PERANCANGAN SISTEM PENGENDALIAN *STEAM* TURBIN-GENERATOR BERBASIS *MODEL PREDICTIVE CONTROL* (MPC) DI PT GEO DIPA ENERGI UNIT DIENG JAWA TENGAH

Nama : **Dhita Kurniarum**
NRP : **2408100098**
Jurusan : **Teknik Fisika, FTI - ITS**
Pembimbing : **Katherin Indriawati, ST, MT**

ABSTRAK

Geothermal power plant (GPP) merupakan suatu pembangkit listrik yang memanfaatkan panas bumi sebagai sumber energinya. Dalam suatu GPP, *steam* turbin dan generator merupakan faktor penting yang menentukan performansi sistem secara keseluruhan. Untuk itu, diperlukan suatu sistem pengendali yang mampu menjaga kecepatan putaran turbin dan tegangan terminal generator tetap sesuai *set point*. Dalam tugas akhir ini dirancang sistem pengendalian *steam* turbin dan generator berbasis *Model Predictive Control* (MPC). Pengendalian seperti ini merupakan jenis pengendalian multivariabel dengan menggunakan struktur *cascade*. Kecepatan putaran turbin dikontrol dengan memanipulasi *flowrate steam* melalui *governing system*, sedangkan tegangan terminal dikontrol dengan memanipulasi arus *exciter* pada *excitation system*. *Plant* yang dimodelkan adalah *steam* turbin dan generator yang ada di PT Geo Dipa Energi. Performansi yang ditunjukkan oleh MPC dibandingkan dengan performansi sistem pengendali yang saat ini terpasang di *real plant* (PI *controller*). Berdasarkan pada respon yang ditunjukkan sewaktu simulasi, pengendali MPC memberikan respon yang lebih baik dibandingkan dengan PI *controller*. Kecepatan putaran *steam* turbin mempunyai nilai Mp 0,1%, Ts 13 s dan Ess $3{,}33 \times 10^{-4}$ %. Tegangan generator mempunyai Ts 0,3 s dan Ess 0,33%. Sedangkan, respon *power active* yang dihasilkan mempunyai Mp 100%, Ts 0,6 s dan 0,33%.

Kata kunci : *excitation system*, generator, *governing system*, MPC, *steam* turbin.

DESIGN OF STEAM TURBINE GENERATOR CONTROL SYSTEM BASED ON MODEL PREDICTIVE CONTROL (MPC) AT PT GEO DIPA ENERGI UNIT DIENG CENTRAL JAVA

Name	**: Dhita Kurniarum**
NRP	**: 2408100098**
Department	**: Teknik Fisika, FTI - ITS**
Supervisor	**: Katherin Indriawati, ST, MT**

ABSTRACT

Geothermal power plant (GPP) is a generator using the heat power as the energy. The performance of the steam turbine and generator in GPP is a crucial factor deciding system performance thoroughly. To maintain both steam turbine and generator, a control system with capability of maintaining the turbine's rotation velocity and generator's terminal voltage to stick to the set point is required. A control system of steam turbine and generator based on Model Predictive Control (MPC) is designed in this final project. This controller is a multivariable with cascade. The velocity of turbine rotation is controlled through manipulation of flowrate steam via governing system, while terminal voltage is controlled by manipulation of exciter current on excitation system. The modelled plant is a steam turbine and generator at PT Geo Dipa Energy. The performance shown by MPC is compared with the existing control system at the real plant (PI Controller). According to the response shown in simulation, MPC controller provides better response than PI Controller. The velocity of turbine rotation has Mp 0,1%, Ts 13 s and Ess 3,33x10^{-4} %. The terminal voltage has Ts 0,3 s dan Ess 0,33%. Then, power active has Mp 100%, Ts 0,6 s and 0,33%.

Keywords : *excitation system, generator, governing system,* MPC, *steam turbine.*

“Halaman ini sengaja dikosongkan.”

NOTASI

$\underline{A}$	: Matriks keadaan berdimensi *n x n*
$\underline{B}$	: Matriks keadaan berdimensi *n x l*
$\underline{C}$	: Matriks keadaan berdimensi *m x n*
D	: Koefisien redaman (pu)
E_{exb}	: *Exciter base voltage* (pu)
F	: *Steam turbine torque fraction*
H	: Konstanta inersia (pu)
Hp	: *Prediction horizon*
Hu	: *Control horizon*
i_{exf}	: *Exciter field current* (pu)
i_q & i_d	: q & d-axis *current* (A)
I_t	: Arus generator (pu)
i'_{fd}	: *Field current* (A)
i'_{kq1}, i'_{kq2} & i'_{kd}	: q & d-axis *current* (A)
K_a	: *Voltage regulator gain*
K_e	: *Exciter gain*
K_f	: *Damper gain*
L_{ls}	: *Leakage* induktansi (H)
L_{mq} & L_{md}	: q & d-axis *magnetizing inductance* (H)
L'_{fd}	: *Field leakage induktansi* (H)
L'_{kq1}, L'_{kq2} & L'_{kd}	: q & d-axis *leakage* induktansi (H)
L_{exf}	: *Depent on saturation* (pu)
$\dot{m}$	: *Flow steam* pada *steam* turbin (pu)
P_e	: *Power electric* (pu)
P_{eo}	: *Power active* (pu)
P_{ref}	: Daya referensi (pu)
Q(i) dan R(i)	: Faktor bobot
R_{exf}	: *Exciter resistance* (pu)
R_p	: *Speed droop steam* turbin (pu)
R_s	: Resistansi stator (Ω)
R'_{fd}	: *Field resistance* (Ω)
R'_{kq1}, R'_{kq2} & R'_{kd}	: q & d-axis resistansi (Ω)

$\underline{r}(k+i|k)$: Nilai *reference trajectory*
S_e : Koefisien saturasi
$T_1, T_2, T_3, \& T_4$: *Lead-lag compensator time constant* (s)
T_a : *Regulator time constant* (s)
T_c & T_b : *Gain reduction time constant* (s)
T_d : *Damper time constant* (s)
T_e : Torsi elektrik (pu)
T_{ex} : *Exciter time constant* (s)
T_f : *Damper time constant* (s)
T_m : Torsi mekanik (pu)
T_{sr} : *Speed relay time constant* (s)
T_{st} : *Steam turbine time constant* (s)
T_{vt} : *Speed transmitter time constant* (s)
T_w : *Speed transmitter time constant* (s)
T''_{do} : d-axis *subtransient open-circuit time constant* (s)
$\underline{T}(k)$: Matriks sinyal acuan (*trajectory*)
u : Sinyal kontrol (pu)
$\underline{u}(k)$: Vektor masukan berdimensi-*l*
V_{ex} : *Exciiter field voltage* (pu)
$V_{go/c}$: *Open-circuit generator voltage* (pu)
V_q & V_d : q & d-axis *voltage* (V)
V_R : Sinyal kontrol dari AVR (pu)
V_t : Tegangan terminal generator (pu)
V'_{fd} : *Field voltage* (V)
V'_{kq1}, V'_{kq2} & V'_{kd} : q & d-axis *voltage* (V)
$V(k)$: Fungsi kriteria
ω : Kecepatan putaran *steam* turbin (rpm)
ω_0 : Nominal *speed* generator (rpm)
X_d : d-axis *synchronous* reactance (pu)
X'_d : d-axis *transient reactance* (pu)
X''_d : d-axis *subtransient reactance* (pu) (pu)
X_l : *Leakage* induktansi (pu)
$\underline{x}(k)$: Vektor keadaan berdimensi-*n*
$\underline{Y}(k)$: Matriks keluaran terprediksi

$\underline{y}(k)$	: Vektor keluaran berdimensi-*m*
$\underline{\hat{y}}(k+i\|k)$	: Keluaran terprediksi untuk i-langkah kedepan saat waktu k
φ_d, φ_q & φ_{fd}	: *Fluks* (V.s)
φ_{md}	: *d-axis mutual flux* (pu)
φ_{mq}	: *q-axis mutual flux* (pu)
$\frac{1}{s}$	: *Integrator*
$\underline{\Delta U}(k)$	: Matriks sinyal acuan (*trajectory*)
$\underline{\Delta \hat{u}}(k+i\|k)$	: Perubahan nilai sinyal kendali terprediksi untuk i- langkah kedepan saat waktu k
ΔY	: Bukaan *valve* / *gate opening* (pu)
$\Delta\delta$	: *Rotor angle deviation* (rad)

“*Halaman ini sengaja dikosongkan*.”

BAB II
TINJAUAN PUSTAKA

2.1 Pembangkit Listrik Tenaga Panas Bumi (*Geothermal Power Plant*)

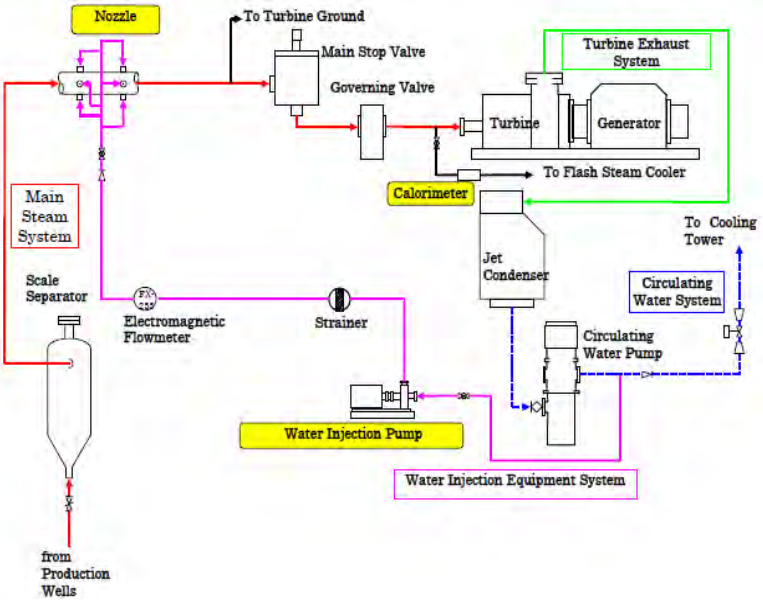
Geothermal power plant merupakan suatu pembangkit listrik yang memanfaatkan panas bumi sebagai bahan baku utamanya. Berdasarkan pada jenis fluida dan kandungan utamanya, sistem *geothermal* dibedakan menjadi dua, yaitu sistem satu fasa dan sistem dua fasa. Pada sistem satu fasa, jenis fluida yang dihasilkan adalah uap kering saja. Sedangkan, sistem dua fasa merupakan campuran air dan uap. Pada umumnya, sumber panas bumi yang ada di dunia menghasilkan fluida dua fasa.

Pembangkit Listrik Tenaga Panas bumi (PLTP) pada prinsipnya sama seperti Pembangkit Listrik Tenaga Uap (PLTU). Perbedaan keduanya terletak pada uap yang digunakan. PLTU menggunakan uap yang dihasilkan oleh boiler, sedangkan uap yang digunakan pada PLTP berasal dari reservoir panas bumi. Apabila fluida di kepala sumur berupa fasa uap, maka uap tersebut dapat dialirkan langsung ke turbin dan kemudian turbin akan mengubah energi panas bumi menjadi energi gerak yang akan memutar generator sehingga menghasilkan energi listrik.

Secara umum suatu PLTP memiliki *well pad*, separator, *steam* turbin, generator, kondenser, *injection pump* dan lainnya seperti yang terlihat pada gambar 2.1. *Well pad* atau yang biasa disebut sumur produksi merupakan tempat eksplorasi fluida kerja yang berasal dari dalam perut bumi. Fluida tersebut merupakan fluida dua fasa, yaitu fasa uap dan fasa cair. Untuk memisahkan kedua fasa tersebut digunakanlah separator. Konstruksi dari *separator* berupa sekat-sekat yang dapat menahan air agar tidak terbawa oleh uap bertekanan tinggi. Hasil dari *separator* berupa *saturated steam* yang akan dikirim ke *power plant*, sedangkan sisanya berupa *brine* (air yang banyak mengandung mineral seperti *silica*) akan dikirim ke kolam penampungan untuk dibuang

atau diinjeksikan kembali ke dalam bumi melalui sumur injeksi. Pada *power plant*, *steam* yang dikirim dari separator akan masuk ke dalam *steam* turbin dan menggerakkan *blade* yang ada di dalamnya. Gerakan ini akan menghasilkan putaran turbin, torsi mekanik dan daya mekanik. Selanjutnya daya mekanik tersebut akan dikonversi oleh generator menjadi energi listrik.

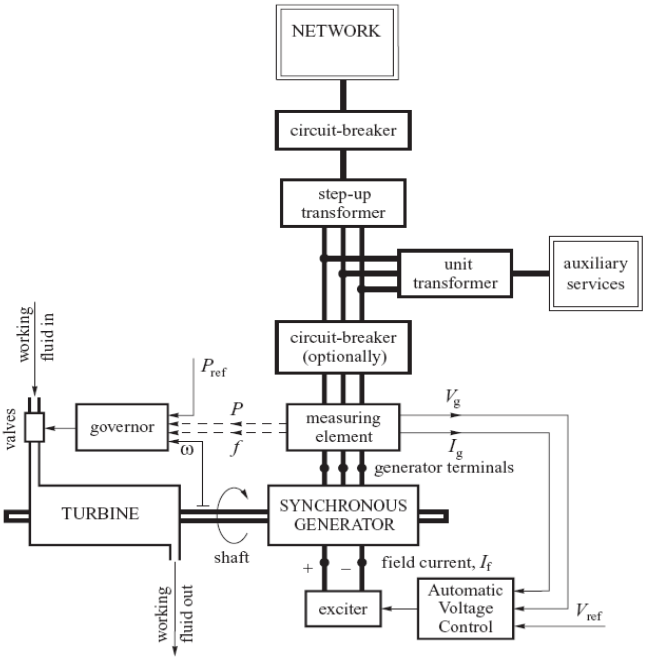
Bagian-bagian tersebut sebenarnya tidak mutlak dimiliki pada setiap PLTP, namun disesuaikan dengan jenis PLTP. Saat ini terdapat banyak jenis PLTP, diantaranya *Dry Steam Single Flash Steam*, *Double Flash Steam*, *Multi Flash Steam*, atau *Binary Cycle* (DiPippo, 2005). Tetapi, apapun jenis PLTP, pastilah ditemui *steam* turbin dan generator pada PLTP tersebut. Berhubung tugas akhir ini membahas tentang sistem pengendalian *steam* turbin dan generator, maka tinjauan pustaka tentang kedua komponen tersebut akan diperdalam lagi.



Gambar 2.1 *Geothermal power plant* (Tohoku *Electric Power*, 2006)

2.2 Perancangan Sistem Pengendalian *Steam* Turbin dan Generator

Sistem pengendalian *steam* turbin dan generator dilakukan demi menjaga variabel proses tetap pada nilai konstan sesuai *set point* yang telah ditentukan. Dalam hal ini, variabel proses yang dikontrol adalah kecepatan putaran turbin dan tegangan output generator. Untuk mengendalikan variabel tersebut, dibutuhkan suatu sistem pengendalian yang terdiri dari beberapa komponen, yaitu *governing system* untuk mengendalikan kecepatan putaran steam turbin dan *excitation system* untuk mengendalikan tegangan output generator (IEEE *Committee Report*, 1973 ; Jadric, 1998 ; Machowski, dkk, 2008 ; Robandi, 2009). Adapun gambaran umum *plant* bisa dilihat pada gambar 2.2.

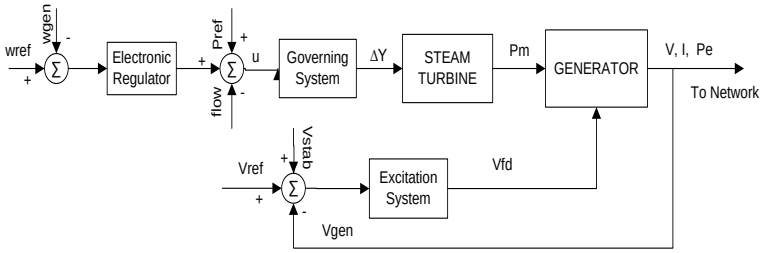


Gambar 2.2 Unit *power generation system* (Machowski, dkk, 2008)

Seperti yang terlihat pada gambar 2.2, sistem bekerja dengan daya referansi tertentu. Daya tersebut bisa terpenuhi apabila *steam* yang masuk ke dalam *steam* turbin sesuai kebutuhan. Agar suplay *steam* tersebut sesuai, maka laju *steam* yang masuk ke dalam *steam* turbin dikendalikan oleh *governing system* dengan cara mengatur bukaan *governor valve*. *Steam* yang masuk ke dalam *steam* turbin akan memutar *blade* yang ada di dalamnya dan secara otomatis akan memutar *shaft* yang menghubungkan *steam* turbin dan generator. Dengan adanya perputaran *shaft* pada daerah medan magnet, maka akan menyebabkan perbedaan fluks sehingga menimbulkan ggl induksi dan dihasilkan energi listrik.

Energi listrik yang dihasilkan oleh *power plant* akan dikirim ke suatu *network*. Agar energi listrik yang dihasilkan bisa dikirim ke *network*, maka antara *power plant* dan *network* harus ada sinkronisasi frekuensi dan tegangan. Karena itu, kedua variabel tersebut harus dikendalikan. Frekuensi bisa dikendalikan dengan mengendalikan putaran *steam* turbin. Hal ini dikarenakan selisih frekuensi pada waktu tertentu sama dengan selisih kecepatan putaran turbin. Sedangkan, tegangan output dari generator dikendalikan oleh *excitation system* (terdiri dari AVR dan *exciter*) dengan cara memanipulasi arus *exciter*.

Berdasarkan gambar 2.2 bisa dibuat sistem pengendalian *steam* turbin dan generator seperti yang terlihat pada gambar 2.3.



Gambar 2.3 Diagram blok sistem pengendalian *steam* turbin dan generator (Machowski, dkk, 2008)

Masing-masing komponen yang ada dalam diagram blok pada gambar 2.3 akan dijelaskan lagi pada sub bab selanjutnya.

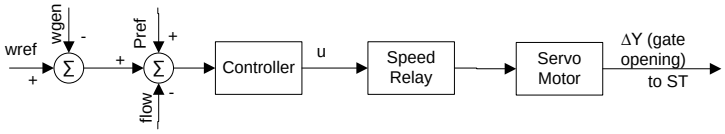
2.2.1 *Steam* Turbin dan *Governing System*

Steam turbin merupakan suatu mesin yang mengubah energi kalor (*steam*) menjadi energi mekanik. *Steam* turbin menggunakan *saturated steam* sebagai fluida kerjanya. *Steam* yang masuk akan menggerakkan *shaft* yang menjadi penghubung antara turbin dan generator. Kuantitas *steam* yang masuk turbin akan mempengaruhi kecepatan putaran turbin dan daya yang dihasilkan. Pada sistem tenaga listrik, kecepatan putaran turbin harus dijaga tetap sesuai *set point*. Hal ini dikarenakan kualitas kecepatan putaran *steam* turbin merupakan faktor utama yang menentukan kualitas sistem secara keseluruhan (Robandi, 2009). Apabila kecepatan putaran yang dihasilkan tidak sesuai, maka akan terjadi trip sampai kerusakan pada generator dan pembangkit.

Gambar 2.4 *Steam* turbin (Sakai, 2009)

Kecepatan putaran *steam* turbin dikendalikan oleh *governing system* dengan cara memanipulasi *gate opening* / bukaan *governor valve*. *Governing system* terdiri dari beberapa komponen, diantaranya : EHC (*Electric Hydraulic Converter*) dan servo motor. EHC akan mengubah sinyal kontol elektrik menjadi

hydraulic. Selanjutnya sinyal *hydraulic* tersebut akan diteruskan menuju servo motor dan servo motor tersebut akan membuat *governor valve* membuka sesuai sinyal kontrol yang diberikan. Adapun diagram blok *governing system* bisa dilihat pada gambar 2.5 di bawah ini.



Gambar 2.5 Diagram blok *governing system* (IEEE *Committee Report*, 1973)

Pemodelan *steam* turbin dan *governor system* terbagi menjadi tiga, yaitu : pemodelan *governor valve*, *steam* turbin dan *shaft*. Adapun penjelasannya bisa dilihat di bawah ini.

- *Governor valve* (IEEE *Committee Report*, 1973)

Pemodelan *governor valve* dibuat berdasarkan persamaan input-output yang ada pada diagram blok gambar 2.5. Untuk lebih jelasnya bisa dilihat pada persamaan di bawah ini.

$$\frac{output}{input} = \frac{\Delta Y}{u} \tag{2.1}$$

dengan

$$u = error * Controller \tag{2.2}$$

dan

$$error = \left(\Delta\omega\left(\frac{1}{R_p}\right) + P_{ref} - \dot{m}\right) \tag{2.3}$$

Sinyal kontrol yang dihasilkan (u) akan diteruskan menuju *speed relay* lalu servo motor. Pemodelan *speed relay* dibuat dalam bentuk fungsi transfer seperti yang ada pada persamaan (2.4).

$$TF_{speed\ relay} = \frac{1}{T_{sr}\ s+1} \tag{2.4}$$

Servo motor merupakan komponen yang mengatur bukaan *valve* sesuai sinyal kontrol yang diberikan.

- *Steam* turbin (IEEE *Committee Report*, 1973)

Pemodelan untuk *steam* turbin dibuat berdasarkan hubungan input-output seperti yang ditunjukkan persamaan di bawah ini.

$$\frac{output}{input} = \frac{T_m}{\Delta Y} \tag{2.5}$$

dengan

$$T_m = F\dot{m} \tag{2.6}$$

F merupakan *fraction* yang mengkonversi *steam flowrate* menjadi torsi mekanik yang dihasilkan.

Pemodelan *steam* turbin dalam bentuk fungsi transfer menjadi seperti di bawah ini.

$$TF_{ST} = \frac{1}{T_{st}\ s+1} \tag{2.7}$$

- *Shaft*

Output dari *shaft* adalah daya mekanik yang diperoleh berdasarkan persamaan di bawah ini.

$$P_m = \omega\ T_m \tag{2.8}$$

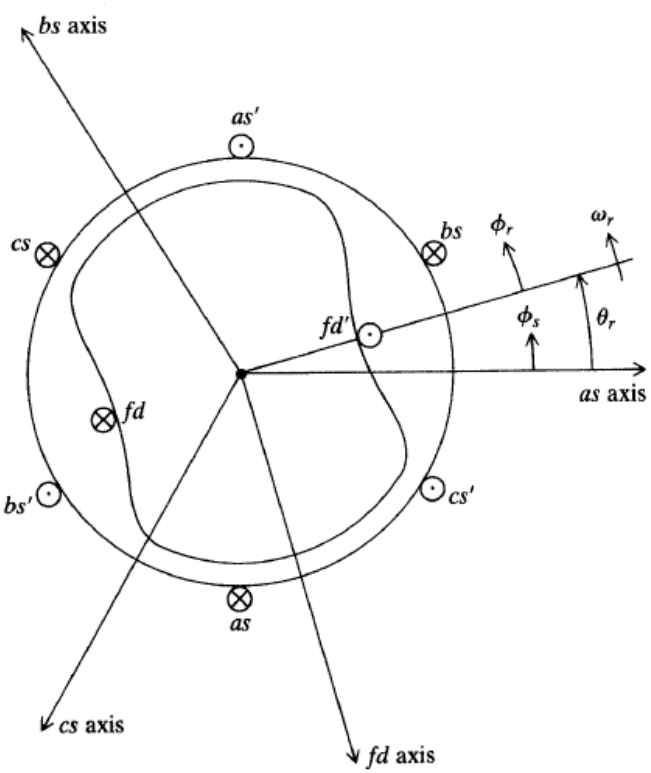
2.2.2 Generator Sinkron Tiga Fasa

Generator merupakan suatu *instrument* yang berfungsi untuk mengkonversi energi mekanik menjadi energi listrik bolak-balik secara elektromagnetik. Input generator merupakan energi mekanik yang berasal dari motor sinkron (*shaft*) yang terhubung dengan *steam* turbin. Energi listrik diperoleh dari proses induksi elektromagnetik yang terjadi pada kumparan-kumparan stator. Secara umum, generator terdiri dari beberapa komponen, yaitu : stator, rotor dan celah udara diantara keduanya. Stator merupakan bagian dari generator sinkron yang diam, sedangkan rotor merupakan bagian yang berputar, dimana diletakkan kumparan medan yang disuplai oleh arus searah dari sistem eksitasi.

Prinsip generator secara umum bisa dijelaskan sebagai berikut.

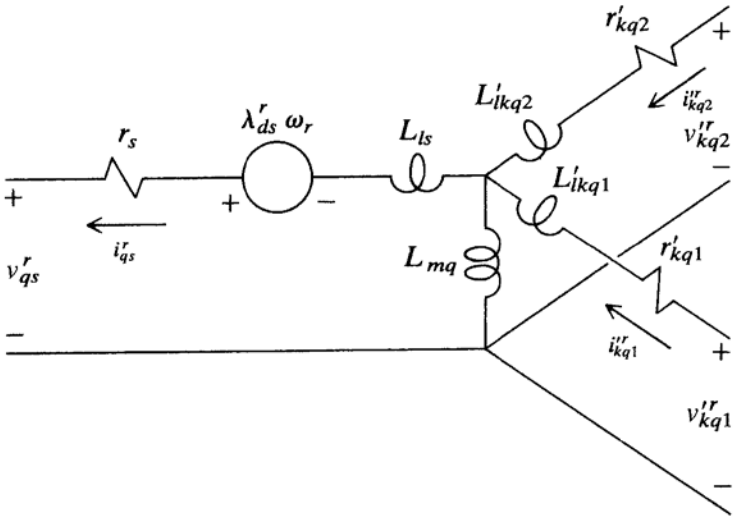
a. Kumparan medan yang diletakkan pada rotor dihubungkan dengan sumber eksitasi yang akan mensuplai arus searah untuk kumparan medan tersebut.
b. Adanya arus searah yang mengalir melalui kumparan medan akan menimbulkan fluks.
c. *Prime mover* yang sudah terkopel dengan rotor segera dioperasikan sehingga rotor akan berputar dengan kecepatan tertentu.
d. Perputaran rotor tersebut sekaligus akan memutar medan magnet yang dihasilkan oleh kumparan medan.
e. Medan putar yang dihasilkan pada rotor akan diinduksikan pada kumparan jangkar sehingga kumparan jangkar yang terletak di stator akan menghasilkan fluks magnetik yang berubah tiap waktu.
f. Adanya perubahan fluks magnet akan menimbulkan ggl induksi pada ujung-ujung kumparan.

Sebuah mesin sinkron terdiri dari tiga *stator winding* dan satu *field winding* seperti yang ditunjukkan oleh gambar 2.6.

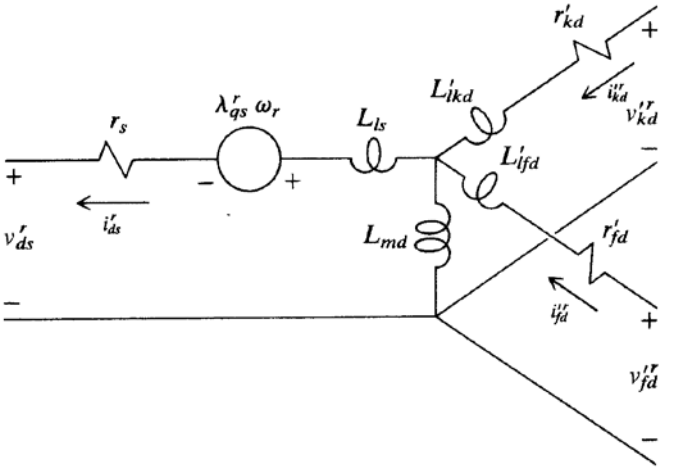


Gambar 2.6 Ilustrasi d-q axis pada mesin sinkron 2 *pole* (Krause, dkk, 2002)

Generator dimodelkan menjadi dua bagian, yaitu : mekanik dan elektrik. Daya yang dihasilkan oleh generator diperoleh dari pemodelan elektriknya. Adapun pemodelan elektriknya didasarkan pada gambar 2.7 yang merupakan rangkaian d-axis dan q-axis yang terdapat pada generator sinkron *salient-pole* tiga fasa.



(a)



(b)

Gambar 2.7 (a) Rangkaian d-axis pada generator sinkron ; (b) Rangkaian q-axis pada generator sinkron (Krause, 2002)

Berdasarkan pada rangkaian pada gambar 2.7, persamaan pemodelan elekttrik *synchronous* generator bisa ditulis seperti di bawah ini.

$$V_d = R_s i_d + \frac{d}{dt}\varphi_d - \omega_R \varphi_q \tag{2.9}$$

$$V_q = R_s i_q + \frac{d}{dt}\varphi_q - \omega_R \varphi_d \tag{2.10}$$

$$V'_{fd} = R'_{fd} i'_{fd} + \frac{d}{dt}\varphi'_{fd} \tag{2.11}$$

$$V'_{kd} = R'_{kd} i'_{kd} + \frac{d}{dt}\varphi'_{kd} \tag{2.12}$$

$$V'_{kq1} = R'_{kq1} i'_{kq1} + \frac{d}{dt}\varphi'_{kq1} \tag{2.13}$$

$$V'_{kq2} = R'_{kq2} i'_{kq2} + \frac{d}{dt}\varphi'_{kq2} \tag{2.14}$$

dengan

$$\varphi_d = L_d i_d + L_{md}\left(i'_{fd} + i'_{kd}\right) \tag{2.15}$$

$$\varphi_q = L_q i_q + L_{mq} i'_{kq} \tag{2.16}$$

$$\varphi'_{fd} = L'_{fd} i'_{fd} + L_{md}\left(i_d + i'_{kd}\right) \tag{2.17}$$

$$\varphi'_{kd} = L'_{kd} i'_{kd} + L_{md}\left(i_d + i'_{fd}\right) \tag{2.18}$$

$$\varphi'_{kq1} = L'_{kq1} i'_{kq1} + L_{mq} i_q \tag{2.19}$$

$$\varphi'_{kq2} = L'_{kq2} i'_{kq2} + L_{mq} i_q \tag{2.20}$$

Parameter yang digunakan dalam pemodelan *plant* menggunakan satuan pu (per unit). Persamaan di atas bisa ditulis menggunakan

satuan pu seperti persamaan di bawah ini (IEEE : *Power System Engineering Commitee*, 1985).

$$L_{ls} = L_{ad} = X_l \tag{2.21}$$

$$L_{md} = X_d - X_l \tag{2.22}$$

$$L_{fd} = L_{md}\left(\frac{X'_d - X_l}{X_d - X'_d}\right) \tag{2.23}$$

$$R_{fd} = \frac{L_{fd} + L_{md}}{\omega_0 T''_{do}} \tag{2.24}$$

$$L_{kd} = \frac{L_{md}\, L_{fd}\,(X''_d - X_l)}{L_{md}\, L_{fd} - [(L_{md} + L_{fd})(X''_d - X_l)]} \tag{2.25}$$

$$R_{kd} = \frac{1}{\omega_0 T''_{do}}\left[L_{kd} + \left(\frac{L_{md}\, L_{fd}}{L_{md} + L_{fd}}\right)\right] \tag{2.26}$$

Perhitungan untuk q-axis juga menggunakan persamaan yang serupa seperti persamaan di atas.

Energi mekanik yang masuk ke dalam generator akan dikonversi menjadi energi listrik. Adapun perhitungannya bisa dilihat pada persamaan di bawah ini.

$$P_e = T_e\, \omega \tag{2.27}$$

Torsi elektrik diperoleh berdasarkan pada persamaan di bawah ini.

$$T_e = \omega_d i_q - \omega_q i_d \tag{2.28}$$

$$\omega = \frac{1}{2Hs}(T_m - T_e - D\,\Delta\omega) \tag{2.29}$$

dengan

$$i_q = \left(\varphi_q - \varphi_{mq}\right)\frac{-1}{L_l} \tag{2.30}$$

$$i_d = \left(\varphi_d - \varphi_{md}\right)\frac{-1}{L_l} \tag{2.31}$$

dan

$$\varphi_{mq} = \varphi_q \frac{L_{aq}}{L_l} + \varphi'_{kq1}\frac{L_{aq}}{L_{kq1}} + \varphi'_{kq2}\frac{L_{aq}}{L_{kq2}} \tag{2.32}$$

$$\varphi_{md} = \varphi_q \frac{L_{aq}}{L_l} + \varphi'_{fd}\frac{L_{aq}}{L_{fd}} + \varphi'_{kd}\frac{L_{aq}}{L_{kd}} \tag{2.33}$$

Sedangkan *rotor angle deviation* dari generator bisa dimodelkan dengan persamaaan di bawah ini.

$$\Delta\delta = \dot{\omega}\frac{1}{s}\left(\omega_0 \frac{pi}{30}\right) \tag{2.34}$$

Output yang dihasilkan generator adalah tegangan, arus dan daya aktif. Adapun perhitungannya bisa dilihat pada persamaan di bawah ini.

$$V_t = \sqrt{V^2{}_q + V^2{}_d} \tag{2.35}$$

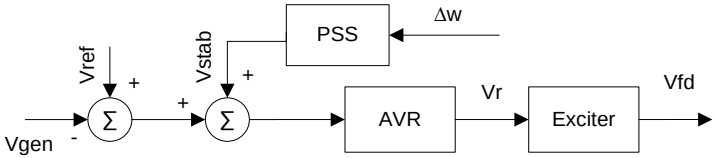
$$I_t = \sqrt{i^2{}_q + i^2{}_d} \tag{2.36}$$

$$P_{eo} = V_d i_d + V_q i_q \tag{2.37}$$

2.2.3 *Excitation System*

Excitation system adalah suatu sistem yang fungsinya untk menjaga tegangan keluaran generator agar tetap berada pada

kondisi konstan sesuai tegangan nominalnya. *Excitation system* terdiri dari *Automatic Voltage Regulator* (AVR), *Power System Stabilizer* (PSS), dan *exciter*. Adapun diagram blok sistem pengendaliannya bisa dilihat pada gambar 2.8 di bawah ini.



Gambar 2.8 Diagram blok *excitation system* (Loukianov, dkk, 2011)

Seperti yang terlihat pada gambar 2.8, input yang digunakan dalam *excitation system* adalah akumulasi dari tegangan referensi generator, tegangan output / terminal generator dan tegangan output *Power System Stabilizer* (PSS). Apabila ditulis dalam persamaan menjadi seperti di bawah ini.

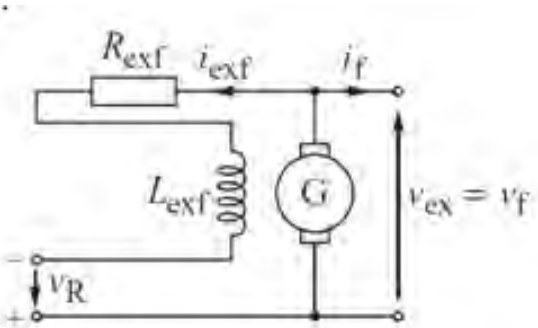
$$V_{in} = V_{ref} - V_{gen} + V_{stab} \tag{2.38}$$

Sinyal V_{in} nantinya akan menjadi input bagi AVR (*Automatic Voltage Regulator*). AVR merupakan sebuah sistem kelistrikan yang berfungsi untuk menjaga agar tegangan generator tetap sesuai *set point*.

Prinsip kerja dari AVR adalah mengatur arus penguatan *exciter*. Apabila tegangan output generator berada dibawah tegangan nominalnya maka AVR akan memperbesar arus pada *exciter*. Sebaliknya, apaabila tegangan output generator melebihi tegangan nominalnya maka AVR akan mengurangi arus pada *exciter*. Bisa dikatakan bahwa AVR tersebut merupakan suatu *controller* yang berfungsi untuk mengubah sinyal error menjadi sinyal kontrol. AVR mempunyai peran yang penting dalam suatu pembangkit. Hal ini dikarenakan AVR mampu menjaga tegangan

terminal generator tetap pada nilai nominalnya walaupun ada perubahan *load*.

Sinyal dari AVR nantinya akan diteruskan menuju *exciter*. Adapun rangkaian *exciter* bisa dilihat pada gambar 2.9 di bawah ini.



Gambar 2.9 *Self exciter* (Machowski, dkk, 2008)

Perubahan arus eksitasi bisa didiskripsikan pada persamaan di bawah ini (Machowski, dkk, 2008).

$$v_r = R_{exf} + L_{ext}\left(\frac{d}{dt} i_{ext}\right) \tag{2.39}$$

dengan

$$i_{exf} = \frac{v_{ex}}{R}(1 + S_e) \tag{2.40}$$

$$L_{exf}\,\frac{d}{dt} i_{exf} = L_{exf}\left[\frac{di_{exf}}{dv_{exf}}\right]_{v_{ex\,0}}\,\frac{d}{dt} V_{ex} \tag{2.41}$$

sehingga

$$V_R = \frac{R_{exf}}{R}(1 + S_e)V_{ex} + L_{exf}\left[\frac{di_{exf}}{dv_{exf}}\right]_{v_{ex\,0}}\,\frac{d}{dt} V_{ex} \tag{2.42}$$

Untuk mempermudah *interface* antara eksitasi dengan generator, maka jumlah basis exciter didefinisikan sebagai E_{exb}. Ini berarti E_{exb} dan tegangan generator bisa dihubungkan dalam suatu persamaan (2.44) seperti di bawah ini.

$$V_{go/c} = \frac{1}{\sqrt{2}} \frac{\omega M_f}{R_f} E_{exb} \tag{2.43}$$

jika

$$R_b = R \tag{2.44}$$

$$I_{exfb} = \frac{E_{exb}}{R_b} \tag{2.45}$$

dan

$$L_{exf} \left[\frac{di_{exf}}{dv_{exf}}\right]_{v_{ex\,0}} \frac{d}{dt} V_{ex} = \frac{L_{exf}}{R} \left[\frac{dI_{exf}}{dV_{exf}}\right]_{v_{ex\,0}} \tag{2.46}$$

maka

$$V_R = \frac{R_{exf}}{R}(1 + S_e)V_{ex} + \frac{L_{exf}}{R} \left[\frac{dI_{exf}}{dV_{exf}}\right]_{v_{ex\,0}} \frac{d}{dt} V_{ex} \tag{2.47}$$

jika

$$E_f = V_{ex} \tag{2.48}$$

$$\frac{L_{exf}}{R} \left[\frac{dI_{exf}}{dV_{exf}}\right]_{v_{ex\,0}} = T_E \tag{2.49}$$

$$K_E = \frac{R_{exf}}{R} \tag{2.50}$$

$$S_E = \frac{R_{exf}}{R} S_e \tag{2.51}$$

maka

$$V_R = (K_E + S_E)E_f + T_E\left(\frac{dE_f}{dt}\right) \quad (2.52)$$

Nilai S_E adalah 0, sehingga persamaan (2.52) bisa ditulis dalam bentuk fungsi transfer seperti yang terlihat pada persamaan (2.53).

$$TF_{Exciter} = \frac{K_e}{T_{ex}\ s+1} \quad (2.53)$$

Sebelum dijadikan sebagai *feedback*, output *excitation* dilewatkan *damper* terlebih dahulu. Adapun pemodelan *damper* tersebut bisa dilihat pada persamaan (2.54) di bawah ini.

$$TF_{Vf.damper} = \frac{K_f\ s}{T_f\ s+1} \quad (2.54)$$

Tegangan terminal generator juga dilewatkan *damper* terlebih dahulu sebelum masuk ke *excitation system*. Adapum pemodelannya dibuat dalam bentuk fungsi transfer seperti yang terlihat pada persamaan (2.55) di bawah ini.

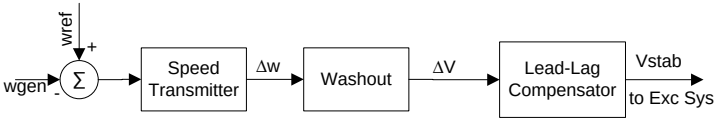
$$TF_{damper} = \frac{1}{T_d\ s+1} \quad (2.55)$$

Aksi AVR dan *exciter* dapat mempengaruhi kestabilan daya yang dihasilkan. Hal ini bisa berlangsung lama sehingga dapat membahayakan keamanan sistem dan batas transfer daya. Faktor-faktor yang dapat mempengaruhi kestabilan tersebut antara lain (Robandi, 2009) :

- Pembebanan generator
- Kemampuan transfer daya dari saluran transmisi
- Faktor daya
- Penguatan AVR dan penguatan yang dilakukan *controller* lainnya.

Untuk menyelesaikan masalah tersebut diperlukan suatu redaman yang bisa mengurangi osilasi yang terjadi. Karena dalam *excitation system* ditambahkan *controller* lain yang mampu mengurangi osilasi tersebut. *Power System Stabilizer* (PSS) merupakan suatu sistem yang biasa digunakan untuk menangani masalah tersebut. Biasanya PSS memang digunakan untuk mendampingi kerja sistem eksitasi dan difungsikan sebagai peredam untuk mengurangi osilasi yang terjadi.

PSS biasanya terdiri dari sensor, *washout*, kompensator dinamik, dan *limiter*. Sinyal kontrol yang digunakan sebagai input PSS adalah selisih kecepatan rotor generator pada waktu tertentu. Setelah melalui PSS, sinyal tersebut akan berubah menjadi teegangan hasil stabilisasi yang nantinya akan diakumulasi bersama tegangan nominal dan terminal generator untuk dijadikan input bagi *excitation system.* Adapun diagram blok dari PSS bisa dilihat pada gambar 2.10 di bawah ini.



Gambar 2.10 Diagram blok PSS (Robandi, 2009)

Pemodelan untuk masing-masing komponen dalam PSS bisa dilihat di bawah ini.

- *Speed transmitter*

Merupakan pendeteksi sinyal kontrol yang masuk dalam PSS. *Transmitter* yang digunakan adalah *velocity transmitter*. Pemodelannya dibuat dalam bentuk fungsi transfer seperti persamaan (2.56) di bawah ini.

$$TF_{vt} = \frac{1}{T_{vt}\, s+1} \qquad (2.56)$$

- *Washout*

Merupakan rangkaian yang digunakan untuk mengeliminasi sinyal bias pada kondisi *steady state* pada output PSS. Rangkaian *washout* penting karena digunakan untuk melewatkan frekuensi tinggi dan harus dapat dilewati oleh semua frekuensi yang digunakan (Robandi, 2009). Adapun pemodelannya bisa dilihat pada persamaan (2.57) di bawah ini.

$$TF_{washout} = \frac{T_w\, s}{T_w\, s+1} \tag{2.57}$$

- Kompensator dinamik

Kompensator dinamik biasa juga disebut dengan *lead-lag compensator*. Dalam dunia industri, biasanya digunakan dua kompensator. Adapun pemodelannya bisa dilihat pada pada persamaan (2.58) dan (2.59).

$$\mathrm{TF}_{\mathrm{lead-lag\_1}} = \frac{\mathrm{T}_1\, s+1}{\mathrm{T}_2\, s+1} \tag{2.58}$$

$$\mathrm{TF}_{\mathrm{lead-lag\_2}} = \frac{\mathrm{T}_3\, s+1}{\mathrm{T}_4 s+1} \tag{2.59}$$

- *Limiter*

Limiter digunakan untuk membatasi output PSS dalam mendampingi kerrja AVR. Pada umumnya, output PSS dibatasi dengan *lower limit* -0,05 pu dan *upper limit* 0,2 pu (Robandi, 2009).

2.2.4 *Network*

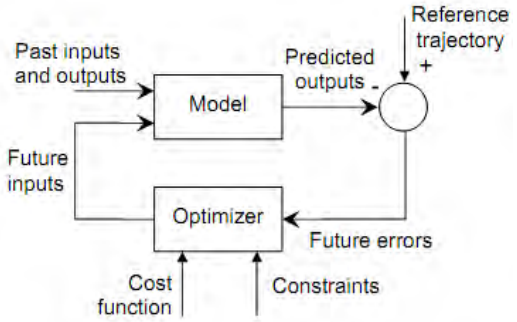
Network yang dimaksud di sini adalah jaringan yang terhubung dengan generator. Di PT Geo Dipa, generator terhubung dengan *step up* transformer 15/150 kV. Sebelum dialirkan ke PLN, tegangan output generator akan dinaikkan dulu menjadi 150 kV melalui *step up* transformer tersebut. Hal ini dilakukan supaya sinkronisasi dua sumber arus AC bisa dihubungkan. Sinkronisasi hanya dapat dilakukan jika kedua

sumber mempunyai tegangan, frekuensi, urutan fasa, dan sudut fasa yang sama. Apabila tegangan pada pembangkit lebih rendah dari tegangan bus PLN, maka jaringan tersebut akan mengalami *reserve power*.

2.3 Algoritma *Model Predictive Control* (MPC)

Model Predictive Control (MPC) adalah suatu *controller* prediktif yang mampu memprediksi sinyal kontrol untuk beberapa waktu ke depan. Algoritma MPC secara eksplisit menggunakan model proses untuk memprediksi efek dari sinyal kontrol (variabel manipulasi) terhadap output proses yang akan datang. Terdapat dua model yang digunakan dalam MPC, yaitu : model proses dan model gangguan. Perubahan sinyal kontrol ditentukan berdasarkan prosedur optimasi dari suatu fungsi objektif yang meminimumkan *error* prediksi. Optimasi dilakukan dengan memperhitungkan gangguan yang dimiliki oleh proses. Oleh karena itu kehadiran model proses merupakan elemen penting dalam algoritma MPC.

Pengendali prediktif menggunakan model proses untuk mengetahui perilaku sistem. Dengan model proses ini, perilaku sistem diprediksi dalam kurun waktu tertentu. Hasil dari prediksi digunakan untuk tiap waktu pencuplikan guna mengoptimalkan keluaran sistem melalui sinyal masukan (Venkat, 2006).



Gambar 2.11 Struktur dasar MPC (Camacho, 1999)

Metode kontrol MPC mempunyai beberapa keunggulan jika dibandingkan metode kontrol lainnya, diantaranya adalah (Fahrudin, 2010) :

- Konsepnya intuitif serta penalaannya mudah.
- Dapat digunakan untuk mengendalikan proses yang beragam, mulai dari proses yang sederhana sampai proses yang kompleks, proses yang mempunyai waktu tunda besar, proses *non-minimum phase* atau pun proses yang *unstable*.
- Dapat menangani sistem pengendalian multivariabel.
- Mempunyai kompensasi terhadap waktu tunda.
- Mempunyai kemampuan dari pengendali *feed forward* untuk mengkompensasi gangguan yang terukur.
- Mudah untuk mengimplementasikan pengendali yang diperoleh.
- *Constraint*s dapat diperhitungkan didalam perancangan pengendali.

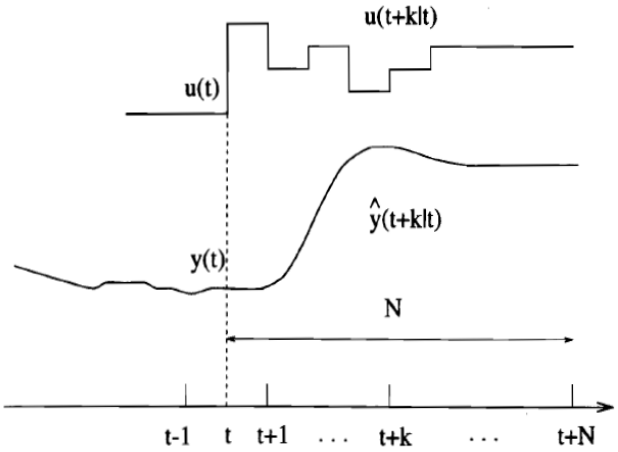
Selain mempunyai banyak keunggulan, MPC juga mempunyai kekurangan. Salah satu diantaranya adalah harus tersedianya model proses yang sesuai dengan *plant*. Dalam hal ini, kualitas model proses merupakan aspek penentu kualitas pengendalian. Hal ini karena apabila ada kesalahan model proses akan berpengaruh pada performansi pengendali. Kekurangan yang lain adalah masalah penurunan aturan sinyal kendali. Walaupun aturan kendali mudah diimplementasikan dan membutuhkan sedikit komputasi, namun penurunan aturan kendali tersebut lebih kompleks dibandingkan pengendali PID.

Konsep Dasar MPC bisa dijelaskan oleh gambar 2.12 dan gambar 2.13 dengan penjelasan sebagai berikut (Camacho, 2004).

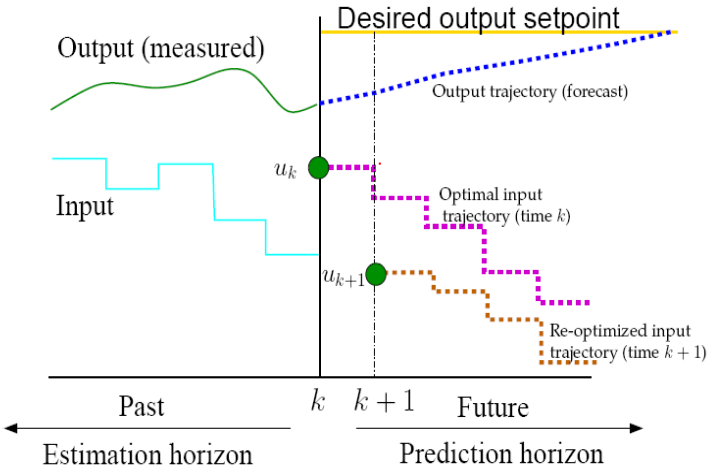
a. Keluaran proses yang akan datang untuk rentang *horizon* N yang ditentukan, dinamakan *prediction horizon*, diprediksi pada setiap waktu pencuplikan dengan menggunakan model proses. Keluaran proses terprediksi ini y(t+k|t) untuk k = 1 ... N bergantung pada nilai masukan dan keluaran lampau serta sinyal kendali yang akan datang u(t+k|t), k =

0 ... N-1 yang akan digunakan sistem dan yang harus dihitung.

b. Serangkaian sinyal kendali dihitung dengan mengoptimasi suatu fungsi kriteria yang ditetapkan sebelumnya, dengan tujuan untuk menjaga proses sedekat mungkin terhadap trayektori acuan w (t+k). Fungsi kriteria tersebut umumnya berupa suatu fungsi kuadratik dari kesalahan antara sinyal keluaran terprediksi dengan trayektori acuan. Dalam banyak kasus tujuan pengendalian seperti pemakaian energi yang minimum disertakan didalam fungsi kriteria. Solusi eksplisit dapat diperoleh jika fungsi kriteria adalah kuadratik, model linier, dan tidak ada *constraint*s. Apabila tidak, optimasi iteratif harus digunakan untuk memecahkannya.
c. Sinyal kendali u(t|t) dikirim ke proses, sedangkan sinyal kendali terprediksi berikutnya dibuang, karena pada pencuplikan berikutnya y(t+1) sudah diketahui nilainya. Langkah pertama diulang dengan nilai keluaran proses yang baru dan semua prosedur perhitungan yang diperlukan diperbaiki. Sinyal kendali yang baru u(t+1|t+1) dihitung dengan menggunakan konsep *receding horizon*.



Gambar 2.12 Strategi kontrol MPC (Camacho, 2004)



Gambar 2.13 Konsep dasar MPC (Venkat, 2006)

Konsep MPC terdiri dari beberapa elemen utama, yaitu: model proses, fungsi kriteria, *constrains*, dan *qudratic progamming* yang akan dijelaskan pada sub bab selanjutnya.

2.3.1 Model Proses

Model proses yang digunakan dalam perancangan pengendali MPC umumnya berupa model *state space* seperti yang ditunjukkan persamaan (2.60) dan (2.61) di bawah ini.

$$\underline{x}(k+1) = \underline{A}\underline{x}(k) + \underline{B}\underline{u}(k) \quad (2.60)$$

$$\underline{y}(k) = \underline{C}\underline{x}(k) \quad (2.61)$$

Setelah model ruang keadaan diskrit linier dari sistem diperoleh, selanjutnya perhitungan prediksi dapat dinyatakan sebagai berikut :

$$\begin{bmatrix} \underline{\hat{x}}(k+1|k) \\ \vdots \\ \underline{\hat{x}}(k+H_u|k) \\ \underline{\hat{x}}(k+H_u+1|k) \\ \vdots \\ \underline{\hat{x}}(k+H_p|k) \end{bmatrix} = \underbrace{\begin{bmatrix} \underline{A} \\ \vdots \\ \underline{A}^{Hu} \\ \underline{A}^{Hu+1} \\ \vdots \\ \underbrace{\underline{A}^{Hp}}_{\underline{\Psi}} \end{bmatrix} \underline{x}(k) + \begin{bmatrix} \underline{B} \\ \vdots \\ \sum_{i=0}^{H_u-1} \underline{A}^i \underline{B} \\ \sum_{i=0}^{H_u} \underline{A}^i \underline{B} \\ \vdots \\ \underbrace{\sum_{i=0}^{H_p-1} \underline{A}^i \underline{B}}_{\underline{\Gamma}} \end{bmatrix} \underline{u}(k-1) +}$$

$$\underbrace{\begin{bmatrix} \underline{B} & \cdots & \underline{O}_{mxn} \\ \underline{A}\,\underline{B}+\underline{B} & \cdots & \underline{O}_{mxn} \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{H_u-1} \underline{A}^i \underline{B} & \cdots & \underline{B} \\ \sum_{i=0}^{H_u} \underline{A}^i \underline{B} & \cdots & \underline{A}\,\underline{B}+\underline{B} \\ \vdots & \vdots & \vdots \\ \sum_{i=0}^{H_p-1} \underline{A}^i \underline{B} & \cdots & \sum_{i=0}^{H_p-H_u} \underline{A}^i \underline{B} \end{bmatrix}}_{\underline{\Theta}} \begin{bmatrix} \underline{\hat{u}}(k) \\ \vdots \\ \underline{\hat{u}}(k+H_u-1) \end{bmatrix} \tag{2.62}$$

Sedangkan persamaan prediksi untuk keluaran $\underline{\hat{y}}(k+1|k)$ sistem dapat dinyatakan sebagai.

$$\begin{bmatrix} \underline{\hat{y}}(k+1|k) \\ \vdots \\ \underline{\hat{y}}(k+H_p|k) \end{bmatrix} = \underbrace{\begin{bmatrix} \underline{C} & \underline{O}_{mxn} & \cdots & \underline{O}_{mxn} \\ \underline{O}_{mxn} & \underline{C} & \cdots & \underline{O}_{mxn} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{O}_{mxn} & \underline{O}_{mxn} & \cdots & \underline{C} \end{bmatrix}}_{\underline{C}_y} = \begin{bmatrix} \underline{\hat{x}}(k+1|k) \\ \vdots \\ \underline{\hat{x}}(k+H_p|k) \end{bmatrix} \tag{2.63}$$

2.3.2 Fungsi Kriteria MPC

Perhitungan sinyal kontrol pada MPC dilakukan dengan meminimumkan suatu fungsi kriteria. Adapun fungsi kriteria yang

diminimumkan dalam algoritma MPC bisa dilihat pada persamaan (2.65).

$$V(k) = \sum_{i=1}^{Hp} \left\| \underline{\hat{y}}(k+i|k) - \underline{r}(k+i|k) \right\|^2_{\underline{Q}(i)} + \sum_{i=0}^{Hu-1} \left\| \underline{\Delta \hat{u}}(k+i|k) \right\|^2_{\underline{R}(i)} \quad (2.64)$$

Pemilihan penggunaan $\underline{\Delta \hat{u}}(k+i|k)$ pada fungsi kriteria bertujuan untuk meminimumkan perubahan sinyal kontrol yang masuk ke *plant*.

2.3.3 Strategi Kontrol MPC dengan *Constraints*

Pada setiap kendali proses, pasti terdapat batasan atau *constraints* pada amplitudo sinyal kendali. Selain itu, besarnya *slew rate* sinyal kendali juga dapat menjadi batasan. Pertidaksamaan *constraints* untuk amplitudo dan *slew rate* sinyal kendali bisa dilihat pada persamaan (2.65) dan (2.66).

$$\underline{FU}(k) \leq \underline{f} \quad (2.65)$$

$$\underline{E\Delta U}(k) \leq \underline{e} \quad (2.66)$$

Pada algoritma MPC, yang akan dihitung adalah nilai optimal perubahan sinyal kendali $\underline{\Delta U}(k)$ sehingga perlu untuk mengubah bentuk *constraints* yang belum mengandung $\underline{\Delta U}(k)$ menjadi bentuk *constraints* yang mengandung $\underline{\Delta U}(k)$.

Constraints yang berupa batasan nilai maksimum dan minimum sinyal kendali, pertidaksamaannya dapat ditulis sebagai berikut.

$$\underline{u}_{min} \leq \underline{u}(k) \leq \underline{u}_{max} \quad (2.67)$$

Pertidaksamaan (2.67) bisa ditulis dalam dua bentuk seperti (2.68) dan (2.69).

$$-\underline{u}(k) \leq -\underline{u}_{min} \quad (2.68)$$

$$\underline{u}(k) \leq \underline{u}_{max} \tag{2.69}$$

Pertidaksamaan (2.68) dan (2.69) bisa ditulis menjadi bentuk yang mengandung $\underline{\Delta U}(k)$ menjadi seperti persamaan (2.70) dan (2.71).

$$-\underline{F'\Delta U}(k) \leq \underline{u}(k) \leq -\underline{u}_{min} + \underline{F_1}u(k-1) \tag{2.70}$$

$$\underline{F'\Delta U}(k) \leq \underline{u}(k) \leq \underline{u}_{max} - \underline{F_1}u(k-1) \tag{2.71}$$

dengan

$$\underline{F'} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix} \tag{2.72}$$

$$\underline{F_1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{Hux\,1} \tag{2.73}$$

Pertidaksamaan (2.66), (2.70), dan (2.71) bisa disusun menjadi sebuah vektor matriks seperti yang ada pada pertidaksamaan (2.74).

$$\underbrace{\begin{bmatrix} -\underline{F'} \\ \underline{F'} \\ \underline{E} \end{bmatrix}}_{\underline{\Omega}} \underbrace{\underline{\Delta U}(k)}_{\underline{\delta}} \leq \underbrace{\begin{bmatrix} -\underline{u}_{min} + \underline{F_1}u(k-1) \\ \underline{u}_{max} + \underline{F_1}u(k-1) \\ \underline{e} \end{bmatrix}}_{\underline{\omega}} \tag{2.74}$$

Vektor matriks pada pertidaksamaan (2.74) selanjutnya digunakan pada perhitungan nilai optimal perubahan sinyal kontrol $\underline{\Delta U}(k)_{opt}$ menggunakan *quadratic programming*.

2.3.4 Metode *Quadratic Programming*

Permasalahan utama proses optimasi ini adalah meminimalkan fungsi kriteria:

$$\underline{\Delta U}^T(k)\ \underline{H\Delta U}(k) - \underline{\Delta U}^T(k)G \tag{2.75}$$

berdasarkan pada pertidaksamaan *constraints* (2.76)

$$\min_{\underline{\theta}}\ \frac{1}{2}\underline{\delta}^T\underline{\Phi\delta} + \underline{\emptyset\delta} \tag{2.76}$$

atau berdasarkan pada *constraints* (2.77)

$$\underline{\Omega\delta}\ <\ \underline{\omega} \tag{2.77}$$

Bentuk (2.75) dan (2.77) adalah masalah optimasi standar yang disebut sebagai permasalahan *quadratic programming*. Apabila ada bagian yang aktif di dalam himpunan *constraints* pada pertidaksamaan (2.77), maka bagian aktif tersebut akan membuat pertidaksamaan (2.77) menjadi persamaan (2.78).

$$\underline{\Omega}_{\mathrm{a}}\underline{\delta}_{\mathrm{a}} = \underline{\omega}_{\mathrm{a}} \tag{2.78}$$

Dengan matriks $\underline{\Omega}_{\mathrm{a}}$ adalah bagian yang aktif dari matriks pertidaksamaan (2.75). Persamaan (2.78) kemudian menjadi *constraints* dari fungsi kriteria pada pertidaksamaan (2.76).

Permasalahan optimasi pada pertidaksamaan (2.76) dengan subyek terhadap persamaan (2.78) dapat diselesaikan dengan teori *Lagrange* seperti yang terlihat pada persamaan (2.79).

$$\min_{\underline{\delta},\underline{\lambda}}\ \mathrm{L}(\underline{\delta},\underline{\lambda}) \tag{2.79}$$

dengan

$$L(\underline{\delta}, \underline{\lambda}) = \frac{1}{2}\underline{\delta}^{T}\underline{\Phi\delta} + \underline{\Phi\delta} + \underline{\lambda}(\underline{\Omega_a}\underline{\delta} - \underline{\omega_a}) \tag{2.80}$$

Selanjutnya dengan melakukan diferensiasi parsial terhadap $\underline{\delta}$ dan $\underline{\lambda}$ dari persamaan (2.80), maka didapatkan kondisi *Karush-Kuhn-Tucker* pada persamaan (2.81) dan (2.82) di bawah ini.

$$\nabla_{\underline{\delta}} L(\underline{\delta}, \underline{\lambda}) = \underline{\Phi\delta} + \underline{\emptyset} + \underline{\Omega_a}^{T}\lambda \tag{2.81}$$

$$\nabla_{\underline{\lambda}} L(\underline{\delta}, \underline{\lambda}) = \underline{\Omega_a}\underline{\delta} - \underline{\omega_a} \tag{2.82}$$

atau

$$\nabla_{\underline{\delta}} L(\underline{\delta}, \underline{\lambda}) = \begin{bmatrix} \underline{\Phi} & \underline{\Omega_a}^{T} \\ \underline{\Omega_a} & \underline{0} \end{bmatrix} \begin{bmatrix} \underline{\delta} \\ \underline{\lambda} \end{bmatrix} - \begin{bmatrix} -\underline{\emptyset} \\ \underline{\omega_a} \end{bmatrix} \tag{2.83}$$

Selanjutnya dengan membuat $\nabla_{\underline{\delta}} L(\underline{\delta}, \underline{\lambda}) = 0$, maka didapatkan solusi optimal untuk $\underline{\delta}$ dan $\underline{\lambda}$ seperti yang ada pada persamaan (2.84).

$$\begin{bmatrix} \underline{\delta} \\ \underline{\lambda} \end{bmatrix}_{\text{opt}} = \begin{bmatrix} \underline{\Phi} & \underline{\Omega_a}^{T} \\ \underline{\Omega_a} & \underline{0} \end{bmatrix}^{-1} \begin{bmatrix} -\underline{\emptyset} \\ \underline{\omega_a} \end{bmatrix} \tag{2.84}$$

Solusi pada *Quadratic Programming* (QP) pada kondisi normal menghasilkan nilai yang *feasible*, yaitu nilai yang memenuhi pertidaksamaan *constraints* yang ada dan dapat menghasilkan nilai fungsi kriteria minimum. Masalah yang paling sering muncul pada optimasi dengan *constraints* adalah solusi yang *infeasible*, dimana nilai yang dihasilkan tidak memenuhi pertidaksamaan *constraints* yang ada. QP *solver* akan menghentikan proses perhitungan jika terjadi solusi yang *infeasible* (Fahrudin, 2010).

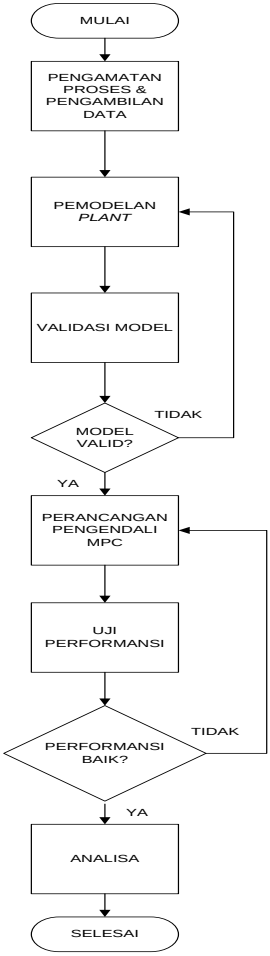
BAB III
METODOLOGI PENELITIAN

3.1 Alur Penelitian

Perancangan sistem pengendalian *steam* turbin dan generator di PT Geo Dipa Energi berbasis *Model Predictive Control* dilakukan melalui beberapa tahap tertentu. Pertama, mempelajari proses serta sistem pengendalian yang saat ini digunakan di PT Geo Dipa Energi. Kedua, membuat pemodelan *plant* menggunakan *simulink* Matlab R2009a. Setelah pemodelan selesai, maka langkah selanjutnya adalah melakukan validasi. Hal ini dilakukan dengan cara membandingkan respon yang dihasilkan model dengan respon pada *real plant*. Apabila model yang dibuat sudah valid, maka langkah selanjutnya adalah merangcang sistem pengendalian berbasis MPC untuk *steam turbine* dan generator. Tetapi, apabila model yang dibuat belum valid, maka yang harus dilakukan adalah membuat pemodelan lagi sampai sesuai dengan kondisi *real plant*. Mengingat bahwa variabel yang dikontrol lebih dari satu, maka sistem pengendalian ini tergolong MIMO (Multi Input Multi Output). Ini berarti sistem pengedalian yang dilakukan merupakan sistem pengedalian multivariabel. Pengendalian seperti ini bisa menggunakan MPC. Perancangan sistem pengendalian yang dilakukan bertujuan untuk mengoptimalkan variabel manipulasi pada kedua *plant*. Jika algoritma pengendalian MPC yang dibuat sudah bisa dijalankan, maka langkah selanjutnya adalah melakukan uji performansi terhadap model yang telah dirancang. Uji performansi yang dimaksud disini adalah uji kestabilan sistem dilihat pada *maximum overshoot*, *settling time*, dan *errorstady state*-nya. Pengujian dilakukan dengan memberikan gangguan berupa *current fault* pada *network*. Apabila respon yang diberikan oleh sistem sudah bagus, maka penelitian akan dilanjutkan pada tahap berikutnya. Tetapi, apabila respon yang ditunjukkan tidak bagus maka perancangan sistem pengendali MPC akan dilakukan lagi sampai diperoleh respon yang baik dari sistem yang telah dibuat.

Tahap akhir adalah melakukan analisa dan membuat kesimpulan atas penelitian yang telah dilakukan.

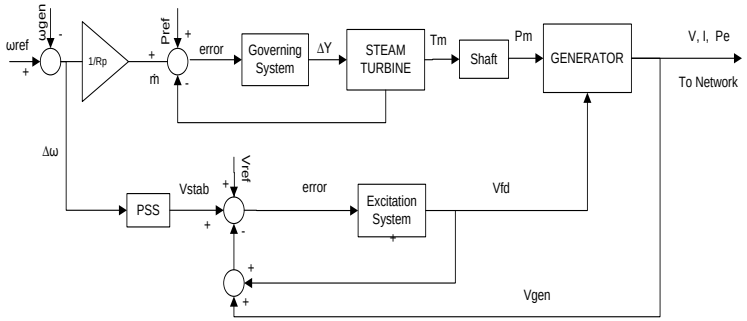
Tahap-tahap yang dilakukan dalam penelitian tugas akhir ini bisa direpresentasikan oleh *flowchartv*yang bisa dilihat pada gambar 3.1 di bawah ini.



Gambar 3.1*Flowchart* penelitian

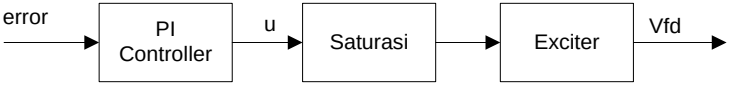
3.2 Pemodelan *Plant* di PT Geo Dipa Energi

Pemodelan *plant* dibuat berdasarkan pada persamaan-persamaan yang telah disebutkan pada tinjauan pustaka di bab 2. Pemodelan didasarkan pada kondisi *real plant* dan spesifikasi komponen-komponen yang digunakan. Perlu diketahui bahwa pemodelan sistem dibuat dengan menggunakan satuan pu (per unit) untuk mempemudah. Diagram blok pemodelan *plant* secara keseluruhan bisa dilihat pada gambar 3.2 di bawah ini.



(a)

(b)



(c)

Gambar 3.2 (a) Diagram blok pemodelan keseluruhan *plant* ; (b) *Governing subsystem* ; (c) *Excitation subSystem*

Adapun pemodelan masing-masing komponen yang ada pada gambar 3.2 bisa dilihat pada sub bab 3.2.1, 3.2.2, dan 3.2.3.

3.2.1 Pemodelan *Steam Turbine* & *Governing System*

Steam turbin dan *governing system* terdiri dari beberapa rangkaian.Untuk lebih jelasnya bisa dilihat di bawah ini.

- *Governing System*

Seperti yang sudah dijelaskan sebelumnya, *governing system* terdiri atas *controller*, *speed relay* dan servomotor. *Governor* yang digunakan berprinsip EHC (*Electric Hydroulic Converter*) yang diagram blok pengendaliannya bisa dilihat pada gambar 3.2 (a) dan (b). Seperti yang sudah dijelaskan sebelumnya, *controller* mendapat sinyal input dari selisih kecepatan putaran turbin, daya yang ingin dihasilkan dan *flow* yang merupakan *feedback* dari *steam* turbin. Selisih kecepatan turbin diperoleh dari pengurangan kecepatan nominal / referensi oleh kecepatan yang dihasilkan pada waktu tertentu. Hasilnya akan dibagi dengan *speed droop* (Rp) dari *steam* turbin. Sementara itu, daya referansi bisa diartikan juga sebagai *flow* referensi. Ini berarti untuk menghasilkan daya sebesar x pu sistem membutuhkan flow *steam* yang masuk *steam* turbin juga sebesar x pu. Misalnya, di PT Geo Dipa digunakan generator dengan kapasitas 75 MVA. Konsumsi *steam* yang dibutuhkan adalah untuk menghasilkan 1 MW adalah 7,5 ton/jam. Apabila ingin menghasilkan 15 MW, maka daya referensi yang digunakan adalah 0,2 pu. Nilai 0,2 pu mewakili 112,5 ton/jam dan mewakili 15 MW.

Controller yang digunakan untuk pemodelan awal adalah PI *controller* dengan nilai Kp sebesar 300 dan Ti sebesar 75 untuk *governing system*. Kecepatan putaran turbin dijaga pada 1 pu (3000 rpm). Nilai *speed drop* dari *steam* turbin diketahui sebesar 0,05, sedangkan *power active* yang saat ini ada di PT Geo Dipa sebesar 0,2 pu (setara 15 MW).

Speed relay dimodelkan dalam bentuk fungsi transfer seperti yang ada pada persamaan (2.4). *Time constant* dari *speed relay* diketahui sebesar 0,001 s, sehingga pemodelan untuk *speed relay* menjadi seperti persamaan (3.1) di bawah ini.

$$TF_{speed\ relay} = \frac{1}{0{,}001\ s+1} \quad (3.1)$$

Setelah melewati *speed relay*, sinyal *control* akan dilanjutkan menuju servo motor. *Time constant* dari servo motor (T_{sm}) diketahui sebesar 0,15 s. Saat ini, bukaan *governor valve* di PT Geo Dipa dibatasi sampai 50% saja, oleh karena itu pada pemodelan servo motor g*ate opening* diberi batasan dengan *lower limit sebesar* 0 pu dan *upper limit* sebesar 0,5 pu. Sedangkan *gate opening speed*-nya dibatasi -0,25 sampai 0,25 pu.

- *Steam Turbine*

Pemodelan *steam* turbin dibuat berdasarkan persamaan (2.6) dan (2.7) yang ada pada bab 2. *Time constant steam* turbin diketahui 0,02 s. *Steam pressure* diasumsikan 1 pu. *Steam* turbin yang digunakan adalah *single* turbin (*non reheater*), jadi nilai F (*turbine torque frictions*) yang digunakan adalah 1. Berdasarkan pada data-data tersebut, maka pemodelan *steam turbine* menjadi seperti permasaan (3.2) di bawah ini.

$$TF_{ST} = \frac{1}{0{,}02\ s+1} \quad (3.2)$$

- *Shatf*

Pemodelan *shaft* didasarkan pada persamaan (2.8) pada bab 2. Torsi mekanik yang digunakan output dari *steam* turbin, sedangkan nilai ω merupakan output dari generator.

Secara keseluruhan, pemodelan *steam* turbin dan *governing system* secara simulasi bisa dilihat pada lampiran C.

3.2.2 Pemodelan Generator Sinkron Tiga Fasa

Pemodelan generator sinkron dibuat menjadi dua bagian, yaitu mekanik dan elektrik. Pemodelan dibuat berdasarkan spesifikasi dari generator yang digunakan. Data lengkapnya bisa

dilihat pada lampiran A. Generator yang digunakan adalah generator sinkron tiga fasa dengan kapasitas 75 MVA dan 15 kV. Output dari generator akan dikirim ke jaringan PLN yang tegangannya mencapai 150 kV. Oleh karena itu, sebelum dikirim ke PLN output generator akan dilewatkan transformator *step up* 15/150 kV untuk menaikkan tegangannya terlebih dahulu.

Generator dimodelkan menggunakan satuan *per unit* (pu). Dalam simulasi, 1 pu mewakili 75 MVA, 15 kV, 2890 A, dan 3000 rpm. Adapun pemodelan generator sinkron dibuat berdasarkan persamaan (2.9) sampai (2.37) pada bab 2. Hasil pemodelan secara simulasi bisa dilihat pada lampiran D.

3.2.3 Pemodelan *Excitation System*

Seperti yang terlihat pada diagram blok pengendaliannya (gambar 2.8), diketahui bahwa input untuk *excitation system* berasal dari tegangan referensi, tegangan output generator, tegangan dari PSS. *Controller* yang digunakan pada model tersebut merupakan penguatan saja. Akan tetapi, dalam tugas akhir ini pemodelan *excitation system* dibuat sedikit berbeda dari model yang ada pada gambar 2.8. *Controller* dimodelkan dengan menggunakan PI *controller*. Selain itu, *feedback* dari *excitation system* sendiri juga ditambahkan sebagai sinyal input. Hal ini terlihat pada gambar 3.2 (a) dan (c).

Seperti yang telah dijelaskan, salah satu sinyal input yang digunakan dalam *excitation system* berasal dari PSS, Oleh karena itu sebelum membahas tentang *excitation system* sebaiknya membahas tentang PSS terlebih dahulu. Di dalam PSS terdapat beberapa komponen, yaitu : sensor, *washout*, kompensator dan *limiter*. Adapun pemodelannya dibuat berdasarkan diagram blok pada gambar 2.10. Ketiga komponen yang ada dalam diagram blok tersebut bisa dimodelkan sebagai berikut.

- Sensor

Pemodelannya dibuat berdasarkan pada persamaan (2.58). *Time constant* dari sensor yang digunakan adalah 0.003 s. Pemodelannya menjadi seperti persamaan (3.3).

$$TF_{vt} = \frac{1}{0.003\ s+1} \tag{3.3}$$

- *Washout*

Rangkaian *washout* dibuat dalam fungsi transfer berdasarkan persamaan (2.59). Δw diperoleh dari pengurangan w_{ref} oleh w_{gen}.Nilai tersebut berubah terhadap waku. *Washout time constant* diketahui sebesar 2 s, sehingga persamaannya menjadi seperti persamaan (3.4) di bawah ini.

$$TF_{washout} = \frac{2\ s}{2\ s+1} \tag{3.4}$$

- Kompensator dinamik

Pemodelan dibuat berdasarkan pada persamaan (2.60) dan (2.61). Nilai T_1, T_2,T_3, dan T_4 masing-masing 0,005 s, 0,0025 s, 0,3 s, dan 0,15 s, maka fungsi transfer untuk kompensator dinamik menjadi seperti persamaan (3.5) dan (3.6) di bawah ini.

$$TF_{lead-lag\_1} = \frac{0{,}005\ s+1}{0{,}0025\ s+1} \tag{3.5}$$

$$TF_{lead-lag\_2} = \frac{0{,}3\ s+1}{0{,}15\ s+1} \tag{3.6}$$

- *Limiter*

Limiter yang digunakan pada *simulink* adalah fungsi saturasi dengan *upper limit* sebesar 0,2 pu dan *lower limit* sebesar -0,2pu.

Output dari PSS akan diakumulsi dengan V_{ref}, V_{gen}, dan *feedback* dari *excitation system*. Hasilnya akan menjadi input bagi *controller*. Adapun parameter *controller* yang digunakan adalah Kp sebesar 50 dan Ti sebesar 0,1.

Sebelum diteruskan ke *exciter*, sinyal kontrol yang dihasilkan *controller* perlu diberi batasan untuk mencegah osilasi yang berlebihan. Oleh karena itu, setelah melewati PI *controller*, sinyal control dilewatkan ke *limiter* terlebih dahulu. Dalam pemodelan ini, sinyal di batasi dengan *upper limit* sebesar 10 pu dan *lower limit* sebesar -10 pu.

- *Exciter*

Setelah melewati *limiter*, selanjutnya sinyal kontrol akan diteruskan menuju *exciter*. Pemodelan *exciter* dibuat berdasarkan persamaan (2.55). *Exciter gain* yang digunakan adalah 1, sedangkan *exciter time constant* diketahui 0.001 s. Maka, fungsi transfernya menjadi seperti persamaan (3.7) di bawah ini.

$$TF_{Exciter} = \frac{1}{0{,}001\ s+1} \tag{3.7}$$

- *Damper*

Pemodelan *damper* untuk keluaran sistem eksitasi dibuat berdasarkan fungsi transfer pada persamaan (2.56). Dengan nilai K_f dan T_f masing-masing 0,001 dan 0,1 s, sehingga fungsi transfernya menjadi seperti persamaan (3.8) di bawah ini.

$$TF_{Vfd.damper} = \frac{0{,}001\ s}{0{,}1\ s+1} \tag{3.8}$$

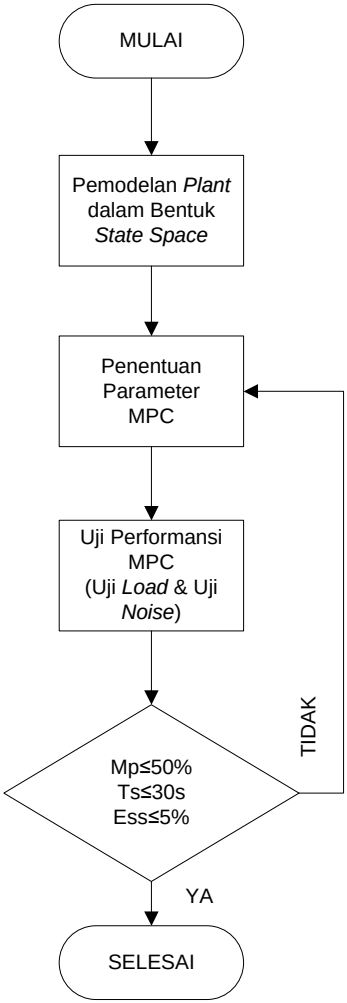
Sedangkan pemodelan *damper* untuk tegangan generator di modelkan berdasarkan persamaan (2.57) dengan nilai *damper time constant* sebesar 0,0022 s menjadi seperti persamaan (3.9).

$$TF_{damper} = \frac{1}{0{,}0022\ s+1} \tag{3.9}$$

Berdasarkan pada pemodelan masing-masing komponen yang dilakukan di atas, maka pemodelan *excitation system* secara keseluruhan menjadi seperti yang ada pada lampiran D.

3.3 Perancangan Algoritma MPC

Perancangan algoritma kontrol prediktif berbasis MPC bisa ditulis dalam bentuk *flowchart* seperti yang terlihat pada gambar 3.3.

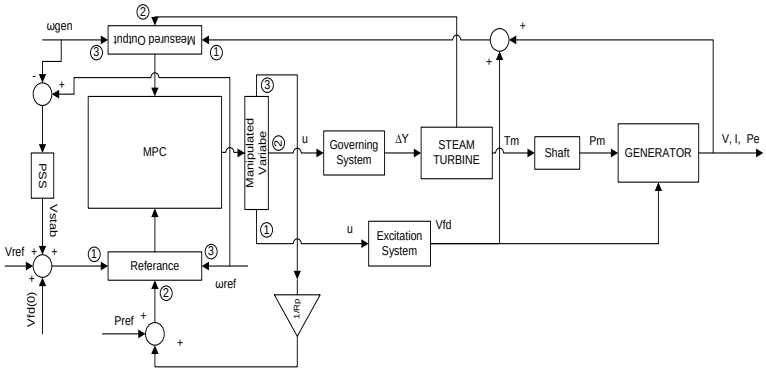


Gambar 3.3 *Flowchart* pembuatan algoritma MPC

Berdasarkan gambar 3.3 diketahui bahwa perancangan pengendli MPC dimulai dengan linierisasi *plant*. Metode linierisasi yang digunakan adalah *gradient descent with elimination* dengan algoritma *block-by block analytic*. Proses linierisasi dilakukan melalui analisa setiap blok di dalam model *plant* dengan mendefinisikan input dan output *plant* terlebih dahulu. Hasil dari linierisasi *plant* yang telah dilakukan adalah model linier dari *plant* dalam bentuk *state-space*. Model inilah yang akan digunakan oleh pengendali MPC untuk memprediksi output pada horizon tertentu.

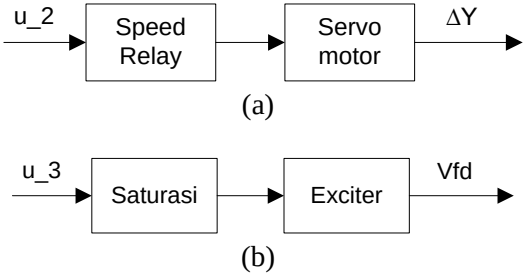
Setelah model proses, hal yang perlu dilakukan adalah optimasi fungsi kriteria dari *plant.* Fungsi kriteria *plant* merupakan penjumlahan fungsi kuadratik dari *error* (selisih antara output prediksi dengan *set point* atau *reference trajectory*) dan selisih perubahan sinyal kontrol. Fungsi kriteria tersebut akan dioptimasi secara terus-menerus sepanjang horizon prediksi. Dengan begitu *trajectory error* (selisih output prediksi dengan *reference trajectory*) dan perubahan sinyal kontrol (Δu) yang dibutuhkan akan semakin kecil. Hal ini membuat output sistem akan dapat mencapai *set point* dengan perubahan sinyal kontrol yang optimal.

Plant yang dimodelkan sama seperti sebelumnya. Perbedaannya hanya pada struktur pengendalian dan *controller* yang digunakan. Sebelumnya, *plant* dimodelkan menggunakan PI *controller* sesuai yang digunakan pada *real plant* yang diagram blok pengendaliaannya bisa dilihat pada gambar 3.2. Kemudian, *plant* dimodelkan lagi menggunakan MPC sebagai pengendalinya. Adapun diagram blok pengendaliaanya bisa dilihat pada gambar 3.4.



Gambar 3.4 Diagram blok pengendalian *plant* dengan MPC

Sedangkan *governing subsystem* dan *excitation subsystem* pada gambar 3.4 bisa dilihat pada gambar 3.5.



Gambar 3.5 (a) *Governing subsystem* ; (b) *Excitation subsystem*

Berdasarkan diagram blok yang ditunjukkan oleh gambar 3.4, bisa dilihat bahwa input MPC terdiri dari sinyal referensi, *measured output* dan *manipulated variable*. Referensi merupakan nilai output *plant* yang diinginkan. *Measured* output merupakan nilai output *plant* yang terukur. Sedangkan, *manipulated variable* merupakan sinyal kontrol yang dihasilkan MPC dan menjadi input bagi aktuator. Pada diagram blok terlihat bahwa terdapat tiga *port* untuk masing-masing input MPC. Hal ini dikarenakan

ada tiga variabel yang dikendalikan. *Port* 1 untuk tegangan generator, *port* 2 untuk daya aktif dan *port* 3 untuk kecepatan putaran *steam* turbin. Karena sinyal masukan yang digunakan ada tiga, maka *manipulated variable* yang dihasilkan juga ada tiga. Hal ini terlihat berbeda jika dibandingkan dengan PI *controller* karena pada pemodelan PI *controller* hanya dihasilkan dua sinyal kontrol.

Perbedaan antara PI dan MPC terletak pada input untuk *governing system*. Pada pemodelan PI *controller* diketahui bahwa perubahan kecepatan *steam* turbin dibagi dengan koefision *speed droop*-nya. Setelah itu, diakumulasi dengan daya referensi dan dikurangi *flow feedback* dari *steam* turbin. Hasilnya berupa sinyal *error* yang menjadi input bagi PI *controller* yang akan menghasilkan sinyal kontrol untuk aktuator. Ini berarti sinyal kontrol yang dihasilkan adalah satu.

Pada MPC, sinyal kontrol untuk *governing system* diperoleh setelah mengalami dua kali pengendalian. Perubahan kecepatan putaran *steam* turbin menjadi sinyal error bagi MPC yang selanjutnya diproses sehingga mengasilkan sinyal kontrol yang dilewatkan pada *port* 3. Sinyal kontrol tersebut tidak langsung dilanjutkan menuju aktuator, melainkan dikonversi dulu menjadi *flow* dengan cara membaginya dengan koefisien *speed droop*. Hasilnya diakumulasi dengan daya referensi dan menjadi input referensi bagi MPC di *port* 2. *Measured* output yang menjadi input di *port* 2 merupakan *flow steam* yang ada pada *steam* turbin. Dengan adanya sinyal referensi dan *measured* output tersebut dihasilkan sinyal *error* dan dalam MPC sinyal tersebut akan dirubah menjadi sinyal kontrol yang dilewatkan pada *port* 2. Sinyal kontrol pada *port* 2 inilah yang akan menjadi input bagi aktuator pada *governing system*. Bisa dikatakan bahwa MPC yang digunakan dimodelkan dalam bentuk *cascade*.

Apabila dilihat dari sistem eksitasinya, pemodelan pengendali PI maupun MPC mempunyai struktur yang hampir sama. Pada PI, sinyal input bagi *controller* merupakan sinyal *error* yang diperoleh dari penjumlahan tegangan referensi, *initial field voltage*, dan tegangan PSS, lalu hasilnya dikurangi dengan

tegangan output tegangan generator dan eksitasi. Sedangkan, pada MPC inputnya adalah sinyal referensi (penjumlahan tegangan referensi, tegangan dari PSS, dan *field voltage* pada *initial condition*) dan *measured output* (tegangan output generator dan eksitasi).

Dalam perancangan alogaritma MPC, terdapat beberapa parameter yang digunakan untuk mengahsilkan sinyal *control*. Adapun parameter-parameter tersebut adalah sebagai berikut.

- *Prediction horizon*
  Prediction horizon merupakan jauhnya jangka waktu ke depan yang digunakan oleh *controller* untuk memprediksi output *plant*.Prediction horizon yang digunakan saat simulasi adalah 3.
- *Control horizon*
  Control horizon merupakan jauhnya jangka waktu ke depan yang digunakan *controller* untuk menghitung sinyal *control*.*Control horizon* yang digunakan saat simulasi adalah 1.
- *Constraint*
  Constraint merupakan suatu rentang tertentu yang digunakan untuk membatasi *manipulated variable*. Adapun *constraint* yang digunakan adalah -0,1 sampai 0,1 untuk sinyal pertama (kecepatan putaran *steam* turbin), -0,2 sampai 0,2 untuk sinyal kedua (*flowrate steam*), dan -2 sampai 2 untuk sinyal ketiga (tegangan generator).
- *Weight* output
  Weight output digunakan untuk menentukan akurasi dari setiap ouput yang harus *tracking* pada *set point* tertentu. Weight output yang digunakan saat simulasi adalah 0,040762.
- *Weight rate*
  Weight rate berfungsi untuk memperkecil *weight controller*. *Weight rate* yang digunakan saat simulasi adalah 0,073598.

Penentuan parameter *prediction* dan *control horizon* dalam perancangan ini dilakukan dengan metode *tuning trial and error*. Setelah melakukan *tuning controller*, langkah selanjutnya adalah melakukan simulasi hasil rancangan *controller* tersebut untuk mengetahui performansi *controller* dalam menjaga nilai output tetap sesuai pada *setpoint*. Selama simulasi akan dilakukan beberapa pengujian. Apabila hasil uji performansi sudah baik, maka pembuatan algoritma MPC telah berhasil. Jika performansi pengendaliannya masih tidak bagus, maka penentuan parameter *controller* MPC harus diulangi lagi sampai didapatkan performansi pengendalian yang baik.

3.4 Uji Performansi Sistem

Uji performansi dilakukan untuk mengetahui performansi dari sistem pengendalian yang telah dirancang. Pengujian dilakukan pada kedua *controller* dengan memberikan perlakuan yang sama. Hasilnya dibandingkan satu sama lain sehingga bisa diketahui sistem pengendali yang memberi respon lebih baik. Adapun pengujian yang dilakukan adalah sebagai berikut.

- Uji *closed loop*
  Uji ini dilakukan untuk mengetahui bahwa sistem memang bisa mengendalikan variabel proses pada kondisi normal.
- Uji *gangguan*
  Uji ini dilakukan untuk mengetahui kemampuan sistem pengendalian dalam mengendalikan variabel proses sesuai *set point* walaupun ada gangguan atau *noise* dari luar. Sinyal *noise* yang digunakan berupa *three phase fault* yang dipasang pada sirkuit menuju *network*. *Fault resistance* yang digunakan 0,001Ω. *Ground resistance fault* sebesar 0,001 Ω. *Snubbers resistance* sebesar $1x10^6$. *Transmition line* pada detik ke-10 s sampai 10,01 s. Ini berarti *fault* akan terjadi sekitar 0,01 s.

BAB IV
ANALISA DATA DAN PEMBAHASAN

4.1 Validasi Model

Pemodelan *steam* turbin dan generator yang telah dibuat divalidasi untuk memastikan bahwa model yang telah dibuat sesuai dengan kondisi *real plant* di PT Geo Dipa. Data proses dari plant yang digunakan sebagai kompensator dalam validasi adalah data pada bulan maret 2012 (Lampiran E). Adapun data yang digunakan adalah kecepatan putaran *steam* turbin, tegangan generator dan *power active* yang dihasilkan. Perbandingan nilai antara *real plant* dan hasil pemodelan bisa dilihat pada tabel 4.1 di bawah ini.

Tabel 4.1 Perbandingan data proses *real plant* dengan model

Variabel	*Real Plant*	Model
Speed steam turbin	3000,668 rpm	3000 rpm
*Tegangan g*enerator	14962,76 V	15000 V
Power Active	15,054MW	14,9989 MW

Data yang digunakan dalam *real plant* merupakan *mean* dari data proses selama maret 2012, sedangkan data pada model merupakan data pada kondisi *steady state* yang diperoleh ketika model disimulasikan. Berdasarkan data yang ada pada tabel 4.1 bisa dikatakan bahwa model yang dibuat sudah valid karena mendekati kondisi *real plant*. Selisih antara *real plant* dan model pada pengendalian putaran turbin hanya 0,668 rpm. Pengendalian tegangan generator mempunyai selisih 37,24 volt. Sedangkan, *power active* mempunyai selisih 55,1 kW.

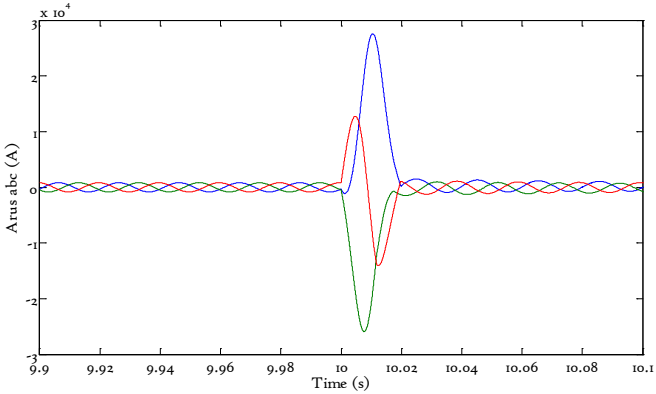
4.2 Hasil Simulasi Pemodelan dengan Uji *Current Fault*

Model yang telah dirancang disimulasikan menggunakan Matlab. Simulasi ini dilakukan dengan tujuan untuk mengetahui respon sistem yang dihasilkan. Kedua model, baik yang

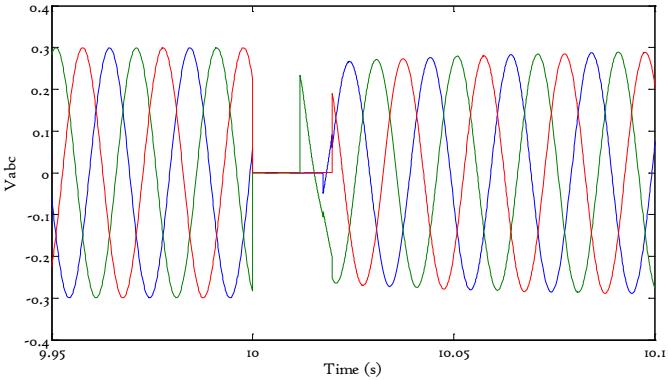
menggunakan PI *controller* ataupun yang menggunakan MPC diberi perlakuan sama. Terdapat tiga parameter yang diamati, yaitu : nilai penyimpangan terbesar dari *set point* yang akan dicapai (*maximum overshoot* / M_p), waktu yang dibutuhkan untuk mencapai nilai *steady* (*settling time* / T_s), dan *error steady state* (E_{ss}).

Parameter yang digunakan pada PI *controller* untuk *governing system* adalah Kp sebesar 300 dan Ti sebesar 75. Sedangkan untuk *excitation system* adalah Kp sebesar 50 dan Ti sebesar 0,1. Parameter yang digunakan pada MPC adalah *prediction horizon* 3, *control horizon* 1, *weight output* 0,040762, *weight rate* 0,073598, *constraints* untuk *controller* 1 (kecepatan putaran turbin) adalah -0,1 sampai 0,1, *constraint* untuk *controller* 2 (*flow* / daya aktif) adalah -0,5 sampai 0,5, dan *constraint* untuk *controller* 3 (tegangan generator) adalah -2 sampai 2.

Seperti yang sudah dijelaskan pada bab 3, simulasi dilakukan dengan memberikan *current fault* pada jaringan selama 0,1 sekon. *Current fault* dilakukan pada detik ke-10 sampai 10,1. Gangguan ini bisa dilihat pada gambar 4.1 dan 4.2.



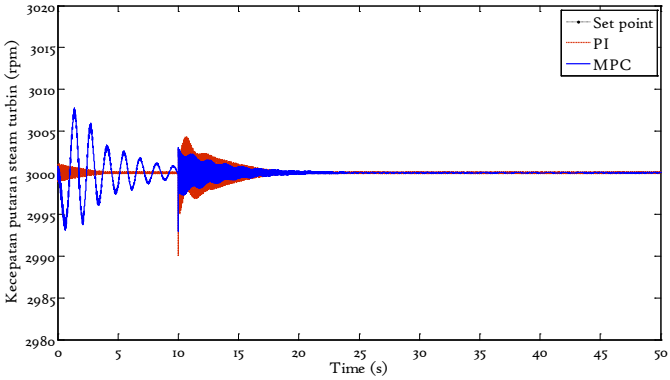
Gambar 4.1 I_abc ketika ada *current fault*



Gambar 4.2 V_abc ketika ada *current fault*

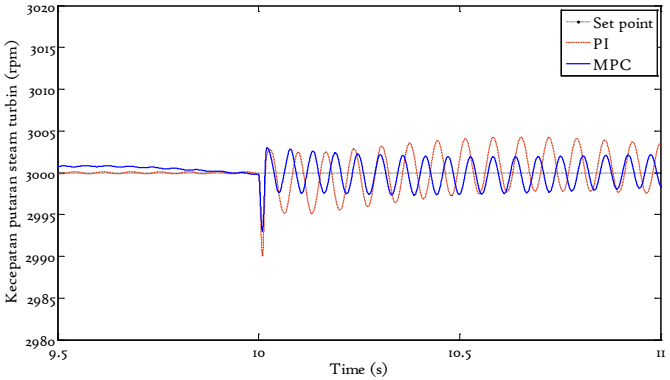
Adanya *current fault* pada jaringan tiga fasa membuat arus yang mengalir pada line tersebut terganggu dan tidak stabil. Adanya gangguan pada arus pasti memberi pengaruh pada tegangannya. Seperti yang terlihat pada gambar 4.1 dan 4.2, kegagalan arus menyebabkan kegagalan pada tegangan juga. Apabila tegangan pada line tiga fasa (Vabc) terganggu, maka tegangan pada generator (Vd & Vq) juga terganggu. Hal ini bisa menyebabkan sistem tidak stabil dan bisa mengalami kerusakan. Untuk menjaga agar sistem tetap bisa menjaga variabel proses tetap sesuai set point, maka digunakanlah suatu controller.

Hasil rancangan sistem pengendalian dengan PI ataupun MPC disimulasikan selama 50 s. Simulasi dilakukan dengan memberikan *initial condition* pada sistem untuk menghasilkan 15 MW. Respon yang diamati adalah kecepatan putaran *steam* turbin, tegangan generator, dan *power active*. Hasil simulasi yang telah dilakukan disajikan dalam bentuk grafik respon sistem seperti yang terlihat pada gambar 4.3 sampai 4.8.



Gambar 4.3 Respon kecepatan putaran *steam* turbin

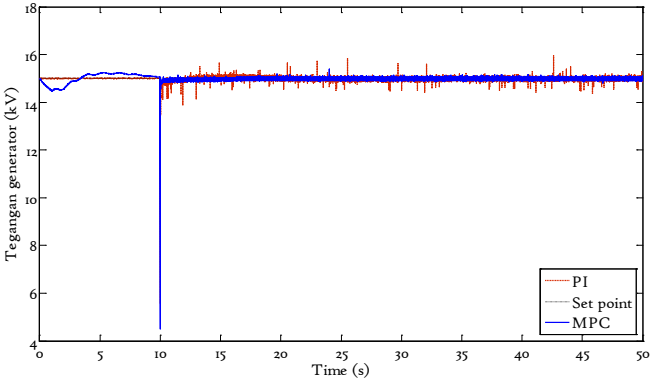
Untuk memperjelas osilasi yang terjadi pada kecepatan putaran *steam* turbin, maka respon yang ditunjukkan pada gambar 4.3 diperbesar lagi sehingga menjadi seperti gambar 4.4.



Gambar 4.4 Osilasi kecepatan putaran *steam* turbin

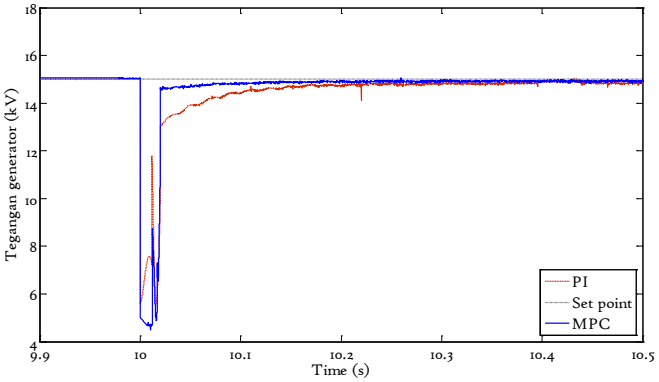
Gambar 4.3 merupakan respon yang terjadi pada kecepatan putaran *steam* turbin. Seperti yang telah diketahui, kecepatan putaran turbin dijaga *constant* pada 3000 rpm. Berdasarkan pada simulasi yang telah dilakukan menggunakan PI *controller*,

diketahui bahwa pada awal simulasi sistem berada pada *initial condition* lalu mengalami sedikit osilasi yang menyebabkan adanya *overshoot* sebesar $6{,}67 \times 10^{-2}$%. *Overshoot* yang dihasilkan oleh MPC pada awal simulasi lebih tinggi dari PI, yaitu : 0,27%. Kestabilan kedua *controller* bisa dilihat pada detik ke-10 ketika ada gangguan arus pada jaringan. Gambar 4.3 menunjukkan bahwa pengendali MPC bisa mengendalikan proses lebih baik dibandingkan MPC. Hal ini bisa dilihat dari *maximum overshoot* yang dihasilkan. Pada MPC timbul *maximum overshoot* sebesar 0,1%, sedangkan PI membuat sistem mengalami *overshoot* lebih tinggi yaitu sebesar 0,17%. *Settling time* untuk PI *controller* adalah 7 s, sedangkan untuk MPC adalah 13 s. Ess untuk PI controller adalah $3{,}33 \times 10^{-3}$ %, sedangkan Ess untuk MPC adalah $3{,}33 \times 10^{-4}$ %.



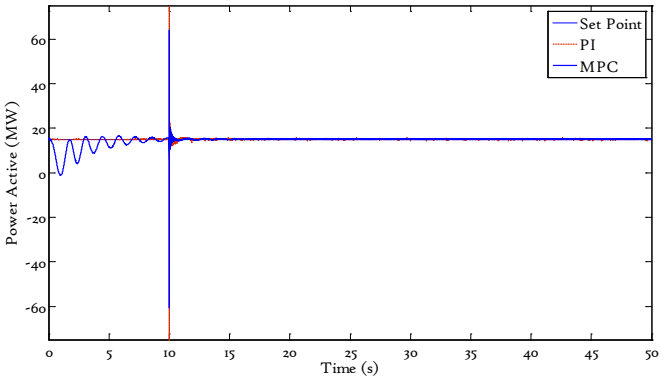
Gambar 4.5 Respon tegangan generator

Untuk memperjelas osilasi yang terjadi pada tegangan generator, maka respon yang ditunjukkan pada gambar 4.5 diperbesar lagi sehingga menjadi seperti gambar 4.6.



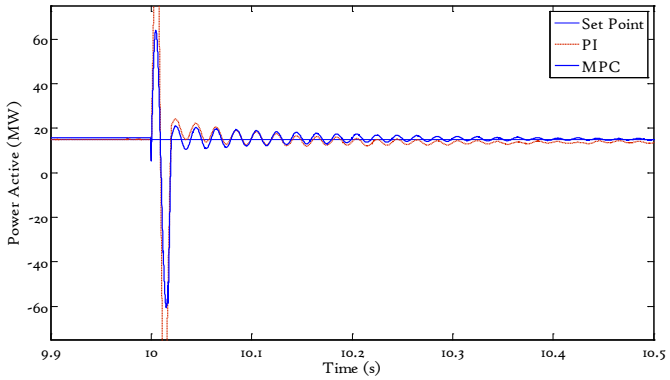
Gambar 4.6 Osilasi pada tegangan generator

Tegangan output generator yang diharapkan adalah 15 kV. Berdasarkan simulasi yang telah dilakukan dengan PI *controller* dan MPC, diketahui bahwa sistem mengalami ketidakstabilan akibat adanya *current fault*. MPC bisa maupun PI mencapai *steady* kembali, hanya saja respon yang ditunjukkan oleh PI *controller* cenderung kurang stabil. Mp, Ts, dan Ess yang diperoleh bisa dilihat pada tabel 4.2



Gambar 4.7 Respon *power active*

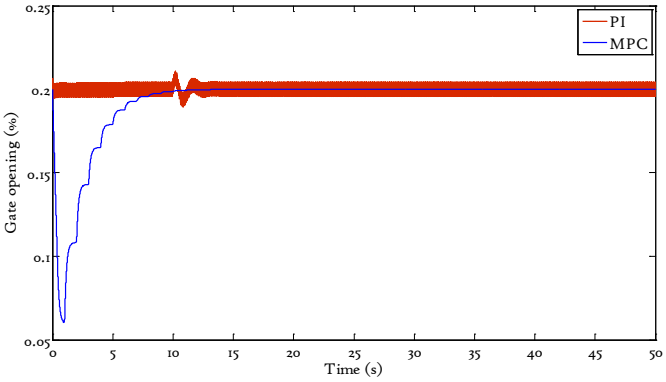
Untuk memperjelas osilasi yang terjadi pada *power active*, maka respon yang ditunjukkan pada gambar 4.7 diperbesar lagi sehingga menjadi seperti gambar 4.8.



Gambar 4.8 Osilasi *power active*

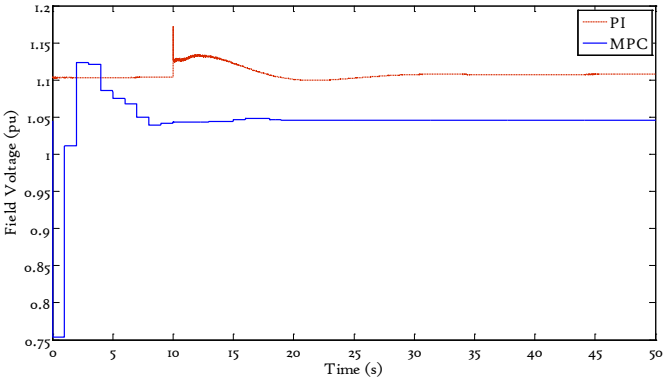
Power active / daya yang dihasilkan oleh PI ataupun MPC hampir sama dengan respon pada tegangan generator. Apabila dilihat pada *maximum overshoot*, *settling time*, dan *error steady state*-nya, MPC memberikan respon yang lebih baik dibandigkan PI *controller*. Nilai Mp, Ts, dan Ess yang dihasilkan bisa dilihat pada tabel 4.2.

Selain ketiga output variabel di atas, *manipulated variable* pada sistem juga perlu diamati. Hal ini dilakukan untuk mengetahui bahwa sinyal kontrol yang diberikan berpengaruh pada *plant*. *Manipulated variable* yang diperoleh ketika simulasi dijadikan input bagi aktuator. Dalam hal ini, output dari aktuator adalah *gate opening* dan *field voltage*. *Gate opening* dan *field voltage* bergantung pada sinyal kontrol yang diberikan, oleh karena itu sinyal kontrol yang dihasilkan oleh *controller* bisa diamati dari respon pada *gate opening* dan *field voltage*. Adapun responnya bisa dilihat pada gambar 4.9 dan 4.10.



Gambar 4.9 *Gate opening*

Gambar 4.9 merupakan output *governing sistem* yang dihasilkan oleh PI *controller* dan MPC. Pada awal simulasi, PI memberikan respon sesuai *initial condition*. Hal ini berbeda dengan MPC yang tidak memberikan respon sesuai *initial condition* pada awal simulasi. Pada algoritma MPC, *plant* dilinierisasi dan dibuat dalam bentuk *state space*. Ini menyebabkan MPC bekerja pada titik nol dan tidak sesuai *initial condition*. MPC baru bisa memberikan respon sesuai *initial condition* setelah berjalan selama 8 s. Pada detik ke-10, sistem diberi gangguan berupa *current fault*. Adanya gangguan ini membuat sistem menjadi tidak stabil. Pengendali PI ataupun MPC bisa membuat sistem kembali pada kondisi *steady* sesuai *set point*. Tetapi, apabila diperhatikan MPC memberikan sinyal kontrol yang lebih stabil dibandingkan PI *controller*. Sinyal kontrol yang lebih stabil memberikan respon yang lebih stabil juga. Hal ini terlihat pada gambar 4.3 dan 4.8.



Gambar 4.10 *Field voltage*

Gambar 4.7 merupakan output dari sistem eksitasi. Sama seperti sebelumnya, pada awal simulasi PI *controller* memberikan respon sesuai *initial condition* yang telah disetting. Sedangkan, MPC mengalami penurunan terlebih dahulu dan baru bisa sesuai *initial condition* ketika sudah berjalan 10 s. Ketika ada *current fault* pada detik ke-10, PI ataupun MPC sempat mengalami kondisi yang tidak stabil. Sinyal kontrol yang dihasilkan PI berada pada kondisi *steady* kembali setelah 20 s. Sedangkan, MPC hanya memerlukan 8 s untuk bisa membuat sistem kembali pada kondisi *steady* sesuai *set point*. Apabila diperhatikan pada gambar 4.5, bisa dilihat bahwa respon yang dihasilkan MPC lebih stabil. Hal ini dikarenakan field voltage yang dihasilkan lebih stabil dan sesuai. Untuk menghasilkan *power active* sebesar 15 MW dibutuhkan input *field voltage* sebesar 1,05 pu.

Pada kondisi awal, MPC memang tidak bisa langsung mencapai kondisi ideal, tetapi apabila dilihat dari kestabilannya MPC mempunyai keunggulan dibandingkan PI *controller* baik dari segi *setling time* ataupun *maximum overshoot*-nya.

4.3 Hasil Perbandingan Respon Sistem Pengendalian

Pada sub-bab ini dilakukan perbandingan respon sistem menggunakan kontroller PI dan kontroller MPC yang telah dirancang. Adapun perbandingannya dilihat dari segi M_p, T_s, dan E_{ss} yang datanya ditampilkan dalam bentuk tabel seperti yang terlihat pada tabel 4.2.

Tabel 4.2 Perbandingan Mp, Ts, & Ess

Variabel	**Controller**	**Kondisi normal**		
		Mp **(%)**	$\boldsymbol{\tau_s}$ **(detik)**	***Ess*** **(%)**
Speed *Steam* Turbin	PI	0,17	7	$3{,}33\text{x}10^{-3}$ %
	MPC	0,1	13	$3{,}33\text{x}10^{-4}$ %
Tegangan generator	PI	-	0,4	0,4
	MPC	-	0,3	0,33
Power active	PI	200	1,5	1,87%
	MPC	100	0,6	0,33%

Secara keseluruhan, berdasarkan hasil simulasi yang telah dilakukan bisa dilihat bahwa MPC mempunyai performansi pengendali yang lebih baik dibandingkan PI *controller*. Selain itu, MPC lebih efisien karena hanya perlu satu *controller* untuk mengendalikan sistem multivariabel. Sedangkan, PI *controller* dipasang pada masing-masing *plant*. Dalam hal ini, karena variabel yang dikontrol ada dua, maka PI *controller* pun juga dipasang dua. Perbedaan PI dan MPC yang paling mencolok terlihat ketika ada *current fault*. MPC mampu mengendalikan variabel proses sesuai *set point* lebih stabil dan lebih cepat dibandingkan PI *controller*.

BAB V
KESIMPULAN

Berdasarkan pada simulasi yng telah dilakukan dapat di ambil beberapa kesimpulan sebagai berikut.

- Sistem pengendalian MPC dimodelkan dengan menggunakan parameter *prediction horizon* 3, *control horizon* 1, *weight* output 0,040762, *weight rate* 0,073598, *constraints controller inner loop* adalah -0.1 sampai 0,1, *constraint controller outer loop* adalah -0.5 sampai 0,5, dan *constraint controller* tegangan adalah -2 sampai 2.
- Berdasarkan pada respon yang ditunjukkan ketika dilakukan uji performansi (*current fault*), MPC menunjukkan respon yang lebih baik dibandingkan PI *controller*.
- Pengendalian kecepatan putaran turbin dengan MPC memberikan respon yang mempunyai Mp 0,1%, Ts 13 s, dan Ess $3{,}33 \times 10^{-4}$%.
- Pengendalian tegangan terminal generator dengan MPC memberikan respon yang mempunyai Ts 0,3 s, dan Ess 0,33%.
- Daya aktif yang dihasilkan mempunyai nilai Mp 100 %, Ts 0,6 s, dan Ess 0,33%.

"*Halaman ini sengaja dikosongkan.*"

LAMPIRAN A

SPESIFIKASI *STEAM* TURBIN & GENERATOR

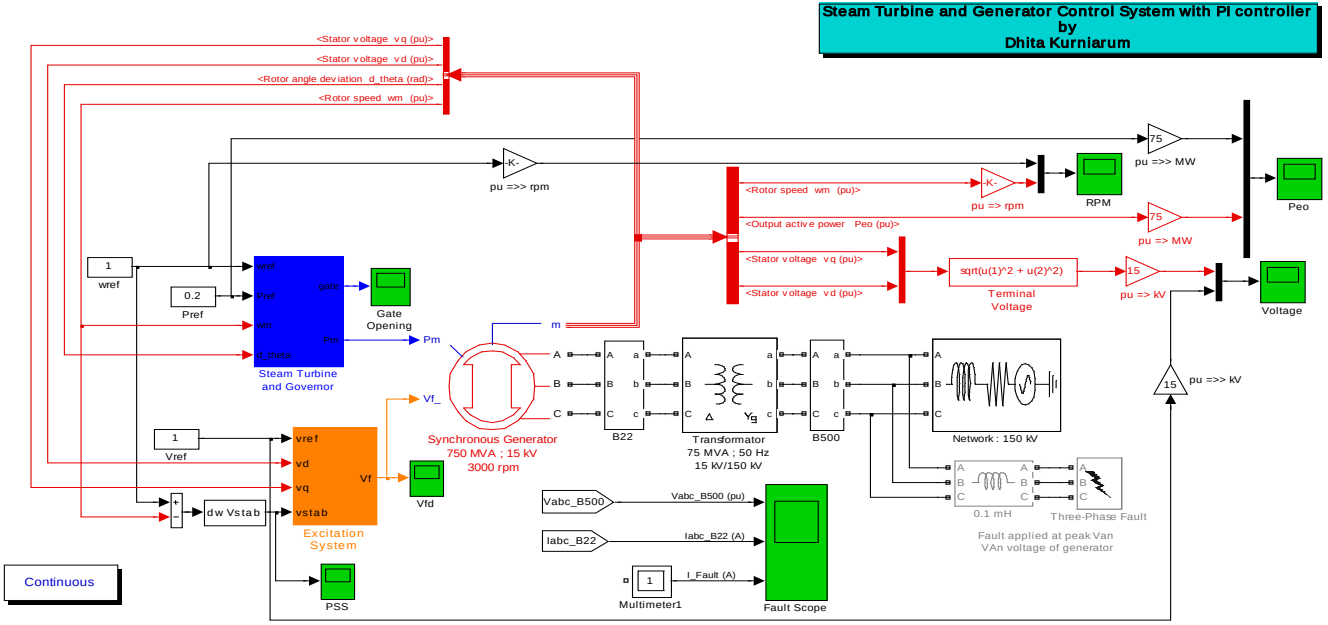
Parameter	*Value*
***Syncronous* Generator Per Unit *Standart* Parameter**	
P	75 MVA
V	15 kV
f	50 Hz
rpm	3000
Fasa	3
Power Factor	0,8
X_d	1.95 pu
X'_d	0.25 pu
X''_d	0.15 pu
X_q	1.38 pu
X''_q	0.2085 pu
X_l	0.1557 pu
T'_d	0.76 s
T''_d	0.03 s
T''_{qo}	0.1 s
R_s	0.0015 pu
H	3.2 s
p	2
Steam Turbine Impuls-Reaction	
Manufacture	Ansaldo, Italia
Year	1990
P	60 MW
rpm	3000
Pressure	0,081 ATA
Stages	7+7
H	3,2
D	0,5
K	83.47
F	1

"*Halaman ini sengaja dikosongkan.*"

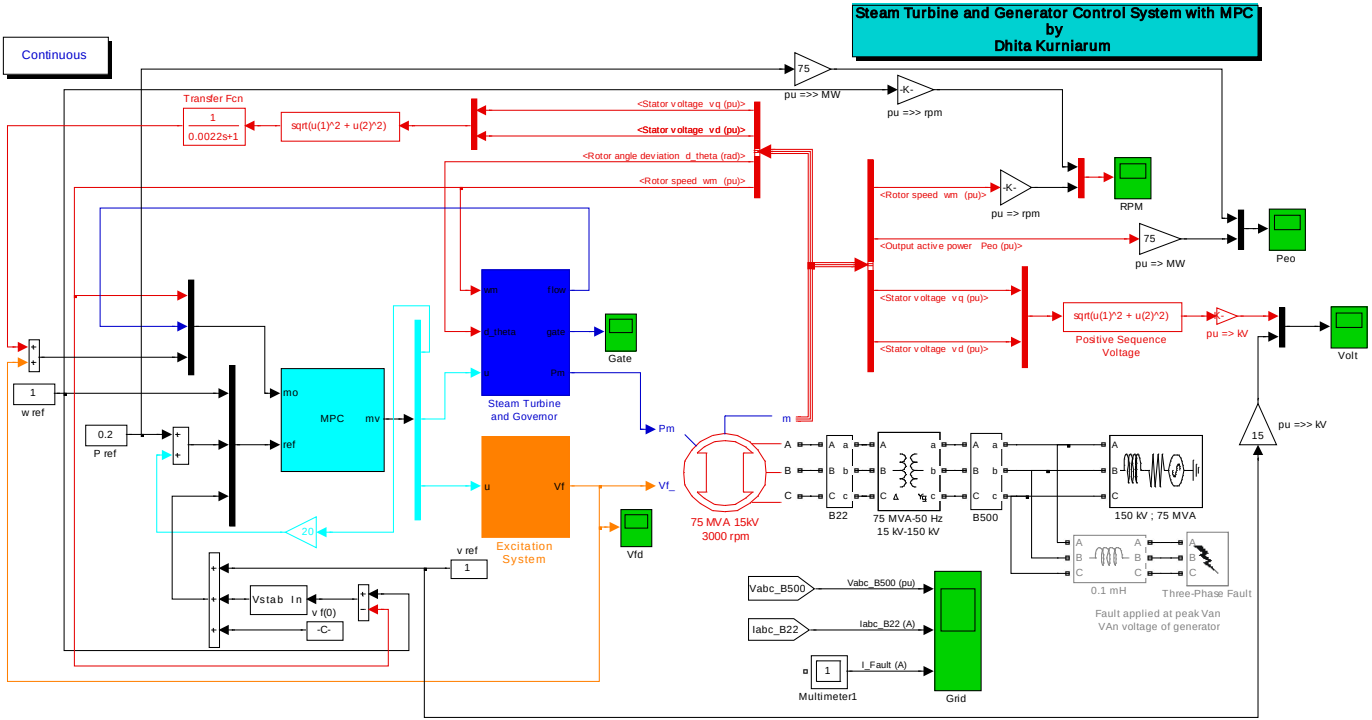
LAMPIRAN B

HASIL PEMODELAN *PLANT*

1. HASIL PEMODELAN *STEAM* TURBIN-GENERATOR DENGAN PI *CONTROLLER*

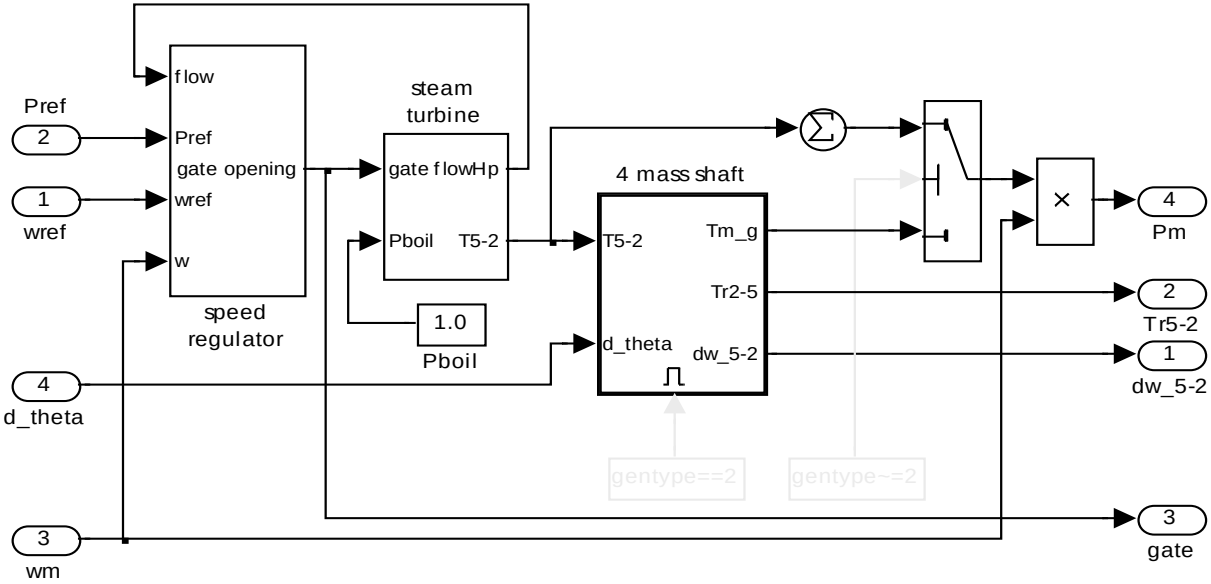


2. HASIL PEMODELAN *STEAM* TURBIN-GENERATOR DENGAN MPC



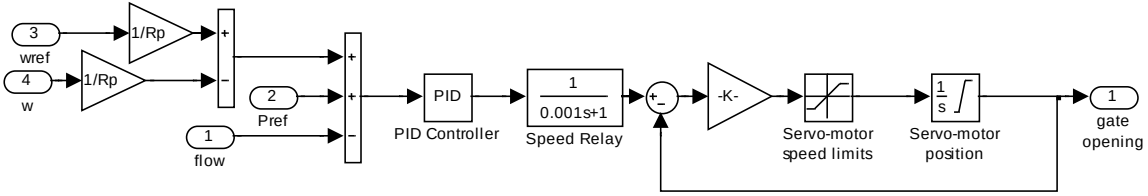
LAMPIRAN C

PEMODELAN *STEAM* TURBIN DAN *GOVERNING SYSTEM*



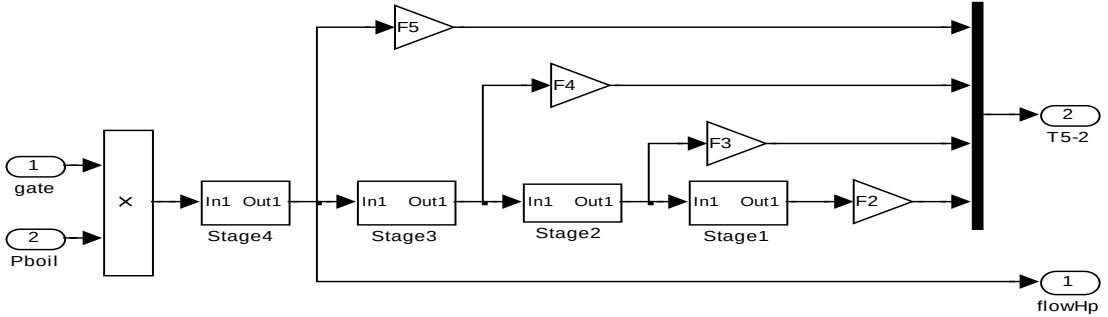
Pemodelan *steam* turbin & *governing system* secara keseluruhan

❖ ***Governing system* / *Speed Regulator***



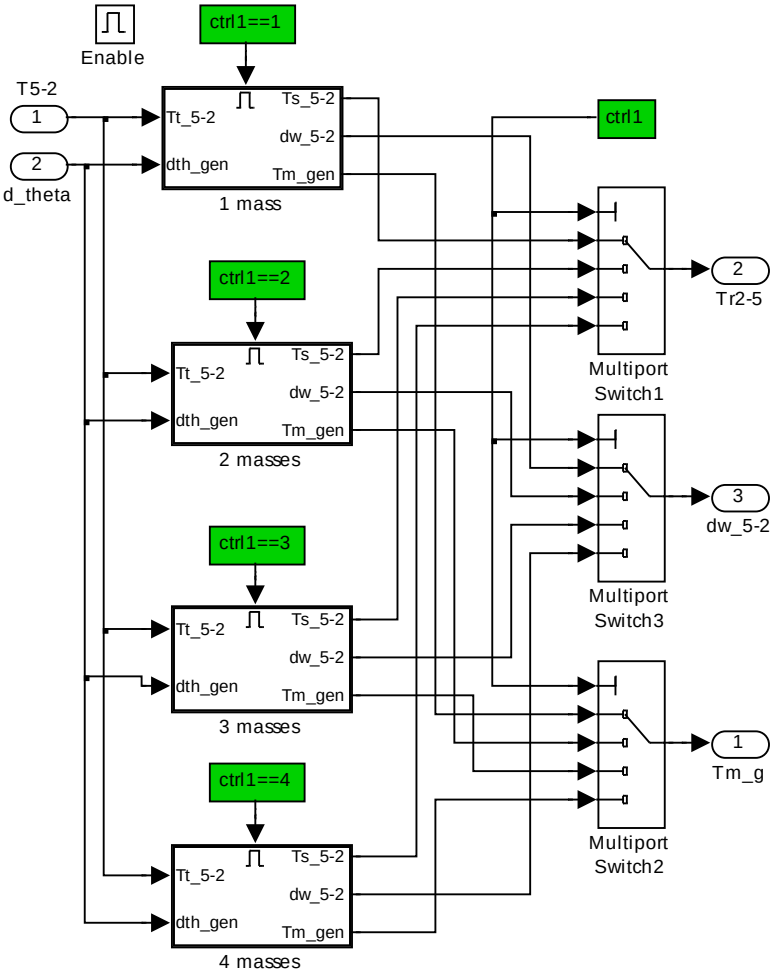
Pemodelan *governing system*

❖ ***Steam* Turbin**



Pemodelan *steam* turbin

❖ ***Shaft***



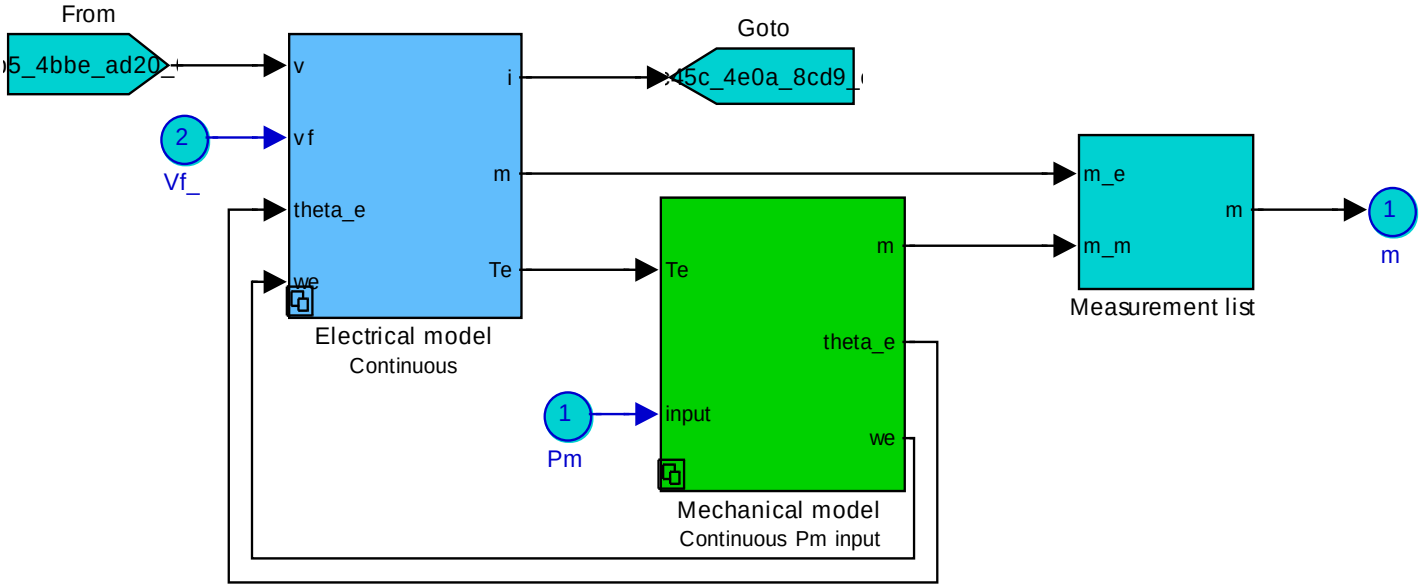
Pemodelan multi *shaft*

“Halaman ini sengaja dikosongkan.”

LAMPIRAN D

PEMODELAN GENERATOR & *EXCITTION SYSTEM*

A. PEMODELAN GENERATOR

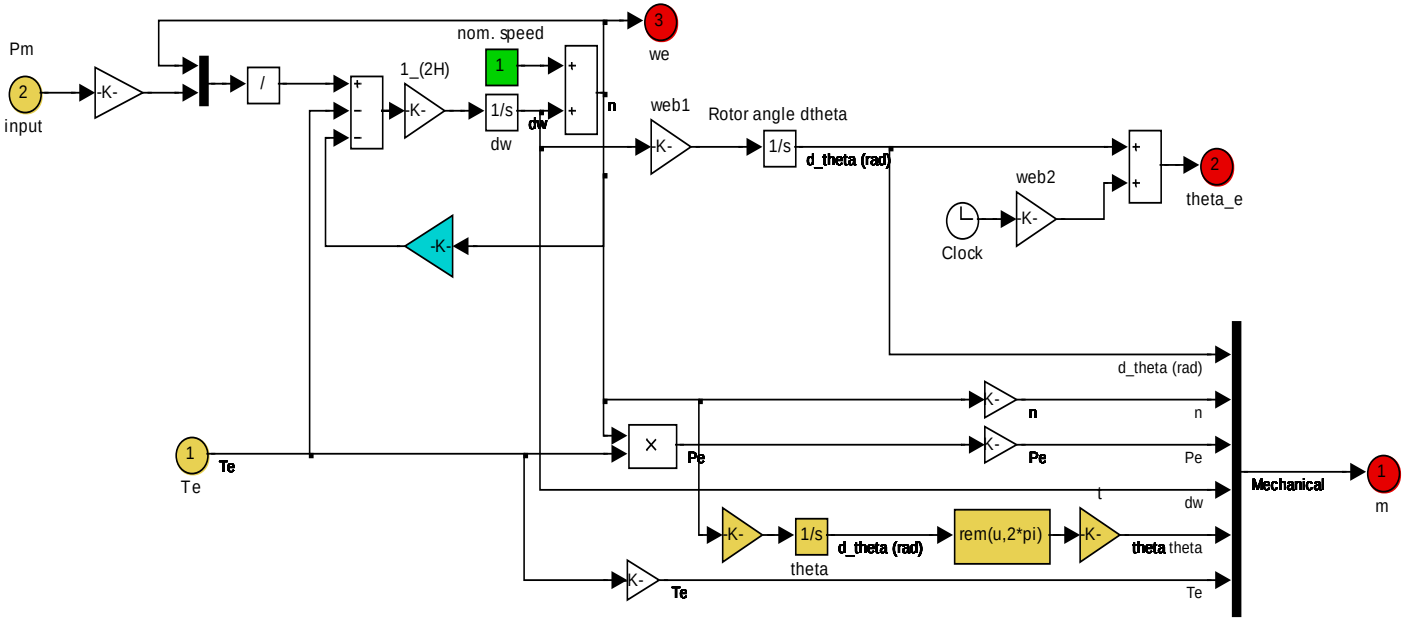


Generator *Subsystem*

D-2

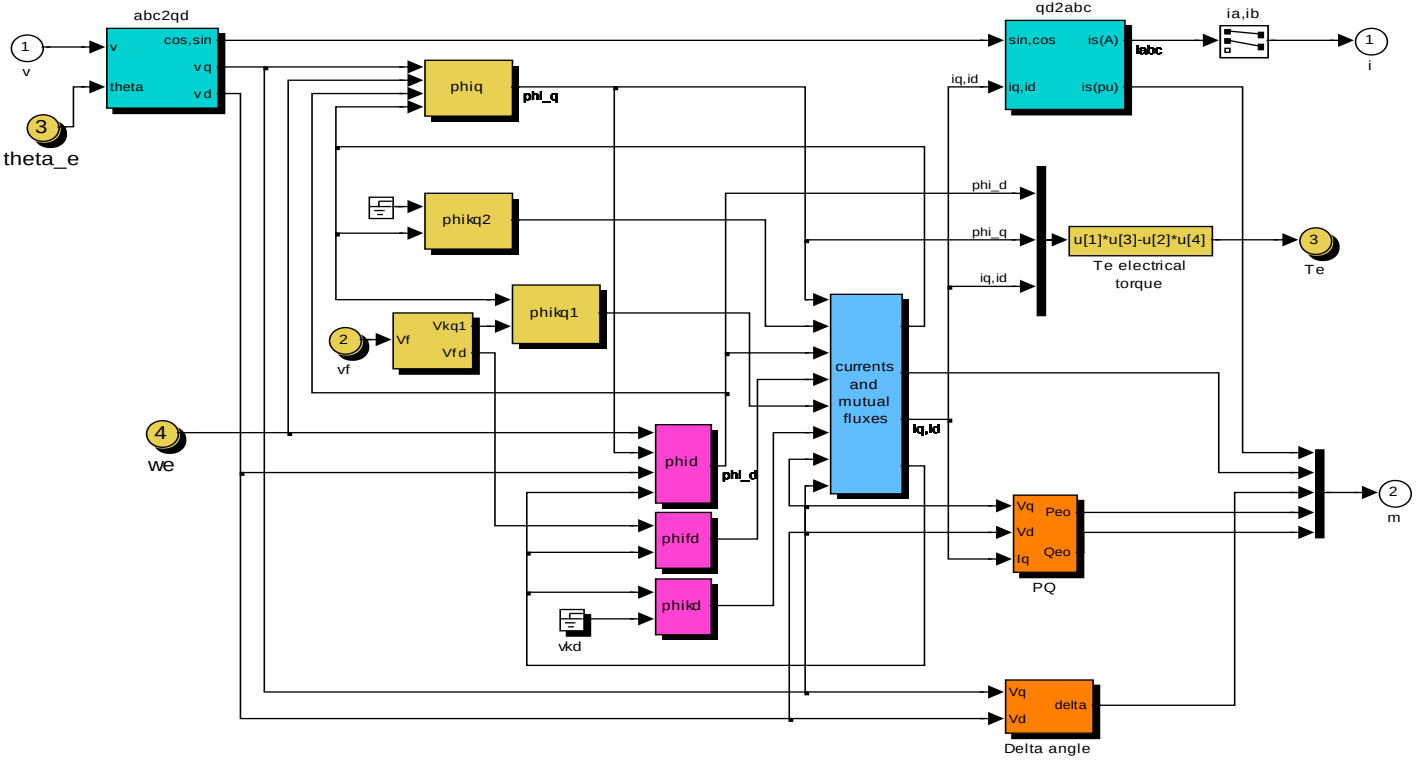
❖ *Mechanical Model of Synchronous Generator*

This subsystem models the mechanical part of the synchronous machine.



Mechanical model

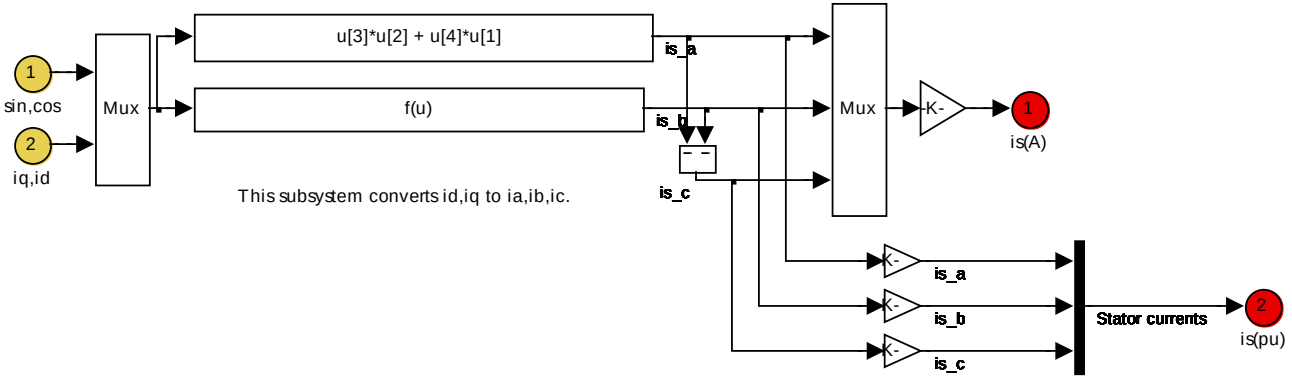
❖ *Electricall Model of Synchronous Generator*



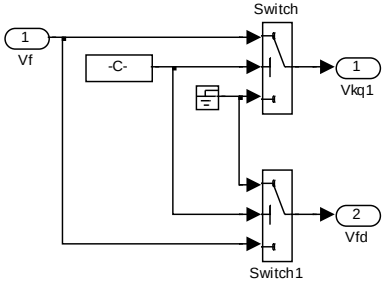
Electrical model

This subsystem converts Vab and Vbc to Vd,Vq. The sine and cosine computed here are sent to the inverse transformation subsystem qd/abc.

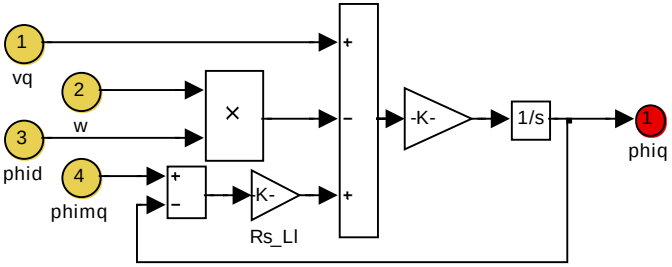
Rangkaian Vabc *to* Vdq *converter*



Rangkaian Idq *to* *Iabc* *converter*

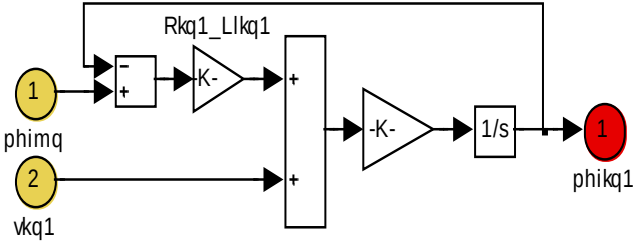


Rangkaian dari *Excitation system*



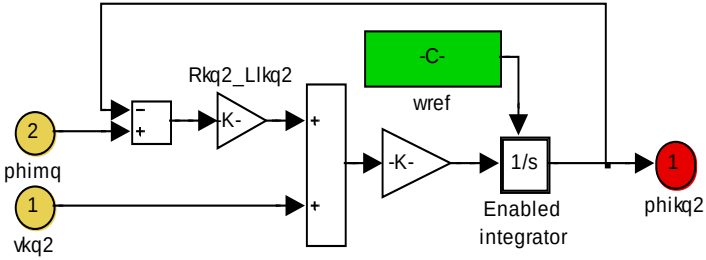
This subsystem computes the quadrature axis flux phiq.

Rangkaian φ_q



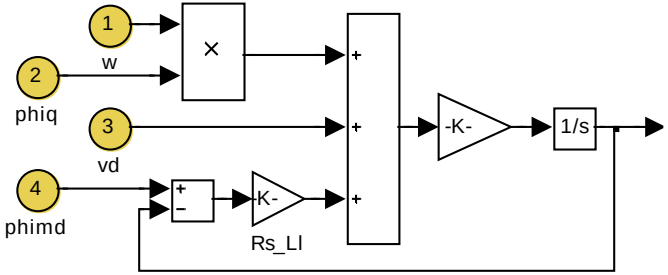
This subsystem computes the quadrature axis damper flux phikq.

Rangkaian φ_{kq1}



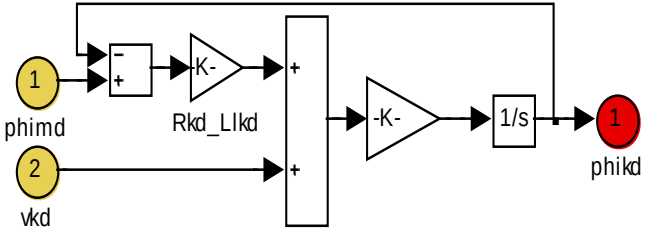
This subsystem computes the quadrature axis damper flux phikq2.

Rangkaian φ_{kq2}



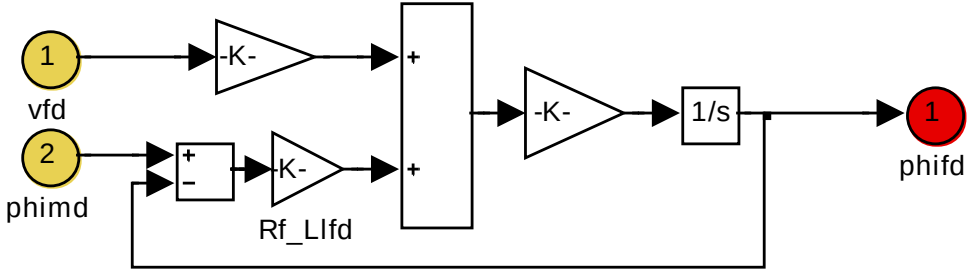
This subsystem computes the direct axis flux phid.

Rangkaian φ_d



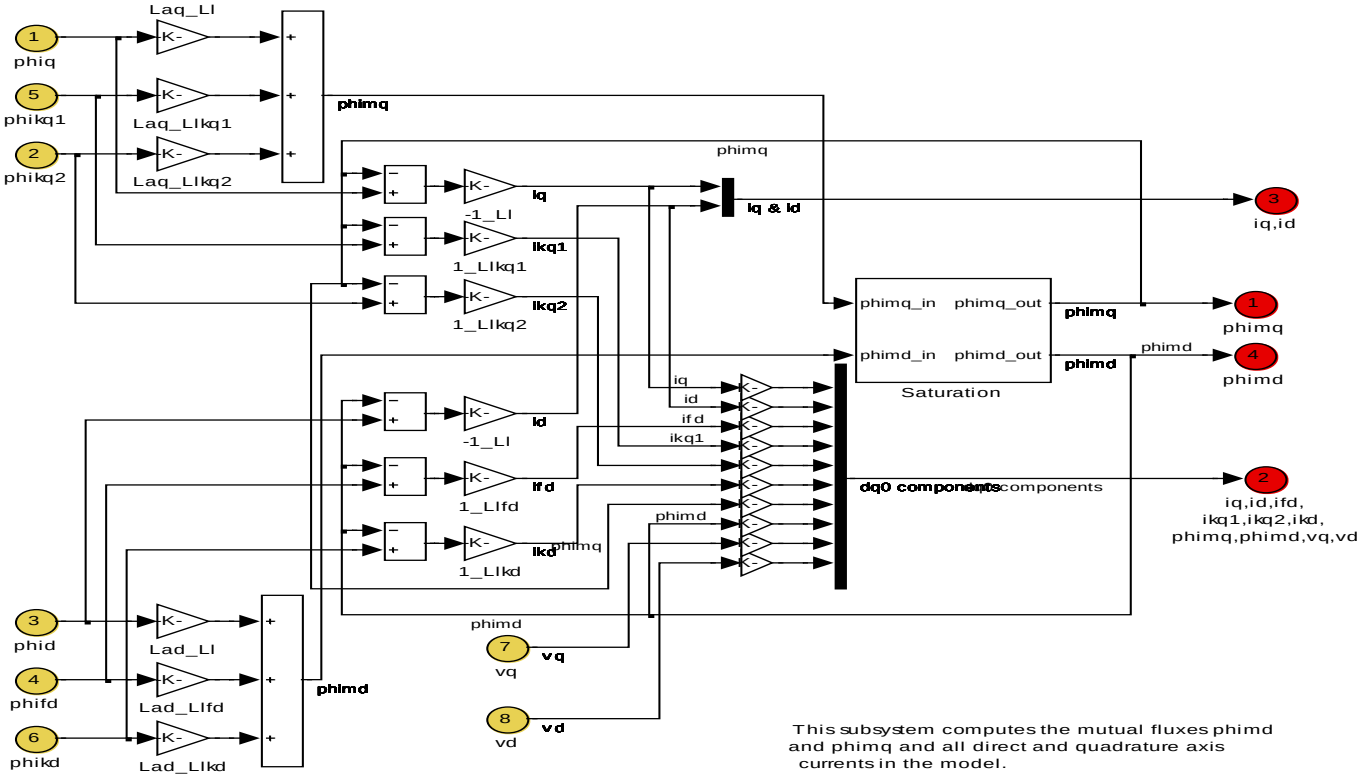
This subsystem computes the direct axis damper flux phikd.

Rangkaian φ_{kd}

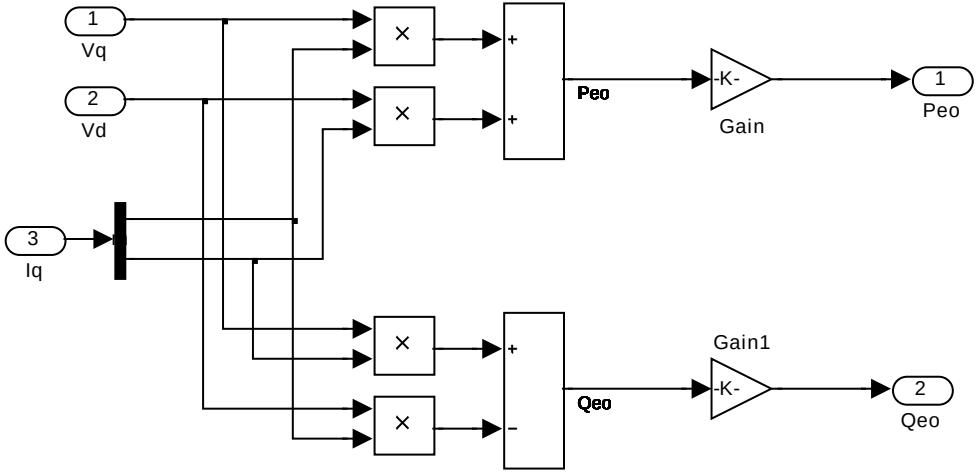


This subsystem computes the field flux phifd.

Rangkaian φ_{fd}

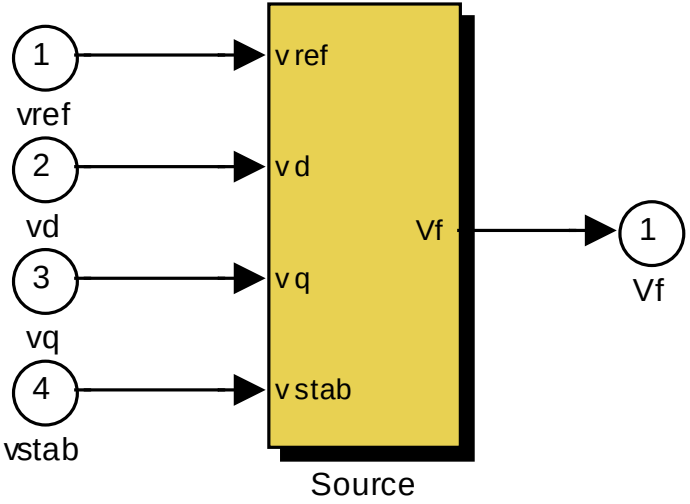


Rangkaian arus dan *mutual fluks*

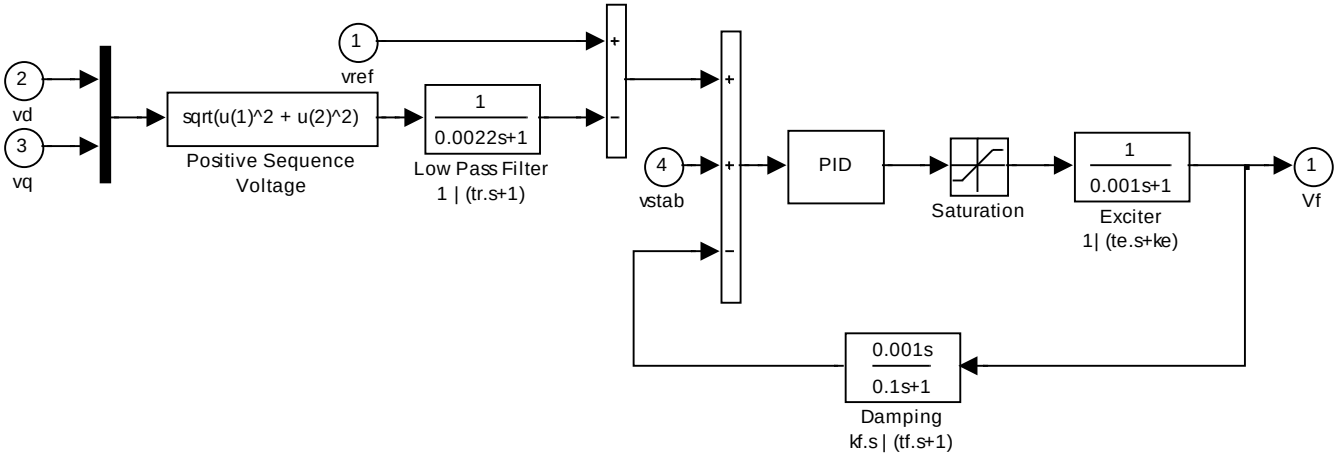


Rangkaian *power active* & *power reactive*

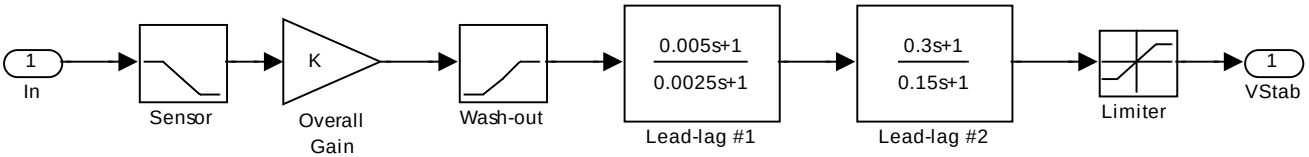
B. PEMODELAN *EXCITATION SYSTEM*



Rangkaian sistem eksitasi



Excitation Subsystem



Power System Stabilizer

LAMPIRAN E

DATA PROSES PT GEO DIPA ENERGI MARET 2012

DATA PROSES PT GEO DIPA ENERGI				
Date	*Time*	RPM	Volt	MW
1	8.00	2998	15163	17.4
	16.00	3000	14825	16
	0.00	3002	14836	17.5
2	8.00	2997	14937	15.53
	16.00	3000	14605	16.3
	0.00	3002	15044	16.7
3	8.00	3004	14909	16.4
	16.00	2990	14690	16.4
	0.00	3007	14861	16.1
4	8.00	2998	15197	16
	16.00	3000	15111	16.2
	0.00	3007	15091	16.5
5	8.00	2980	15030	15.9
	16.00	2990	14950	16.9
	0.00	3012	15190	16.7
6	8.00	2988	15128	15.4
	16.00	3007	14898	18.4
	0.00	2997	15103	15.9
7	8.00	2998	14910	15.7
	16.00	3008	14635	16.2
	0.00	2993	15080	15
8	8.00	3009	15149	14.7
	16.00	3002	14785	16.6
	0.00	3006	15125	15

9	8.00	2993	15168	14.8
	16.00	2997	14737	14.5
	0.00	3008	15200	14.7
10	8.00	3000	15222	15.6
	16.00	3008	14713	15
	0.00	2999	14845	15.4
11	8.00	2994	15012	15.7
	16.00	2984	15022	14.6
	0.00	3001	14922	15.5
12	8.00	2998	15012	15.1
	16.00	2992	15104	13.8
	0.00	3013	14940	15.3
13	8.00	3003	15052	16.2
	16.00	3001	14749	16.5
	0.00	3005	14908	16.1
14	8.00	3015	15159	14.4
	16.00	3012	14867	15.3
	0.00	2987	14987	15
15	8.00	2998	14959	14
	16.00	3012	14725	15.4
	0.00	3014	14660	14
16	8.00	3005	14905	14
	16.00	3004	14698	14
	0.00	2988	15000	14.8
17	8.00	3016	15050	14.5
	16.00	3000	14961	13.7
	0.00	2992	14902	12.3
18	8.00	3000	15061	14
	16.00	2989	15115	14.3
	0.00	3002	15090	14.2

19	8.00	3000	15146	14.1
	16.00	3004	14742	15.1
	0.00	3000	15102	15.1
20	8.00	3011	15050	15.2
	16.00	3002	14910	15.7
	0.00	2989	15050	16
21	8.00	3011	15225	16.8
	16.00	3007	14927	17.5
	0.00	3003	14988	16.8
22	8.00	2997	15165	17.6
	16.00	3000	14723	15.7
	0.00	3002	15055	15.9
23	8.00	3000	14817	15.1
	16.00	3002	14968	16
	0.00	3007	14708	16.1
24	8.00	3002	15192	15
	16.00	3004	14918	14.6
	0.00	3006	15185	15.4
25	8.00	2994	15008	15.8
	16.00	2999	14887	14.6
	0.00	3013	14952	14.8
26	8.00	2993	14918	15.2
	16.00	3004	14752	16
	0.00	2987	14786	14.6
27	8.00	3001	14875	14.6
	16.00	2997	14675	13.8
	0.00	3011	15056	14.5
28	8.00	3007	15154	14.2
	16.00	2998	14797	13.8
	0.00	3001	15004	14.2

29	8.00	3008	15076	12.5
	16.00	2995	14613	11.6
	0.00	3012	15087	15.4
30	8.00	3009	15121	15.1
	16.00	3000	14857	14.5
	0.00	2983	14807	14.5
31	8.00	2984	15110	15.4
	16.00	2998	14999	13.3
	0.00	2998	15135	14.4
Mean		3000.688	14962.76	15.05409

LAMPIRAN F

P & ID *PLANT*