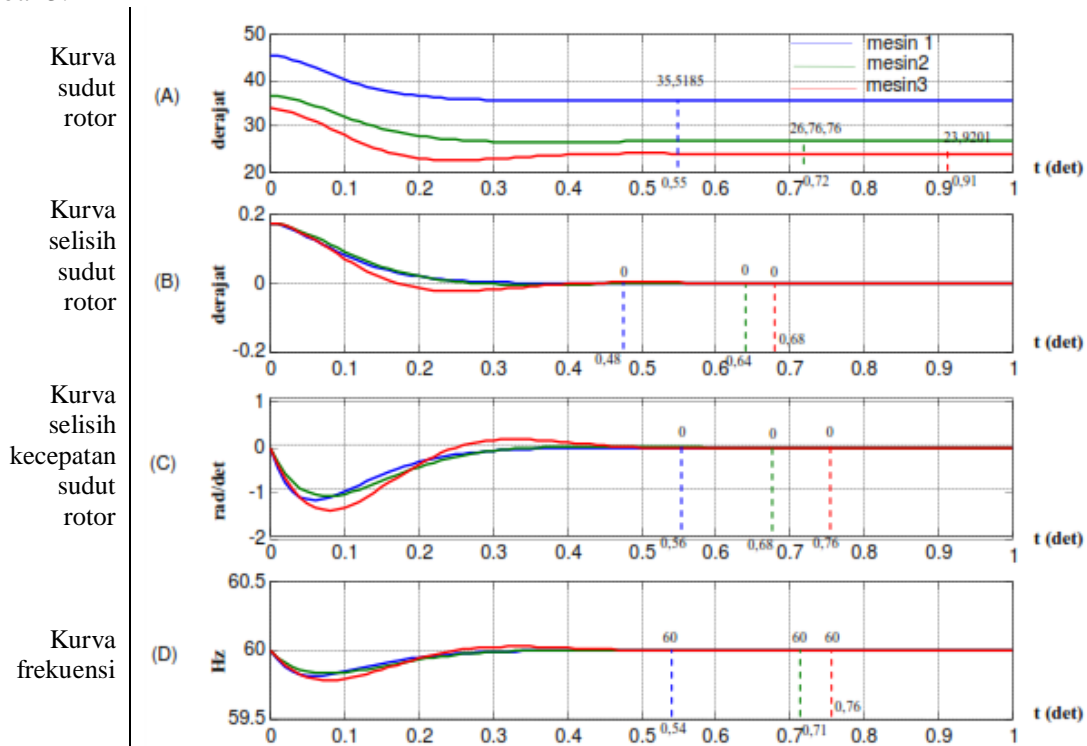


```
xlabel('t (sec)'), ylabel('Delta Omega in (rad/sec)')
subplot(4,1,1), plot(t1,d1,t1,d2,t1,d3), grid % Grafik perubahan sudut rotor mesin 3
xlabel('t (sec)'), ylabel('Rotor Angle in (degree)')
subplot(111)
disp('*****Grafik 1*****')
disp('')
disp('***** END *****')
```

Berdasarkan uraian tersebut ditunjukkan, bahwa struktur sintaks program untuk fenomena *steady state stability* terdiri atas (i) nilai daya mekanis sebagai data masukan yang diperoleh dari hasil penghitungan program aliran daya; (ii) data tiap mesin berupa nilai tegangan, momen inerti, rasio *damping*, frekuensi, dan resistans dalam mesin; dan (iii) persamaan-persamaan untuk pencarian nilai arus, tegangan, fekuensi, sudut rotor, delta sudut rotor, delta kecepatan sudut, dan daya maksimum untuk tiap-tiap mesin yang berpengaruh terhadap ketidakstabilan.

## 2.4. Hasil Simulasi Fenomena *Steady State Stability* dengan Data Nilai-nilai Asumsi Awal

Hasil simulasi fenomena *steady state stability* dengan data nilai-nilai asumsi awal, seperti ditunjukkan Gambar 3.



Gambar 3. Hasil simulasi fenomena *steady state stability* dengan data nilai-nilai asumsi awal

Berdasarkan Gambar 3 ditunjukkan, bahwa waktu pencapaian kondisi stabil diawali oleh mesin-1, kemudian mesin-2, dan terakhir mesin-3 untuk semua parameter. Nilai-nilai sejumlah parameter hasil simulasi dengan data nilai-nilai asumsi, seperti ditunjukkan pada Tabel 5.

Tabel 5. Nilai-nilai sejumlah parameter hasil simulasi dengan data nilai-nilai asumsi

Parameter	Mesin ke-		
	1	2	3
(A) Sudut Rotor			
Nilai awal tidak stabil (derajat)	45,5198	36,7641	33,9222
Nilai stabil (derajat)	35,5185	26,7676	23,9201
Pencapaian waktu stabil (detik)	0,55	0,72	0,91
(B) Selisih sudut rotor			
Nilai awal tidak stabil (derajat)	0,1745	0,1745	0,1745
Nilai stabil (derajat)	0,0000	0,0000	0,0000
Pencapaian waktu stabil (detik)	0,48	0,64	0,68
(C) Selisih kecepatan sudut rotor			
Nilai awal tidak stabil (rad./det.)	-0,4400	-0,3119	-0,359
Nilai stabil (rad./det.)	0,0000	0,0000	0,0000
Pencapaian waktu stabil (detik)	0,56	0,68	0,76
(D) Frekuensi			
Nilai awal tidak stabil (Hz.)	59,9300	59,9504	59,9428
Nilai stabil (Hz.)	60,0000	60,0000	60,0000
Pencapaian waktu stabil (detik)	0,54	0,71	0,76

Berdasarkan Tabel 5 ditunjukkan, bahwa terdapat kemunculan sejumlah fenomena, yaitu (a) perubahan nilai sudut rotor saat pencapaian stabil, (b) pencapaian waktu saat delta rotor mencapai 0 derajat (stabil), (c) pencapaian waktu saat nilai delta omega mencapai 0 rad/det (stabil), dan (d) pencapaian waktu saat nilai frekuensi mencapai 60 Hz. (stabil). Mesin pembangkit dengan pembangkitan daya terbesar (dalam hal ini mesin-3) butuh waktu lebih lama untuk pencapaian kondisi kestabilan kembali setelah terdapat perubahan beban.

### 3. Kesimpulan

Berdasarkan sejumlah uraian pada pembahasan, maka ditarik kesimpulan sesuai tujuan penelitian.

- (1) Struktur sintaks program terdiri atas (i) data *bus* yang meliputi nomor *bus*, tipe *bus*, dan data daya beban (*demand*), (ii) data generator yang meliputi daya pembangkitan, nilai tegangan, dan data basis, dan (iii) data impedans saluran antar *bus*.
- (2) Struktur sintaks program untuk simulasi fenomena aliran daya digunakan untuk penentuan daya mekanis setiap mesin pembangkitan daya akibat perubahan beban, dengan pengoperasian melalui masukan data pada program peng-*input*-an data nilai-nilai asumsi awal. Nilai daya mekanis awal yang di-*input*-kan ke struktur sintaks program untuk fenomena *steady state stability* sebesar 35,43 untuk mesin-1, 45 untuk mesin-2, dan 50 untuk mesin-3.
- (3) Struktur sintaks program untuk fenomena *steady state stability* terdiri atas (i) nilai daya mekanis sebagai data masukan yang diperoleh dari hasil penghitungan program aliran daya; (ii) data tiap mesin berupa nilai tegangan, momen inerti, rasio *damping*, frekuensi, dan resistans dalam mesin; dan (iii) persamaan-persamaan untuk pencarian nilai arus, tegangan, frekuensi, sudut rotor, delta sudut rotor, delta kecepatan sudut, dan daya maksimum untuk tiap-tiap mesin yang berpengaruh terhadap ketidakstabilan.
- (4) Hasil simulasi fenomena *steady state stability* dengan data nilai-nilai asumsi awal berupa kemunculan sejumlah fenomena, yaitu (a) perubahan nilai sudut rotor saat pencapaian stabil, (b) pencapaian waktu saat delta rotor mencapai 0 derajat (stabil), (c) pencapaian waktu saat nilai delta omega mencapai 0 rad/det (stabil), dan (d) pencapaian waktu saat nilai frekuensi mencapai 60 Hz. (stabil). Mesin pembangkit dengan pembangkitan daya terbesar (dalam hal ini mesin-3) butuh waktu lebih lama untuk pencapaian kondisi kestabilan kembali setelah terdapat perubahan beban.

## Daftar Pustaka

- [1] J. Kiusalaas, 2005. "Introduction to MATLAB," in *Numerical Methods in Engineering with MATLAB*. New York, NY: Cambridge University Press, pp. 1-27.
- [2] H. Moore, 2012. "About MATLAB," in *MATLAB for Engineers*, Third Edition. Upper Saddle River, NJ: Pearson Education, Inc., pp. 1-8.
- [3] D. McMahon, 2007. "The MATLAB Environment," in *MATLAB Demystified*. New York, NY: McGraw-Hill Companies, pp. 1-14.
- [4] P. Kundur, J. Paserba, V. Ajjarapu, G. Andersson, A. Bose, C. Canizares, N. Hatziaargyriou, D. Hill, A. Stankovic, C. Taylor, T. van Cutsem, and V. Vittal. "Definition and Classification of Power System Stability," in *IEEE Transactions on Power Systems*, 19(2), May 2004, pp. 1387-1401.
- [5] P. Kundur, 1994. *Power System Stability and Control*. New York, NY: McGraw-Hill, Inc., pp. 3-16.
- [6] H. Saadat, 1999. *Power System Analysis*. New York, NY: McGraw-Hill, Inc., pp. 460-526.
- [7] P.M. Anderson, & A.A. Fouad, 2003. *Power System Control and Stability*. Hoboken: Wiley-IEEE Press, pp. 3-12.
- [8] P.S.R. Murthy, 2007. *Power System Analysis*. Heyderabad: BS Publication, pp. 259-289.
- [9] S.L. Uppal, 1984. *Electrical Power: Generation, Transmission, Distribution, Witchgear and Protection, Utilization of Electrical Energy and Electric Traction*,. Delhi: Khanna Publisher, pp. 654-699.
- [10] D. Das, 2006. *Electrical Power Systems*. New Delhi: New Age International Limited, pp. 276-306.
- [11] B. Ravindranath, & M. Chander, 1977. *Power System Protection and Switchgear*, New Delhi: Wiley Eastern Limited, pp. 1-14.
- [12] U.A. Bakshi, & M.V. Bakshi, 2009. *Protection and Switchgear*. Pune: Technical Publication, pp. 1-34.
- [13] B. Ram, & D.N. Vishwakarma, 2011. *Power System Protection and Switchgear*, Second Edshiition. New Delhi: Tata McGraw-Hill Education Private Limited, pp. 1-31.
- [14] A. Goeritno, & Saidah. "Simulation of Single-phase to Ground Fault to Anticipate Against the Sympathetic Tripping Phenomena," in *Proceeding of the 1<sup>st</sup> ICETIA*, UMS, Surakarta, 2014, pp. 229-233. [Online]. Available: <https://publikasiilmiah.ums.ac.id/xmlui/bitstream/handle/11617/4994/37-Arief%20Goeritno.pdf?sequence=1>
- [15] M.P. Bahrman, & B.K. Johnson. "Transmission Lines: Steady State Operation," in *Power System Analysis and Design*, 5<sup>th</sup> ed. J. Duncan Glover, Mulukutla S. Sarma, and Thomas J. Overbye (editors). Stamford, CT: Cengage Learning, 2012, pp. 233-293..
- [16] IEEE Committee Report, "Proposed Terms and Definitions for Power System Stability," in *IEEE Transaction on Power Apparatus and System*, 101(7), 1982, pp. 1894-1898.
- [17] S.C. Savulescu, "Fast Computation of the Steady-State Stability Limit," in *Real-Time Stability in Power Systems: Techniques for Early Detection of the Risk of Blackout*, 2<sup>nd</sup> ed. S.C. Savulescu (Editor). Cham: Springer International Publishing, 2014, pp. 27-61.
- [18] A. Goeritno, & S. Rasiman. "Performance of Bulk Oil Circuit Breaker (BOCB) Influenced by Its Parameters (Case Study at the Substation of Bogor Baru)," in *the 3<sup>rd</sup> International Conference on Engineering, Technology and Industrial Application* (the 3<sup>rd</sup> ICETIA, 2016), 2017, pp. (020001) 1-12. [Online]. Available: <http://aip.scitation.org/doi/pdf/10.1063/1.4985446>.
- [19] K. Yong-Hak, K. Tae-Kyun, K. No-Hong, C. Jin-Boo, J. Joon-Young, & S. Seok-Ha. "Determination of Synchronous Machine and Excitation Systems Parameters From Field Tests," in *IFAC Power Plants and Power Systems Control*, 2003, pp. 757-760.
- [20] J.J. Grainger, & W.D. Stevenson, 1994. *Power System Analysis*. New York, NY: McGraw-Hill, Inc., pp. 695-747.
- [21] W.K. Ang. *Energy Function for Power Stability Assessment*, Thesis, the University of Queensland, Brisbane, 2003.
- [22] A. Goeritno, & Z. Hardiyanto, (2008, April). *Simulasi Fenomena Aliran Daya pada Sistem Tenaga Listrik "IEEE 5-Bus" Berbasis Metode Numeris dan Berbantuan Aplikasi MATLAB*. Retrieved from: [https://www.researchgate.net/publication/321427006\\_SIMULASI\\_FENOMENA\\_ALIRAN\\_DAYA\\_PA\\_DA\\_SISTEM\\_TENAGA\\_LISTRIK\\_IEEE\\_5-BUS\\_BERBASIS\\_METODE\\_NUMERIS\\_DAN\\_BERBANTUAN\\_APLIKASI\\_MATLAB](https://www.researchgate.net/publication/321427006_SIMULASI_FENOMENA_ALIRAN_DAYA_PA_DA_SISTEM_TENAGA_LISTRIK_IEEE_5-BUS_BERBASIS_METODE_NUMERIS_DAN_BERBANTUAN_APLIKASI_MATLAB).