



**UNIVERSITAS INDONESIA**

**PERANCANGAN PENGENDALI *MODEL PREDICTIVE CONTROL* DENGAN *CONSTRAINT* PADA SISTEM TATA UDARA PRESISI**

**SKRIPSI**

**RISE HAPSHARY SURAYUDA  
06 06 07 4312**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
JULI 2010**



**UNIVERSITAS INDONESIA**

**PERANCANGAN PENGENDALI *MODEL PREDICTIVE CONTROL* DENGAN *CONSTRAINT* PADA SISTEM TATA UDARA PRESISI**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana**

**RISE HAPSHARY SURAYUDA  
06 06 07 4312**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
JULI 2010**

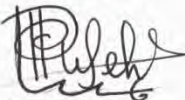
**HALAMAN PERNYATAAN ORISINALITAS**

**Skripsi ini adalah hasil karya saya sendiri,  
dan semua sumber baik yang dikutip maupun dirujuk  
telah saya nyatakan dengan benar.**

**Nama : Rise Hapshary Surayuda**

**NPM : 0606074312**

**Tanda Tangan :**



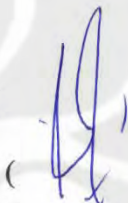


**Tanggal : 7 Juli 2010**

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :  
Nama : Rise Hapshary Surayuda  
NPM : 0606074312  
Program Studi : Teknik Elektro  
Judul Skripsi : Perancangan Pengendali *Model Predictive Control*  
dengan *Constraint* pada Sistem Tata Udara Presisi

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia.

### DEWAN PENGUJI

Pembimbing : Ir. Aries Subiantoro, M.Sc. (  )  
Penguji : Dr. Ir. Feri Yusivar, M.Eng. (  )  
Penguji : Prof. Dr.Eng. Drs. Benyamin Kusumoputro, M.Eng. (  )

Ditetapkan di : Depok

Tanggal : 7 Juli 2010

## KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Teknik Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

1. Ir. Aries Subiantoro, M.Sc., selaku dosen pembimbing yang telah menyediakan waktu, tenaga dan pikiran untuk mengarahkan dan membantu saya dalam penyusunan skripsi ini;
2. Dr.-Ing. Ir. Nasruddin, M.Eng. yang telah banyak membantu saya dalam memahami seluk-beluk mesin tata udara presisi;
3. Suria yang telah membantu saya dalam memahami algoritma MPC tanpa *constraint* yang saya gunakan pada skripsi ini;
4. orang tua dan keluarga saya yang telah memberikan bantuan dan dukungan berupa material, doa dan moral; dan
5. teman-teman Teknik Elektro tercinta dan juga sahabat-sahabat saya yang telah banyak memberikan semangat dalam menyelesaikan skripsi ini.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu di Indonesia, khususnya di Universitas Indonesia.

Depok, 7 Juli 2010

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS  
AKHIR UNTUK KEPENTINGAN AKADEMIS**

---

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Rise Hapshary Surayuda  
NPM : 0606074312  
Program Studi : Teknik Elektro  
Departemen : Teknik Elektro  
Fakultas : Teknik  
Jenis Karya : Skripsi

demikian pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul:

**PERANCANGAN PENGENDALI *MODEL PREDICTIVE CONTROL*  
DENGAN *CONSTRAINT* PADA SISTEM TATA UDARA PRESISI**

berserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di: Depok

Pada tanggal: 7 Juli 2010

Yang menyatakan



(Rise Hapshary Surayuda)



## ABSTRAK

Nama : Rise Hapshary Surayuda  
Program Studi : Teknik Elektro  
Judul : Perancangan Pengendali *Model Predictive Control* dengan *Constraint* pada Sistem Tata Udara Presisi

*Model Predictive Control* (MPC) merupakan salah satu metode pengendali prediktif yang populer digunakan pada dunia industri. Beberapa keuntungan yang ditawarkan oleh pengendali ini di antaranya adalah kemampuannya dalam menangani sistem multivariabel dengan cukup mudah dan juga kemampuannya untuk memberikan *constraint* atau batasan tertentu baik pada sinyal pengendali maupun pada keluaran sistem. Sistem tata udara presisi yang akan dikendalikan pada skripsi ini juga merupakan sistem multivariabel berorde tinggi yang mempunyai dua masukan dan dua keluaran. Model sistem yang dipakai berupa model linier diskrit yang didapat dari hasil identifikasi menggunakan metode N4SID. Hasil pengendalian menggunakan MPC akan dibandingkan dengan pengendalian sistem menggunakan metode pengendali penempatan kutub dari umpan balik keadaan.

Kata kunci : kendali prediktif, MPC, *constraint*, C-MEX, tata udara, multivariabel, linier, orde tinggi

## ABSTRACT

Name : Rise Hapshary Surayuda  
Study Program : Electrical Engineering  
Title : Design of Model Predictive Control with Constraints for Precision Air Conditioning System

Model Predictive Control (MPC) is one of predictive controller methods and popular for being used in industry. Some advantages offered by this controller are its ability to easily handle multivariable systems and its feature to provide constraints both for control signals and output signals. Precision air conditioning (PAC) system which will be controlled here is also a high-order multivariable system with two inputs and two outputs. Its model that is used in the controller is a discrete linear model from identification using N4SID method. The results of using MPC for this PAC system will be compared with the ones using another method of controller which is state-feedback pole placement.

Keywords : predictive control, MPC, constraints, C-MEX, air conditioning, multivariable, linear, high order

## DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN ORISINALITAS.....	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR .....	iv
HALAMAN PERSETUJUAN PUBLIKASI TUGAS AKHIR.....	v
ABSTRAK .....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	ix
DAFTAR GAMBAR .....	x
<b>1. PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Tujuan Penelitian.....	2
1.3 Pembatasan Masalah .....	2
1.4 Metodologi Penelitian .....	2
1.5 Sistematika Penulisan.....	3
<b>2. DASAR TEORI .....</b>	<b>4</b>
2.1 Konsep Dasar <i>Model Predictive Control</i> .....	4
2.1.1 Model Proses .....	6
2.1.2 Perhitungan Prediksi.....	7
2.1.3 Fungsi Kriteria.....	11
2.1.4 Minimasi Fungsi Kriteria MPC tanpa <i>Constraint</i> .....	12
2.1.5 Pembentukan <i>Constraint</i> pada MPC .....	13
2.2 <i>Relative Gain Array</i> .....	15
2.3 Sistem Tata Udara Presisi.....	18
2.3.1 Prinsip Kerja Sistem Tata Udara Presisi .....	19
2.3.2 Persamaan Matematis Sistem Tata Udara Presisi .....	22
2.3.2.1 Model Kompresor .....	23
2.3.2.2 Model Evaporator.....	24
2.3.2.3 Model Kondenser Kedua.....	26
2.3.2.4 Model Udara Masuk ( <i>Supply Air</i> ).....	27
2.3.2.5 Model Kabinet.....	28
<b>3. PERANCANGAN PENGENDALI MPC UNTUK SISTEM TATA UDARA PRESISI.....</b>	<b>29</b>
3.1 Model Linier PAC .....	29
3.2 Uji Interaksi Antarvariabel Masukan-Keluaran .....	30
3.3 Perancangan MPC tanpa <i>Constraint</i> Menggunakan C-MEX <i>S-function</i> 32	
3.3.1 Blok <i>S-function</i> MPC tanpa <i>Constraint</i> .....	33
3.3.1.1 Fungsi <i>MdlInitializeSizes</i> .....	34
3.3.1.2 Fungsi <i>MdlInitializeSampleTimes</i> .....	34
3.3.1.3 Fungsi <i>MdlInitializeConditions</i> .....	35
3.3.1.4 Fungsi <i>MdlUpdate</i> .....	36
3.3.1.5 Fungsi <i>MdlOutputs</i> .....	38
3.3.1.6 Fungsi <i>MdlTerminate</i> .....	39



3.3.2	Blok <i>S-function</i> Model PAC.....	39
3.3.2.1	Fungsi <i>MdlInitializeSizes</i> .....	39
3.3.2.2	Fungsi <i>MdlInitializeSampleTimes</i> .....	40
3.3.2.3	Fungsi <i>MdlInitializeConditions</i> .....	40
3.3.2.4	Fungsi <i>MdlUpdate</i> .....	40
3.3.2.5	Fungsi <i>MdlOutputs</i> .....	41
3.3.2.6	Fungsi <i>MdlTerminate</i> .....	41
3.4	Perancangan MPC dengan <i>Constraint</i> Menggunakan <i>M-file</i> .....	42
3.4.1	Algoritma MPC dengan <i>Constraint</i> pada <i>M-file</i> 'mpc_cons_pac.m' .....	42
3.4.2	Fungsi Kriteria pada <i>M-file</i> 'lossfct_pac.m' .....	45
<b>4.</b>	<b>HASIL SIMULASI DAN ANALISIS .....</b>	<b>46</b>
4.1	Pengendali MPC tanpa <i>Constraint</i> .....	46
4.1.1	Hasil Pengendali tanpa Blok Saturasi .....	46
4.1.1.1	Hasil Simulasi dan Analisis Variasi Nilai <i>Horizon</i> .....	46
4.1.1.2	Hasil Simulasi dan Analisis Variasi Bobot Matriks R.....	50
4.1.1.3	Hasil Simulasi dan Analisis Variasi Bobot Matriks Q.....	53
4.1.2	Hasil Pengendali dengan Blok Saturasi.....	56
4.2	Pengendali MPC dengan <i>Constraint</i> .....	58
4.2.1	Perbandingan Hasil Pengendali MPC tanpa dan dengan <i>Constraint</i> .....	58
4.2.2	Hasil Simulasi dan Analisis Variasi Bobot Matriks Q .....	59
4.2.3	Hasil Simulasi dan Analisis Variasi Bobot Matriks R .....	64
4.3	Pengendali Penempatan Kutub dengan Umpan Balik Keadaan.....	69
4.3.1	Penurunan Persamaan Pengendali.....	69
4.3.2	Hasil Simulasi dan Analisis Pengendali Penempatan Kutub .....	71
4.3.2.1	Hasil Pengendali tanpa Blok Saturasi .....	71
4.3.2.2	Hasil Pengendali dengan Blok Saturasi .....	73
<b>5.</b>	<b>KESIMPULAN DAN SARAN .....</b>	<b>75</b>
5.1	Kesimpulan.....	75
5.2	Saran.....	76
	DAFTAR REFERENSI .....	77

## DAFTAR TABEL

Tabel.	Hasil Eksperimen Sistem PAC tanpa Beban dengan Kecepatan Kompresor 3.100 rpm dan Katup Kondenser Kedua Tertutup Penuh .....	30
--------	---	----



## DAFTAR GAMBAR

Gambar 2.1.	Struktur Dasar MPC .....	4
Gambar 2.2.	Konsep <i>Receding Horizon</i> .....	6
Gambar 2.3.	Struktur Masukan-Keluaran Bentuk <i>P-Canonical</i> .....	15
Gambar 2.4.	Bagan Sistem Tata Udara Presisi .....	19
Gambar 2.5.	Diagram Pipa Sistem Tata Udara Presisi.....	20
Gambar 2.6.	Diagram P-h Siklus Refrigerasi.....	21
Gambar 2.7.	Skema Aliran Udara di Evaporator dan Kondenser Kedua.....	24
Gambar 2.8.	Skema Aliran Udara dan Refrigeran di Evaporator .....	25
Gambar 2.9.	Skema Aliran Udara di Kondenser Kedua .....	26
Gambar 3.1.	Diagram Blok <i>Simulink</i> MPC tanpa <i>Constraint</i> tanpa Blok Saturasi .....	32
Gambar 3.2.	Diagram Blok <i>Simulink</i> MPC tanpa <i>Constraint</i> dengan Blok Saturasi .....	33
Gambar 4.1.	Keluaran Sistem Hasil Pengendali MPC tanpa <i>Constraint</i> tanpa Blok Saturasi dengan Variasi Nilai <i>Horizon</i> .....	47
Gambar 4.2.	Masukan Sistem Hasil Pengendali MPC tanpa <i>Constraint</i> tanpa Blok Saturasi dengan Variasi Nilai <i>Horizon</i> .....	49
Gambar 4.3.	Keluaran Sistem Hasil Pengendali MPC tanpa <i>Constraint</i> tanpa Blok Saturasi dengan Variasi Nilai Bobot R.....	51
Gambar 4.4.	Masukan Sistem Hasil Pengendali MPC tanpa <i>Constraint</i> tanpa Blok Saturasi dengan Variasi Nilai Bobot R.....	52
Gambar 4.5.	Keluaran Sistem Hasil Pengendali MPC tanpa <i>Constraint</i> tanpa Blok Saturasi dengan Variasi Nilai Bobot Q.....	54
Gambar 4.6.	Masukan Sistem Hasil Pengendali MPC tanpa <i>Constraint</i> tanpa Blok Saturasi dengan Variasi Nilai Bobot Q.....	55
Gambar 4.7.	Keluaran dan Masukan Sistem Hasil Pengendali MPC tanpa <i>Constraint</i> dengan Blok Saturasi pada $H_p = H_u = 2$ , $Q = 0,5$ dan $R = 2$ .....	57

Gambar 4.8. Keluaran dan Masukan Sistem Hasil Pengendali MPC dengan <i>Constraint</i> pada $H_p = H_u = 2$ dan $Q = R = 1$ .....	58
Gambar 4.9. Keluaran Sistem Hasil Pengendali MPC dengan <i>Constraint</i> dengan Variasi Nilai Bobot $Q$ .....	61
Gambar 4.10. Masukan Sistem Hasil Pengendali MPC dengan <i>Constraint</i> dengan Variasi Nilai Bobot $Q$ .....	62
Gambar 4.11. Keluaran dan Masukan Sistem Hasil Pengendali MPC dengan <i>Constraint</i> dengan Nilai Awal Referensi $r_1 = 23,778$ .....	63
Gambar 4.12. Keluaran Sistem Hasil Pengendali MPC dengan <i>Constraint</i> dengan Variasi Nilai Bobot $R$ .....	65
Gambar 4.13. Masukan Sistem Hasil Pengendali MPC dengan <i>Constraint</i> dengan Variasi Nilai Bobot $R$ .....	66
Gambar 4.14. Keluaran dan Masukan Sistem Hasil Pengendali MPC dengan <i>Constraint</i> dengan Nilai Referensi Konstan $r_1 = 24$ dan $r_2 = 0,6$ .....	67
Gambar 4.15. Keluaran dan Masukan Sistem Hasil Pengendali MPC dengan <i>Constraint</i> dengan Nilai Referensi Konstan $r_1 = 23$ dan $r_2 = 0,6$ .....	68
Gambar 4.16. Diagram Blok Pengendali Penempatan Kutub dengan Umpan Balik Keadaan .....	69
Gambar 4.17. Keluaran dan Masukan Sistem Hasil Pengendali Penempatan Kutub dengan Umpan Balik Keadaan tanpa Blok Saturasi .....	71
Gambar 4.18. Keluaran dan Masukan Sistem Hasil Pengendali Penempatan Kutub dengan Umpan Balik Keadaan dengan Blok Saturasi .....	73

# BAB 1 PENDAHULUAN

## 1.1 Latar Belakang

Ruang pusat data (*data center*) menyimpan peralatan IT yang digunakan sebagai peralatan telekomunikasi. Peralatan IT yang digunakan biasa disimpan dalam kabinet di ruang pusat data tersebut. Untuk itulah dibutuhkan suatu sistem tata udara yang bisa mengatur temperatur dan kelembaban di dalam kabinet tersebut supaya dapat memperpanjang masa hidup (*lifetime*) peralatan IT.

Di Indonesia sendiri, kebanyakan pengaturan temperatur dan kelembaban tersebut masih dilakukan secara manual atau menggunakan tata udara ruangan biasa. Hal ini disebabkan tidak adanya produk lokal dari sistem tata udara untuk peralatan IT dan harga produk importnya yang relatif mahal. Padahal tata udara ruangan biasa berbeda dengan tata udara untuk kabinet jika dilihat dari beban dan kapasitas pendinginan yang dibutuhkan.

Oleh karena itu, dirancang suatu alat refrigerasi yang dapat mengatur temperatur dan kelembaban kabinet dan dikenal dengan nama tata udara presisi atau *precision air conditioning* (PAC). Alat ini dimaksudkan untuk menjaga temperatur kabinet di kisaran 20-22°C dan kelembaban relatif kabinet sebesar 45-55%. Agar alat ini dapat bekerja seperti yang diharapkan, dibutuhkan suatu metode pengendali yang dapat bekerja secara optimal untuk mendapatkan temperatur dan kelembaban yang selalu berada pada spesifikasi yang diinginkan. Metode pengendali yang dipilih adalah *model predictive control* (MPC).

MPC digunakan karena kemudahannya untuk diterapkan pada sistem multivariabel dan diharapkan mampu mengendalikan sistem tata udara presisi yang mempunyai dua masukan dan dua keluaran. Algoritma yang digunakan lebih sederhana dan bisa diterapkan pada sistem *single input single output* (SISO) dan sistem *multi input multi output* (MIMO) secara sama. Di samping itu, MPC dapat menyuguhkan nilai lebih berupa kemampuannya dalam memberikan batasan-batasan (*constraints*) baik untuk sinyal kendali maupun untuk keluaran sistem. Hal ini tentunya akan sangat berguna pada proses pengendalian sistem jika

dibandingkan dengan metode pengendali konvensional seperti pengendali dengan dua derajat kebebasan.

## 1.2 Tujuan Penelitian

Tujuan penelitian pada skripsi ini adalah untuk merancang suatu pengendali prediktif, yaitu MPC dengan *constraint*, yang diterapkan pada sistem tata udara presisi dan diharapkan dapat mengatur temperatur dan kelembaban agar berada pada nilai yang diinginkan.

## 1.3 Pembatasan Masalah

Masalah pada skripsi ini dibatasi pada perancangan MPC dengan *constraint* untuk sistem tata udara presisi dengan menggunakan perangkat lunak Matlab 7.6.0.324 (R2008a) untuk mengendalikan temperatur dan kelembaban sistem. Batasan atau *constraint* diberikan untuk perubahan sinyal kendali (*slew rate*) dan besar nilai sinyal kendali yang berupa kecepatan kompresor dan bukaan katup kondenser kedua.

Model yang dipakai adalah model linier diskrit hasil identifikasi menggunakan metode N4SID pada penelitian seminar. Hasil pengendalian menggunakan MPC akan dibandingkan dengan hasil pengendalian menggunakan pengendali penempatan kutub dari umpan balik keadaan (*state-feedback pole placement*).

## 1.4 Metodologi Penelitian

Metodologi penelitian yang dipakai pada skripsi ini di antaranya:

- Studi literatur, yaitu dengan membaca buku dan tesis mengenai MPC.
- Konsultasi dengan dosen pembimbing dan berdiskusi dengan teman yang melakukan penelitian serupa.
- Merancang pengendali MPC menggunakan perangkat lunak Matlab 7.6.0.324 (R2008a), yaitu *simulink C-MEX s-function* untuk MPC tanpa *constraint* dan pengendali penempatan kutub, serta *m-file* untuk MPC dengan *constraint*.

## 1.5 Sistematika Penulisan

Laporan skripsi ini akan dibagi menjadi lima bab, di mana masing-masing bab memuat hal berikut:

- Bab 1: Pendahuluan

Bab ini menjelaskan latar belakang, tujuan penelitian, pembatasan masalah, metodologi penelitian dan sistematika penulisan.

- Bab 2: Dasar Teori

Bab ini menjelaskan dasar teori mengenai konsep dasar MPC dan metode *Relative Gain Array* (RGA), serta membahas sistem tata udara presisi beserta persamaan matematisnya.

- Bab 3: Perancangan Pengendali MPC untuk Sistem Tata Udara Presisi

Bab ini menjelaskan model sistem tata udara presisi yang dipakai, uji interaksi antarvariabel masukan dan keluaran menggunakan metode RGA serta langkah-langkah dalam merancang pengendali MPC untuk sistem tata udara presisi, baik dengan maupun tanpa *constraint*, seperti algoritma MPC tanpa *constraint* menggunakan C-MEX *s-function* dan algoritma MPC dengan *constraint* menggunakan *m-file* pada sistem tata udara presisi.

- Bab 4: Hasil Simulasi dan Analisis

Bab ini membahas hasil pengendalian sistem tata udara presisi menggunakan MPC tanpa dan dengan *constraint*, hasil pengendalian menggunakan *state-feedback pole placement* untuk perbandingan dan analisis hasil simulasi yang telah didapat dari perancangan pengendali untuk sistem tata udara presisi.

- Bab 5: Kesimpulan dan Saran

Bab ini menjelaskan kesimpulan yang dapat diambil dari penelitian skripsi ini dan saran yang dapat diberikan untuk memperbaiki hasil pengendalian sistem.



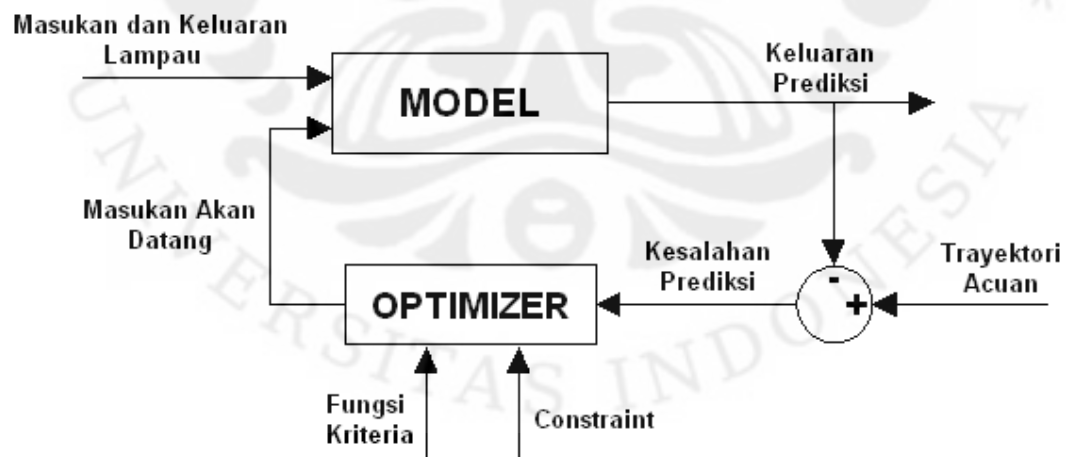
## BAB 2 DASAR TEORI

### 2.1 Konsep Dasar *Model Predictive Control*

*Model predictive control* (MPC) termasuk dalam metode pengendali prediktif yang berbasiskan kepada model proses. Setiap jenis pengendali prediktif mempunyai ide dasar sebagai berikut [1].

- Penggunaan model proses secara eksplisit untuk memprediksi keluaran proses yang akan datang di dalam rentang waktu tertentu (*horizon*).
- Perhitungan rangkaian sinyal kendali dengan meminimasi suatu fungsi kriteria.
- Strategi surut; pada setiap waktu pencuplikan (pada waktu  $k$ ), *horizon* dipindahkan menuju waktu pencuplikan berikutnya (pada waktu  $k + 1$ ) dengan melibatkan pemakaian sinyal kendali pertama, yaitu  $u(k)$ , untuk mengendalikan proses, dan prosedur di atas diulang dengan menggunakan informasi terakhir.

Struktur dasar MPC dapat dilihat pada gambar di bawah ini.

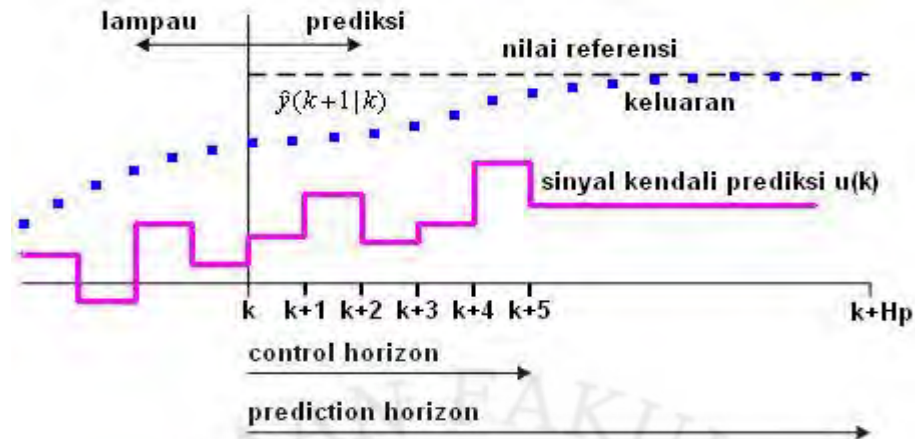


Gambar 2.1. Struktur Dasar MPC

Metodologi pengendali yang termasuk dalam kategori jenis MPC dideskripsikan secara umum oleh gambar 2.1 di atas dan dapat dijelaskan seperti berikut ini [1].

1. Keluaran proses yang akan datang untuk rentang *horizon*  $H_p$  yang ditentukan dinamakan *prediction horizon* dan diprediksi pada setiap waktu pencuplikan dengan menggunakan model proses. Keluaran proses prediksi ini,  $\hat{y}(k + i|k)$  untuk  $i = 1 \dots H_p$ , bergantung pada nilai masukan dan keluaran lampau, serta kepada sinyal kendali prediksi  $\hat{u}(k + i|k)$  untuk  $i = 0 \dots H_u - 1$  di mana  $H_u$  merupakan *control horizon* atau rentang yang ditentukan untuk perhitungan prediksi sinyal kendali yang akan digunakan pada sistem.
2. Serangkaian sinyal kendali dihitung dengan mengoptimasi suatu fungsi kriteria yang ditetapkan sebelumnya, dengan tujuan untuk menjaga proses sedekat mungkin terhadap trayektori acuan  $r(k + i)$ . Fungsi kriteria tersebut umumnya berupa fungsi kuadratik dari kesalahan antara sinyal keluaran prediksi dengan trayektori acuan. Dalam banyak kasus, tujuan pengendalian seperti pemakaian energi yang minimum disertakan di dalam fungsi kriteria. Solusi eksplisit dapat diperoleh jika fungsi kriteria adalah kuadratik, model linier dan tidak ada *constraint*. Jika tidak, optimasi iteratif harus digunakan untuk memecahkannya. Langkah pertama dan kedua ini dapat diilustrasikan pada gambar 2.2 di halaman berikutnya.
3. Sinyal kendali  $\hat{u}(k|k)$  dikirim ke proses, sedangkan sinyal kendali prediksi berikutnya dibuang karena keluaran pada pencuplikan berikutnya ( $y(k + 1)$ ) sudah diketahui nilainya. Oleh karena itu, langkah pertama diulang dengan nilai keluaran proses yang baru dan semua prosedur perhitungan yang diperlukan diperbaiki. Sinyal kendali baru  $\hat{u}(k + 1|k + 1)$  mempunyai nilai yang berbeda dari  $\hat{u}(k + 1|k)$  dan dihitung menggunakan konsep *receding horizon*.

Gambar 2.2 di halaman berikutnya mendeskripsikan konsep *receding horizon* yang merupakan ide dasar dari metode MPC ini.



Gambar 2.2. Konsep *Receding Horizon*

Metode MPC juga memiliki beberapa keuntungan yang merupakan nilai lebih dibandingkan metode pengendali lain, yaitu [1]:

- Konsepnya sangat intuitif serta penalaannya sangat mudah.
- Dapat digunakan untuk mengendalikan proses yang beragam, mulai dari proses yang sederhana sampai proses yang kompleks, mempunyai waktu tunda besar dan *non-minimum phase*.
- Dapat menangani sistem multivariabel.
- Mempunyai kompensasi terhadap waktu tunda.
- Mempunyai kemampuan dari pengendali *feed forward* untuk mengompensasi gangguan yang terukur.
- Mudah untuk diimplementasikan.
- *Constraint* dapat diperhitungkan dalam perancangan pengendali.
- Sangat berguna jika sinyal referensi untuk masa datang diketahui.

Selain keuntungan yang ditawarkan, metode MPC juga memiliki kerugian-kerugian [1], yaitu penurunan aturan sinyal kendali yang lebih rumit dibandingkan pengendali PID klasik dan kebutuhannya terhadap model proses yang baik sehingga jika model proses yang digunakan tidak terlalu bagus, hasil pengendalian juga tidak akan sesuai dengan yang diharapkan.

### 2.1.1 Model Proses

Seperti yang telah disebutkan sebelumnya, solusi eksplisit dari metode MPC dapat dicari dengan asumsi model proses bersifat linier sehingga persamaan

model proses yang dipakai pada metode ini merupakan persamaan model ruang keadaan linier seperti di bawah ini:

$$\underline{x}(k+1) = \underline{A} \underline{x}(k) + \underline{B} \underline{u}(k) \quad (2.1a)$$

$$\underline{y}(k) = \underline{C} \underline{x}(k) + \underline{D} \underline{u}(k) \quad (2.1b)$$

di mana  $\underline{x} \in \mathbb{R}^n$  merupakan vektor keadaan berdimensi orde ( $n$ ),  $\underline{u} \in \mathbb{R}^m$  merupakan vektor masukan sistem berupa sinyal kendali berdimensi jumlah masukan ( $m$ ),  $\underline{y} \in \mathbb{R}^p$  vektor keluaran sistem berdimensi jumlah keluaran ( $p$ ) dan  $\underline{A} \in \mathbb{R}^{n \times n}$ ,  $\underline{B} \in \mathbb{R}^{n \times m}$ ,  $\underline{C} \in \mathbb{R}^{p \times n}$ ,  $\underline{D} \in \mathbb{R}^{p \times m}$  merupakan matriks parameter sistem.

Adanya matriks  $\underline{D}$  menyatakan sistem memiliki *direct feedthrough*, yaitu masukan sistem ( $\underline{u}$ ) berpengaruh langsung terhadap keluaran sistem ( $\underline{y}$ ). Dalam hal ini, nilai *horizon* yang ditentukan harus memenuhi  $Hp = Hu$  jika  $\underline{D} \neq 0$ . Sebaliknya, jika  $\underline{D} = 0$ , maka nilai  $Hp$  tidak harus sama dengan  $Hu$ . Pada persamaan model proses di atas, diasumsikan tidak ada gangguan atau *disturbance* pada sistem.

### 2.1.2 Perhitungan Prediksi

Persamaan model proses (2.1) menyatakan nilai keluaran sistem dapat dicari dengan mendapatkan nilai keadaan ( $\underline{x}$ ) dan nilai masukan sistem ( $\underline{u}$ ) terlebih dahulu. Hal ini juga berlaku untuk mencari nilai keluaran prediksi sistem di mana estimasi keluaran prediksi bergantung pada nilai keadaan prediksi dan nilai masukan prediksi. Untuk itu, nilai estimasi keadaan prediksi  $\hat{\underline{x}}(k+i|k)$  selama rentang *prediction horizon* ( $Hp$ ) perlu dihitung terlebih dahulu sebagai berikut [2].

$$\begin{aligned} \hat{\underline{x}}(k+1|k) &= \underline{A} \underline{x}(k) + \underline{B} \hat{\underline{u}}(k|k) \\ \hat{\underline{x}}(k+2|k) &= \underline{A} \hat{\underline{x}}(k+1|k) + \underline{B} \hat{\underline{u}}(k+1|k) \\ &= \underline{A}^2 \underline{x}(k) + \underline{A} \underline{B} \hat{\underline{u}}(k|k) + \underline{B} \hat{\underline{u}}(k+1|k) \\ &\quad \vdots \\ \hat{\underline{x}}(k+Hp|k) &= \underline{A} \hat{\underline{x}}(k+Hp-1|k) + \underline{B} \hat{\underline{u}}(k+Hp-1|k) \\ &= \underline{A}^{Hp} \underline{x}(k) + \underline{A}^{Hp-1} \underline{B} \hat{\underline{u}}(k|k) + \dots + \underline{B} \hat{\underline{u}}(k+Hp-1|k) \end{aligned} \quad (2.2)$$

Besaran  $\hat{u}(k|k)$  menyatakan nilai estimasi sinyal kendali prediksi saat  $k$  untuk pencuplikan  $k$ . Notasi ini dipakai pada persamaan di atas karena saat estimasi keadaan prediksi, nilai  $\underline{u}(k)$  yang merupakan sinyal kendali saat  $k$  belum diketahui.

Selain nilai keadaan prediksi, nilai masukan prediksi  $\hat{u}(k+i|k)$  juga dibutuhkan untuk mencari nilai keluaran prediksi  $\hat{y}(k+i|k)$ . Hanya saja pada perhitungan dengan metode MPC ini, nilai sinyal kendali yang dihitung selama rentang *control horizon* adalah perubahan sinyal kendalinya ( $\Delta\hat{u}(k+i|k)$ ). Akan tetapi, nilai sinyal kendali prediksi bisa didapatkan dengan menggunakan persamaan:

$$\underline{u}(k) = \Delta\underline{u}(k) + \underline{u}(k-1) \quad (2.3)$$

sehingga estimasi sinyal kendali prediksi  $\hat{u}(k+i|k)$  selama rentang *control horizon* ( $Hu$ ) dengan  $i = 0, \dots, Hu - 1$  adalah [2]:

$$\begin{aligned} \hat{u}(k|k) &= \Delta\hat{u}(k|k) + \underline{u}(k-1) \\ \hat{u}(k+1|k) &= \Delta\hat{u}(k+1|k) + \Delta\hat{u}(k|k) + \underline{u}(k-1) \\ &\vdots \\ \hat{u}(k+Hu-1|k) &= \Delta\hat{u}(k+Hu-1|k) + \dots + \Delta\hat{u}(k|k) + \underline{u}(k-1) \end{aligned} \quad (2.4)$$

Nilai sinyal kendali yang telah didapat pada persamaan (2.4) dimasukkan ke persamaan (2.2) sehingga estimasi keadaan prediksi menjadi [2]:

$$\begin{aligned} \hat{x}(k+1|k) &= \underline{A} \underline{x}(k) + \underline{B} \underbrace{(\Delta\hat{u}(k|k) + \underline{u}(k-1))}_{\hat{u}(k|k)} \\ \hat{x}(k+2|k) &= \underline{A}^2 \underline{x}(k) + \underline{A} \underline{B} \underbrace{(\Delta\hat{u}(k|k) + \underline{u}(k-1))}_{\hat{u}(k|k)} \\ &\quad + \underline{B} \underbrace{(\Delta\hat{u}(k+1|k) + \Delta\hat{u}(k|k) + \underline{u}(k-1))}_{\hat{u}(k+1|k)} \\ &= \underline{A}^2 \underline{x}(k) + (\underline{A} + \underline{I}) \underline{B} \Delta\hat{u}(k|k) + \underline{B} \Delta\hat{u}(k+1|k) \\ &\quad + (\underline{A} + \underline{I}) \underline{B} \underline{u}(k-1) \\ &\vdots \end{aligned}$$

$$\begin{aligned}\hat{\underline{x}}(k + Hu|k) &= \underline{A}^{Hu} \underline{x}(k) + (\underline{A}^{Hu-1} + \dots + \underline{A} + \underline{I}) \underline{B} \Delta \hat{\underline{u}}(k|k) + \dots \\ &\quad + \underline{B} \Delta \hat{\underline{u}}(k + Hu - 1|k) + (\underline{A}^{Hu-1} + \dots + \underline{A} + \underline{I}) \underline{B} \\ &\quad \underline{u}(k - 1)\end{aligned}\quad (2.5)$$

Jika  $\underline{D} = 0$  dan  $Hp \neq Hu$ , maka estimasi keadaan prediksi di persamaan (2.5) dilanjutkan dengan mengingat bahwa  $\Delta \hat{\underline{u}}(k + i|k) = 0$  untuk  $i \geq Hu$  sesuai dengan prinsip *receding horizon* pada gambar 2.2 sehingga  $\hat{\underline{u}}(k + i|k) = \hat{\underline{u}}(k + Hu - i|k)$ . Lanjutan estimasi keadaan prediksi dari persamaan (2.5) menjadi [2]:

$$\begin{aligned}\hat{\underline{x}}(k + Hu + 1|k) &= \underline{A}^{Hu+1} \underline{x}(k) + (\underline{A}^{Hu} + \dots + \underline{A} + \underline{I}) \underline{B} \Delta \hat{\underline{u}}(k|k) + \dots \\ &\quad + (\underline{A} + \underline{I}) \underline{B} \Delta \hat{\underline{u}}(k + Hu - 1|k) \\ &\quad + (\underline{A}^{Hu} + \dots + \underline{A} + \underline{I}) \underline{B} \underline{u}(k - 1) \\ &\quad \vdots \\ \hat{\underline{x}}(k + Hp|k) &= \underline{A}^{Hp} \underline{x}(k) + (\underline{A}^{Hp-1} + \dots + \underline{A} + \underline{I}) \underline{B} \Delta \hat{\underline{u}}(k|k) + \dots \\ &\quad + (\underline{A}^{Hp-Hu} + \dots + \underline{A} + \underline{I}) \underline{B} \Delta \hat{\underline{u}}(k + Hu - 1|k) \\ &\quad + (\underline{A}^{Hp-1} + \dots + \underline{A} + \underline{I}) \underline{B} \underline{u}(k - 1)\end{aligned}\quad (2.6)$$

Persamaan (2.5) dan (2.6) di atas dapat diubah ke dalam bentuk matriks menjadi seperti di bawah ini [2]:

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

lampau

$$\begin{aligned}
& + \underbrace{\begin{bmatrix} \underline{B} & \cdots & \underline{0}_{n \times m} \\ \underline{A}\underline{B} + \underline{B} & \cdots & \underline{0}_{n \times m} \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{Hu-1} \underline{A}^i \underline{B} & \cdots & \underline{B} \\ \sum_{i=0}^{Hu} \underline{A}^i \underline{B} & \cdots & \underline{A}\underline{B} + \underline{B} \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{Hp-1} \underline{A}^i \underline{B} & \cdots & \sum_{i=0}^{Hp-Hu} \underline{A}^i \underline{B} \end{bmatrix}}_{\underline{\theta} \in \mathbb{R}^{(n \times Hp) \times (m \times Hu)}} \begin{bmatrix} \Delta \hat{u}(k|k) \\ \vdots \\ \Delta \hat{u}(k + Hu - 1|k) \end{bmatrix} \\
& \hspace{15em} \text{prediksi} \\
& \hspace{20em} (2.7)
\end{aligned}$$

Setelah mendapatkan nilai keadaan prediksi dan masukan prediksi, nilai keluaran prediksi  $\hat{y}(k + i|k)$  dengan  $i = 1, \dots, Hp$  sepanjang *prediction horizon* dapat dicari dengan:

$$\begin{aligned}
\hat{y}(k + 1|k) &= \underline{C} \hat{x}(k + 1|k) + \underline{D} \hat{u}(k|k) \\
\hat{y}(k + 2|k) &= \underline{C} \hat{x}(k + 2|k) + \underline{D} \hat{u}(k + 1|k) \\
&\vdots \\
\hat{y}(k + Hp|k) &= \underline{C} \hat{x}(k + Hp|k) + \underline{D} \hat{u}(k + Hu - 1|k)
\end{aligned} \tag{2.8}$$

Persamaan (2.8) di atas berlaku untuk sistem dengan  $\underline{D} \neq 0$  dan  $Hp = Hu$ . Jika  $\underline{D} = 0$ , bagian yang ada  $\underline{D}$ -nya dihilangkan. Persamaan tersebut jika diubah dalam bentuk matriks menjadi:

$$\begin{aligned}
\begin{bmatrix} \hat{y}(k + 1|k) \\ \vdots \\ \hat{y}(k + Hp|k) \end{bmatrix} &= \underbrace{\begin{bmatrix} \underline{C} & \underline{0}_{p \times n} & \cdots & \underline{0}_{p \times n} \\ \underline{0}_{p \times n} & \underline{C} & \cdots & \underline{0}_{p \times n} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{0}_{p \times n} & \underline{0}_{p \times n} & \cdots & \underline{C} \end{bmatrix}}_{\underline{C}_y \in \mathbb{R}^{(p \times Hp) \times (n \times Hp)}} \begin{bmatrix} \hat{x}(k + 1|k) \\ \vdots \\ \hat{x}(k + Hp|k) \end{bmatrix} \\
&+ \underbrace{\begin{bmatrix} \underline{D} & \underline{0}_{p \times m} & \cdots & \underline{0}_{p \times m} \\ \underline{0}_{p \times m} & \underline{D} & \cdots & \underline{0}_{p \times m} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{0}_{p \times m} & \underline{0}_{p \times m} & \cdots & \underline{D} \end{bmatrix}}_{\underline{D}_y \in \mathbb{R}^{(p \times Hp) \times (m \times Hu)}} \begin{bmatrix} \hat{u}(k|k) \\ \vdots \\ \hat{u}(k + Hu - 1|k) \end{bmatrix} \tag{2.9}
\end{aligned}$$

atau jika persamaan (2.7) dimasukkan ke persamaan (2.9) akan menghasilkan:



$$\begin{aligned}
\begin{bmatrix} \underline{\hat{y}}(k+1|k) \\ \underline{\hat{y}}(k+2|k) \\ \vdots \\ \underline{\hat{y}}(k+Hp|k) \end{bmatrix} &= \begin{bmatrix} \underline{C} \underline{A} \\ \underline{C} \underline{A}^2 \\ \vdots \\ \underline{C} \underline{A}^{Hp} \end{bmatrix} \underline{x}(k) + \begin{bmatrix} \underline{C} \underline{B} + \underline{D} \\ \underline{C} \underline{A} \underline{B} + \underline{D} \\ \vdots \\ \underline{C} (\sum_{i=0}^{Hp-1} \underline{A}^i \underline{B}) + \underline{D} \end{bmatrix} \underline{u}(k-1) \\
&+ \begin{bmatrix} \underline{C} (\sum_{i=0}^0 \underline{A}^i \underline{B}) + \underline{D} & \underline{D} & \underline{0}_{p \times m} & \cdots & \underline{0}_{p \times m} \\ \underline{C} (\sum_{i=0}^1 \underline{A}^i \underline{B}) + \underline{D} & \underline{C} (\sum_{i=0}^0 \underline{A}^i \underline{B}) + \underline{D} & \underline{D} & \cdots & \underline{0}_{p \times m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \underline{C} (\sum_{i=0}^{Hp-2} \underline{A}^i \underline{B}) + \underline{D} & \underline{C} (\sum_{i=0}^{Hp-3} \underline{A}^i \underline{B}) + \underline{D} & \underline{C} (\sum_{i=0}^{Hp-4} \underline{A}^i \underline{B}) + \underline{D} & \cdots & \underline{D} \\ \underline{C} (\sum_{i=0}^{Hp-1} \underline{A}^i \underline{B}) + \underline{D} & \underline{C} (\sum_{i=0}^{Hp-2} \underline{A}^i \underline{B}) + \underline{D} & \underline{C} (\sum_{i=0}^{Hp-3} \underline{A}^i \underline{B}) + \underline{D} & \cdots & \underline{C} (\sum_{i=0}^0 \underline{A}^i \underline{B}) + \underline{D} \end{bmatrix} \\
&\quad \underline{C}_y \underline{\Theta} \in \mathbb{R}^{(p \times Hp) \times (m \times Hu)}
\end{aligned}
\tag{2.10}$$

### 2.1.3 Fungsi Kriteria

Hal paling penting pada metode MPC adalah mendapatkan nilai perubahan sinyal kendali prediksi dengan meminimasi fungsi kriteria yang ditentukan. Fungsi kriteria ini merupakan fungsi kuadrat yang umumnya merupakan selisih antara keluaran prediksi dengan trayektori acuan. Persamaan fungsi kriteria yang dipakai pada metode MPC adalah [2]:

$$V(k) = \sum_{i=1}^{Hp} \left\| \underline{\hat{y}}(k+i|k) - \underline{r}(k+i|k) \right\|_{\underline{Q}(i)}^2 + \sum_{i=0}^{Hu-1} \left\| \Delta \underline{\hat{u}}(k+i|k) \right\|_{\underline{R}(i)}^2
\tag{2.11}$$

di mana:

$\underline{\hat{y}}(k+i|k)$  : estimasi keluaran prediksi untuk  $i$  langkah ke depan saat  $k$

$\underline{r}(k+i|k)$  : trayektori acuan  $i$  langkah ke depan saat  $k$

$\Delta \underline{\hat{u}}(k+i|k)$  : estimasi perubahan sinyal kendali prediksi untuk  $i$  langkah ke depan saat  $k$

$\underline{Q}(i)$  dan  $\underline{R}(i)$  : matriks faktor bobot di mana  $\underline{Q} \in \mathbb{R}^{(p \times Hp) \times (p \times Hp)}$  dan

$$\underline{R} \in \mathbb{R}^{(m \times Hu) \times (m \times Hu)}$$

$Hp$  : prediction horizon

$Hu$  : control horizon

Matriks bobot  $\underline{Q}$  dan  $\underline{R}$  merupakan matriks diagonal dengan persamaan  $\underline{Q} = c_1 \underline{I}_{(p \times Hp)}$  dan  $\underline{R} = c_2 \underline{I}_{(m \times Hu)}$  di mana  $c_1$  dan  $c_2$  merupakan nilai skalar tertentu yang besarnya dipilih berdasarkan kecocokan dengan sistem. Nilai  $c_1$  dan  $c_2$  ditentukan dengan melihat nilai matriks bobot yang menghasilkan keluaran sistem terbaik mengikuti sinyal referensi.

#### 2.1.4 Minimasi Fungsi Kriteria MPC tanpa *Constraint*

Fungsi kriteria pada persamaan (2.11) dapat diubah ke dalam bentuk sederhana sebagai berikut [2]:

$$V(k) = \|\underline{Y}(k) - \underline{T}(k)\|_{\underline{Q}}^2 + \|\Delta\underline{U}(k)\|_{\underline{R}}^2 \quad (2.12)$$

di mana

$$\underline{Y}(k) = \begin{bmatrix} \hat{y}(k+1|k) \\ \vdots \\ \hat{y}(k+Hp|k) \end{bmatrix}, \underline{T}(k) = \begin{bmatrix} r(k+1|k) \\ \vdots \\ r(k+Hp|k) \end{bmatrix}, \Delta\underline{U}(k) = \begin{bmatrix} \Delta\hat{u}(k|k) \\ \vdots \\ \Delta\hat{u}(k+Hu-1|k) \end{bmatrix}$$

Berdasarkan persamaan (2.10), keluaran  $\underline{Y}(k)$  dapat diubah menjadi [2]:

$$\underline{Y}(k) = \underline{C}_y \underline{\Psi} \underline{x}(k) + \underline{C}_y \underline{\Gamma} \underline{u}(k-1) + \underline{C}_y \underline{\Theta} \Delta\underline{U}(k) \quad (2.13)$$

Untuk meminimasi fungsi kriteria di atas, persamaan (2.12) tersebut perlu diubah kembali ke dalam suatu bentuk sederhana yang mengandung  $\Delta\underline{U}(k)$  supaya didapatkan nilai perubahan sinyal kendali optimal setelah fungsi kriteria tersebut diminimasi. Untuk itu dibutuhkan matriks kesalahan penjejukan  $\underline{E}(k)$ , yaitu selisih antara nilai trayektori acuan yang akan datang dengan respon bebas sistem. Respon bebas adalah nilai keluaran di sepanjang *horizon* saat tidak ada perubahan sinyal kendali ( $\Delta\underline{U}(k) = 0$ ). Persamaan matematis dari  $\underline{E}(k) \in \mathbb{R}^{(p \times Hp)}$  [2]:

$$\underline{E}(k) = \underline{T}(k) - \underline{C}_y \underline{\Psi} \underline{x}(k) - \underline{C}_y \underline{\Gamma} \underline{u}(k-1) \quad (2.14)$$

Persamaan fungsi kriteria (2.12) kemudian dapat diubah menjadi suatu persamaan yang mengandung  $\underline{E}(k)$  dan  $\Delta\underline{U}(k)$  seperti berikut [2]:

$$\begin{aligned}
V(k) &= \|\underline{C}_y \underline{\theta} \Delta \underline{U}(k) - \underline{E}(k)\|_{\underline{Q}}^2 + \|\Delta \underline{U}(k)\|_{\underline{R}}^2 \\
&= (\Delta \underline{U}^T(k) \underline{\theta}^T \underline{C}_y^T - \underline{E}^T(k)) \underline{Q} (\underline{C}_y \underline{\theta} \Delta \underline{U}(k) - \underline{E}(k)) + \Delta \underline{U}^T(k) \underline{R} \Delta \underline{U}(k) \\
&= \underbrace{\underline{E}^T(k) \underline{Q} \underline{E}(k)}_c - \underbrace{\Delta \underline{U}^T(k) \underline{2 \theta^T \underline{C}_y^T \underline{Q} \underline{E}(k)}}_{\underline{G} \in \mathbb{R}^{(m \times Hu)}} \\
&\quad + \underbrace{\Delta \underline{U}^T(k) (\underline{\theta}^T \underline{C}_y^T \underline{Q} \underline{C}_y \underline{\theta} + \underline{R}) \Delta \underline{U}(k)}_{\underline{H} \in \mathbb{R}^{(m \times Hu) \times (m \times Hu)}} \\
V(k) &= c - \Delta \underline{U}^T(k) \underline{G} + \Delta \underline{U}^T(k) \underline{H} \Delta \underline{U}(k)
\end{aligned} \tag{2.15}$$

Nilai  $\underline{E}^T(k) \underline{Q} \underline{E}(k)$  merupakan nilai konstan karena tidak mengandung  $\Delta \underline{U}(k)$ . Dari persamaan (2.15), didapat vektor  $\underline{G}$  dan matriks  $\underline{H}$  seperti yang terlihat di bawah ini [2].

$$\underline{G} = 2 \underline{\theta}^T \underline{C}_y^T \underline{Q} \underline{E}(k) \tag{2.16}$$

$$\underline{H} = \underline{\theta}^T \underline{C}_y^T \underline{Q} \underline{C}_y \underline{\theta} + \underline{R} \tag{2.17}$$

Nilai minimum dari fungsi kriteria persamaan (2.15) bisa didapat dengan mencari turunan pertama atau gradien  $\nabla_{\Delta \underline{U}(k)} V(k) = 0$ . Persamaan gradien tersebut menjadi [2]:

$$\nabla_{\Delta \underline{U}(k)} V(k) = -\underline{G} + 2 \underline{H} \Delta \underline{U}(k) \tag{2.18}$$

sehingga dengan  $\nabla_{\Delta \underline{U}(k)} V(k) = 0$  didapat  $\Delta \underline{U}(k)$  optimal:

$$\Delta \underline{U}(k)_{opt} = \frac{1}{2} \underline{H}^{-1} \underline{G} \tag{2.19}$$

### 2.1.5 Pembentukan *Constraint* pada MPC

Pada dasarnya, fungsi kriteria yang dipakai pada MPC dengan *constraint* sama dengan MPC tanpa *constraint*. Yang membedakan adalah terdapat kriteria tambahan berupa batasan-batasan (*constraints*) yang diberikan dalam bentuk pertidaksamaan dan/atau persamaan. Batasan-batasan ini bisa ditetapkan untuk perubahan sinyal kendali (*slew rate*), nilai sinyal kendali dan keluaran sistem.

Sebelum meminimasi fungsi kriteria, perlu ditentukan batasan-batasan yang akan dipakai pada perancangan MPC nanti. Sebagai contoh adalah *constraint*

untuk perubahan sinyal kendali dan nilai sinyal kendali yang ditunjukkan pada pertidaksamaan berikut.

$$\underline{u}_{min} \leq \underline{u} \leq \underline{u}_{max} \quad (2.20a)$$

$$\Delta \underline{u}_{min} \leq \Delta \underline{u} \leq \Delta \underline{u}_{max} \quad (2.20b)$$

Pertidaksamaan di atas dapat diubah ke dalam bentuk:

$$-\underline{u} \leq -\underline{u}_{min} \quad \text{dan} \quad \underline{u} \leq \underline{u}_{max} \quad (2.21a)$$

$$-\Delta \underline{u} \leq -\Delta \underline{u}_{min} \quad \text{dan} \quad \Delta \underline{u} \leq \Delta \underline{u}_{max} \quad (2.21b)$$

Jika diubah ke dalam bentuk pertidaksamaan matriks:

$$\underbrace{\begin{bmatrix} -1 & 0 & 0 & & \\ 0 & \ddots & 0 & & \\ 0 & 0 & -1_{(2 \times m \times Hu)} & & \\ 1 & 0 & 0 & & \\ 0 & \ddots & 0 & & \\ 0 & 0 & 1_{(2 \times m \times Hu)} & & \end{bmatrix}}_{\underline{F} \in \mathbb{R}^{(4 \times m \times Hu) \times (2 \times m \times Hu + m)}} \underbrace{\begin{bmatrix} \Delta \hat{u}(k|k) \\ \vdots \\ \Delta \hat{u}(k + Hu - 1|k) \\ \hat{u}(k|k) \\ \vdots \\ \hat{u}(k + Hu - 1|k) \\ \underline{u}(k - 1) \end{bmatrix}}_{\underline{x}_{aug} \in \mathbb{R}^{(2 \times m \times Hu + m)}} \leq \underbrace{\begin{bmatrix} -\Delta \underline{u}_{min(m \times Hu)} \\ -\underline{u}_{min(m \times Hu)} \\ \Delta \underline{u}_{max(m \times Hu)} \\ \underline{u}_{max(m \times Hu)} \end{bmatrix}}_{\underline{F}_1 \in \mathbb{R}^{(4 \times m \times Hu)}} \quad (2.22)$$

atau  $\underline{F} \underline{x}_{aug} \leq \underline{F}_1$ .

Mengingat persamaan (2.3), bentuk pertidaksamaan matriks di atas dapat diubah ke dalam bentuk persamaan matriks berikut:

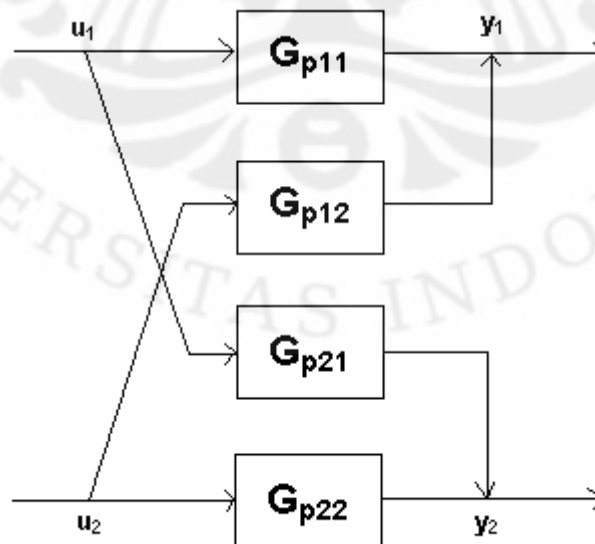
$$\underbrace{\begin{bmatrix} 1 & 0 & \dots & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & \dots & 0 & 0 & -1 & 0 & 0 & 0 & 1_{(m)} \\ \vdots & \vdots & \ddots & \vdots & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & \dots & 1_{(m \times Hu)} & 0 & 1_{(m \times Hu - m)} & 0 & -1_{(m \times Hu)} & 0 & 0 \end{bmatrix}}_{\underline{F}_{eq} \in \mathbb{R}^{(m \times Hu) \times (2 \times m \times Hu + m)}}$$

$$\underbrace{\begin{bmatrix} \Delta \hat{u}(k|k) \\ \vdots \\ \Delta \hat{u}(k + Hu - 1|k) \\ \hat{u}(k|k) \\ \vdots \\ \hat{u}(k + Hu - 1|k) \\ \underline{u}(k - 1) \end{bmatrix}}_{\underline{x}_{aug} \in \mathbb{R}^{(2 \times m \times Hu + m)}} = \underbrace{\mathbf{0}_{(m \times Hu)}}_{\underline{F}_{1eq}} \quad (2.23)$$

atau  $\underline{F}_{eq} \underline{x}_{aug} = \underline{F}_{1eq}$ . Setelah *constraint* didapat dalam bentuk matriks,  $\Delta \underline{u}$  optimal bisa didapatkan dengan meminimasi fungsi kriteria dengan beberapa algoritma optimasi. Salah satunya adalah *active-set* dan minimasi fungsi kriteria ini bisa dilakukan dengan memanggil fungsi *fmincon* ( ) pada Matlab.

## 2.2 Relative Gain Array

*Relative Gain Array* (RGA) merupakan metode yang digunakan untuk menguji interaksi antarvariabel masukan dan keluaran. Metode ini diperkenalkan oleh Bristol dan digunakan untuk mencari pasangan masukan dan keluaran yang terbaik untuk sistem multivariabel, yaitu masukan yang paling mempengaruhi keluaran pasangannya. Dengan memilih pasangan masukan dan keluaran terbaik, efek *coupling* antarvariabel bisa direduksi. Perhatikan gambar berikut.



Gambar 2.3. Struktur Masukan-Keluaran Bentuk *P-Canonical*

Gambar di atas memperlihatkan interaksi antarloop masukan-keluaran dalam bentuk *feedforward coupling*. Persamaan keluaran bentuk *P-canonical* tersebut adalah [3]:

$$y_1 = u_1 G_{11}^p + u_2 G_{12}^p \quad (2.24a)$$

$$y_2 = u_1 G_{21}^p + u_2 G_{22}^p \quad (2.24b)$$

dengan  $G^p$  adalah fungsi alih proses.

Fungsi alih proses  $G_{ij}^p$  memiliki *gain* tunak  $K_{ij}$ . Diasumsikan  $u_2$  konstan, maka perubahan  $u_1$  sebesar  $\Delta u_1$  akan menghasilkan  $\Delta y_1$  pada  $y_1$  sehingga *gain* antara  $u_1$  dan  $y_1$  saat  $u_2$  konstan diberikan oleh [3]:

$$K_{11}|_{u_2} = \left. \frac{\Delta y_1}{\Delta u_1} \right|_{u_2} \quad (2.25)$$

Sekarang  $y_2$  dibuat konstan menggantikan  $u_2$  dengan menutup loop antara  $u_2$  dan  $y_2$ . Perubahan  $u_1$  sebesar  $\Delta u_1$  akan menghasilkan perubahan  $y_1$  yang lain sehingga *gain* pada kondisi ini menjadi [3]:

$$K_{11}|_{y_2} = \left. \frac{\Delta y_1}{\Delta u_1} \right|_{y_2} \quad (2.26)$$

Walaupun kedua *gain* di atas didapat dengan variabel  $\Delta u_1$  dan  $\Delta y_1$  yang sama, nilai *gain* yang didapat bisa jadi berbeda karena kondisi sistem yang berbeda. Jika terjadi interaksi, maka kedua *gain* di atas akan mempunyai nilai yang berbeda dengan perbandingan [3]:

$$\lambda_{11} = \frac{K_{11}|_{u_2}}{K_{11}|_{y_2}} \quad (2.27)$$

di mana  $\lambda_{11}$  merupakan *gain* relatif antara  $y_1$  dan  $u_1$ .

Untuk sistem dengan jumlah masukan  $m$  dan jumlah keluaran  $p$  (dengan ketentuan  $m = p$ ), terdapat  $(m \times p)$  elemen *gain* relatif antara keluaran  $y_i$  dan masukan  $u_j$  yang diberikan oleh [3]:

$$\lambda_{ij} = \left. \frac{K_{ij}|_u}{K_{ij}|_y} \right|_y = \frac{\left( \frac{\Delta y_i}{\Delta u_j} \right) \Big|_u}{\left( \frac{\Delta y_i}{\Delta u_j} \right) \Big|_y} \quad (2.28)$$

Dengan demikian, *relative gain array* (RGA) menjadi [3]:

$$\Lambda = \begin{bmatrix} \lambda_{11} & \cdots & \lambda_{1m} \\ \vdots & \ddots & \vdots \\ \lambda_{p1} & \cdots & \lambda_{pm} \end{bmatrix} \quad (2.29)$$

Untuk sistem dengan dua masukan dan dua keluaran, RGA-nya adalah  $\Lambda = \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{bmatrix}$  dengan ketentuan  $\lambda_{12} = 1 - \lambda_{11}$ ,  $\lambda_{21} = \lambda_{12}$  dan  $\lambda_{22} = \lambda_{11}$ . Oleh karena itu, untuk sistem ini, *gain* relatif lainnya dapat diketahui dengan hanya menghitung satu elemen *gain* relatif.

Prosedur untuk mencari RGA di atas merupakan prosedur eksperimen. Prosedur analitis dapat dilakukan jika model memiliki nilai tunak sehingga [3]:

$$y_1 = K_{11}u_1 + K_{12}u_2 \quad (2.30a)$$

$$y_2 = K_{21}u_1 + K_{22}u_2 \quad (2.30b)$$

maka  $K_{11}|_{u_2} = \left. \frac{\partial y_1}{\partial u_1} \right|_{u_2} = K_{11}$  dan dengan mensubstitusi  $u_2$  dari persamaan (2.30b) ke (2.30a) menghasilkan:

$$y_1 = K_{11}u_1 + K_{12}(y_2 - K_{21}u_1)/K_{22} \quad (2.31)$$

Kemudian, dengan menurunkan persamaan (2.31) terhadap  $u_1$  dan menjaga  $y_2$  konstan didapat  $K_{11}|_{y_2} = \left. \frac{\partial y_1}{\partial u_1} \right|_{y_2} = K_{11} - K_{12}K_{21}/K_{22}$  sehingga *gain* relatif  $\lambda_{11}$  menjadi [3]:

$$\lambda_{11} = \left. \frac{K_{11}|_{u_2}}{K_{11}|_{y_2}} \right|_y = \frac{1}{1 - (K_{12}K_{21})/(K_{11}K_{22})} \quad (2.32)$$



Berdasarkan *gain* tunak yang telah dihitung seperti di atas, RGA bisa didapatkan dengan persamaan:

$$\Lambda = \mathbf{K} .* (\mathbf{K}^T)^{-1} \quad (2.33)$$

Notasi ‘.\*’ menunjukkan perkalian antareleman dan  $\mathbf{K} = \begin{bmatrix} K_{11} & \cdots & K_{1m} \\ \vdots & \ddots & \vdots \\ K_{p1} & \cdots & K_{pm} \end{bmatrix}$ .

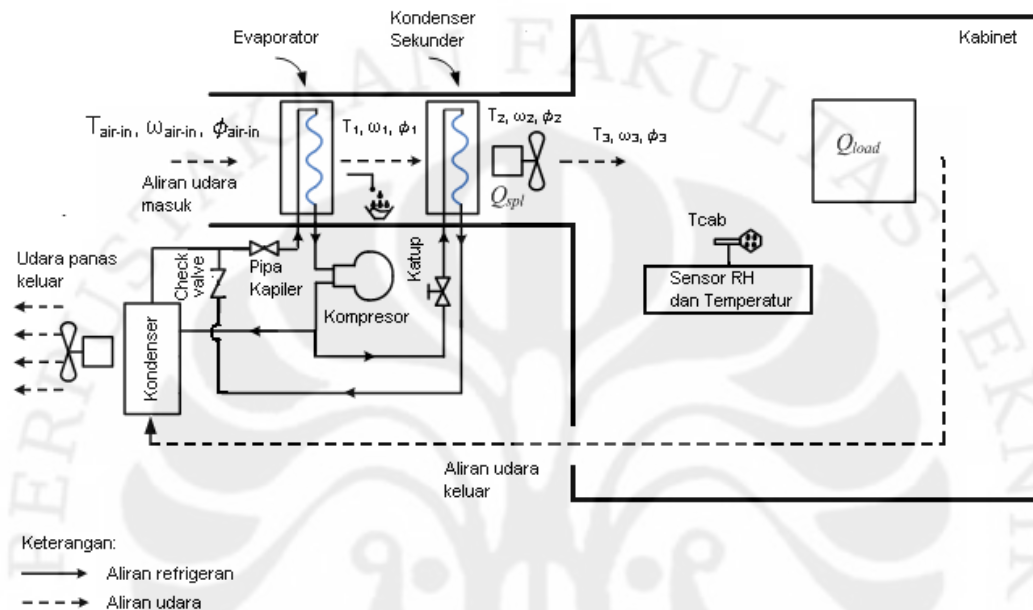
Hasil dari RGA yang didapat bisa dianalisis untuk mendapatkan pasangan masukan-keluaran terbaik dengan keterangan sebagai berikut [3]:

- $\lambda_{ij} = 1$ . Tidak ada interaksi antarlup masukan-keluaran.
- $\lambda_{ij} = 0$ . Masukan  $u_i$  tidak mempengaruhi keluaran  $y_j$ .
- $\lambda_{ij} = 0,5$ . Lup yang lain memiliki pengaruh yang sama kuat terhadap keluaran  $y_j$ . Ini merupakan kondisi terburuk karena walaupun pasangan masukan-keluarannya ditukar, tidak akan mengurangi efek *coupling*.
- $0,5 < \lambda_{ij} < 1$ . Terdapat interaksi antarlup tetapi ini merupakan pasangan terbaik untuk mengurangi efek *coupling*.
- $0 < \lambda_{ij} < 0,5$ . Terdapat interaksi antarlup dan  $u_i$  tidak terlalu mempengaruhi  $y_j$  sehingga pasangan masukan-keluaran harus ditukar.
- $\lambda_{ij} > 1$ . Interaksi mempengaruhi efek *gain* dari lup kendali sehingga dibutuhkan *gain* pengendali yang lebih besar.
- $\lambda_{ij} < 0$ . Nilai negatif mengindikasikan menutup lup kendali akan mengubah tanda dari *gain* sehingga terjadi ketidakstabilan untuk semua pengendali umpan balik.

### 2.3 Sistem Tata Udara Presisi

Sistem tata udara presisi atau yang lebih dikenal dengan *Precision Air Conditioning* (PAC) merupakan mesin refrigerasi yang bekerja berdasarkan konsep termodinamika. Mesin refrigerasi sendiri adalah alat yang melakukan proses pemindahan panas dari media temperatur rendah ke media temperatur tinggi dengan memanfaatkan siklus refrigrasi atau *vapor-compression cycle*.

PAC digunakan di ruang pusat data untuk menjaga temperatur dan kelembaban di dalam kabinet agar peralatan IT di dalam kabinet tidak cepat rusak. Temperatur ideal yang harus dicapai di dalam kabinet berkisar antara 20-22°C, sedangkan kelembaban relatif (RH) yang harus dijaga di dalam kabinet berkisar antara 45-55%. Berikut ini adalah gambar bagan sistem PAC.



Gambar 2.4. Bagan Sistem Tata Udara Presisi [4]

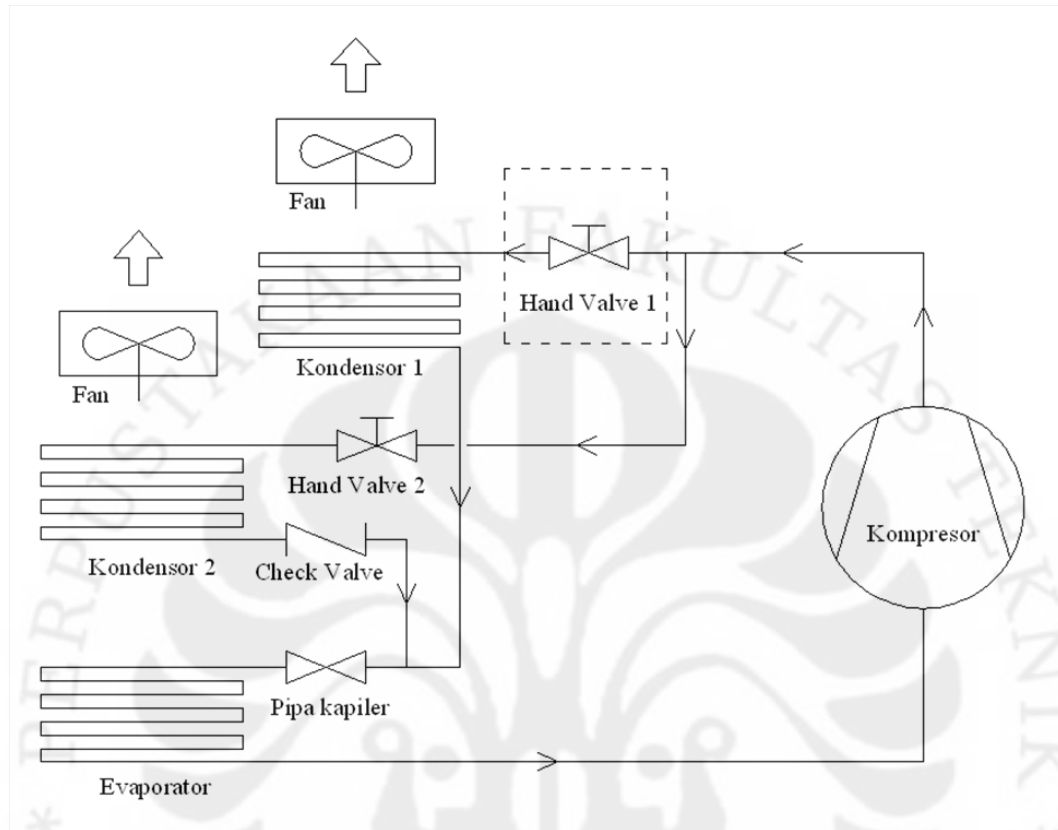
“telah diolah kembali”

Pada skripsi ini hanya akan dibahas seperlunya mengenai sistem tata udara presisi karena pemodelan PAC secara lebih lengkap, konsep dasar dari mesin refrigerasi, *vapor-compression cycle*, persamaan matematis yang didapat berdasarkan konsep termodinamika dan hal-hal mendetail dan mendasar lainnya sudah dibahas oleh Sutarna [4] berdasarkan jurnal oleh Deng dan Qi [5].

### 2.3.1 Prinsip Kerja Sistem Tata Udara Presisi

Sebelum membahas prinsip kerja dari sistem tata udara presisi, komponen-komponen yang terdapat pada mesin PAC ini perlu diketahui terlebih dahulu, antara lain kompresor, evaporator, dua kondenser, dua *fan*, pipa kapiler, katup (*electronic valve*, tapi pada saat proses identifikasi dilakukan, katup masih berupa

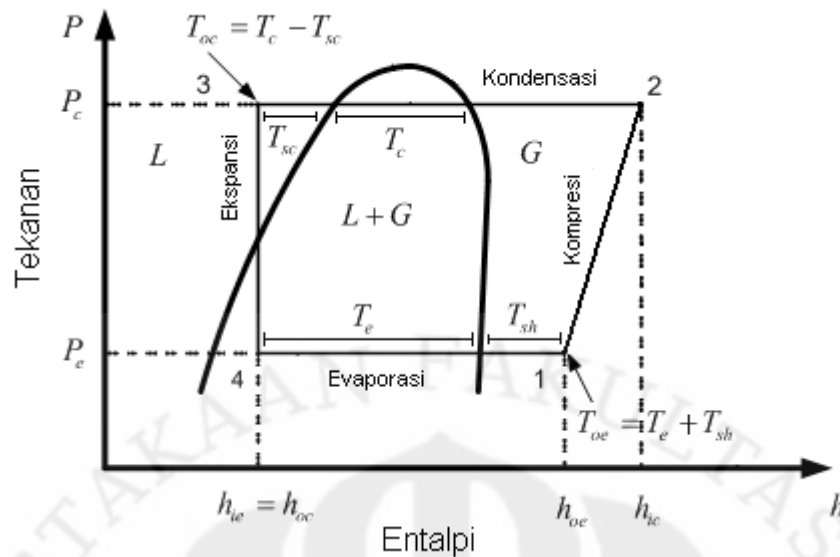
*hand valve*) dan *check valve*. Untuk memudahkan penjelasan, disediakan diagram PAC di bawah ini.



Gambar 2.5. Diagram Pipa Sistem Tata Udara Presisi [6]

Fluida yang bekerja sebagai refrigeran pada PAC ini adalah R134a, sedangkan prinsip kerja PAC sendiri adalah sebagai berikut. Pada sistem tata udara presisi ini terdapat dua aliran, yaitu aliran udara dan aliran refrigeran. Aliran udara ditarik masuk oleh *fan* dari ruang pusat data ke dalam PAC melalui evaporator kemudian ke kondenser kedua dan masuk ke dalam kabinet. Dari kabinet udara kembali masuk ke dalam PAC untuk kemudian dibuang keluar oleh kondenser pertama (lihat gambar 2.4).

Di sisi lain untuk melihat tahap aliran refrigeran pada PAC, akan lebih mudah jika melihat gambar berikut.



Gambar 2.6. Diagram P-h Siklus Refrigerasi [4]

“telah diolah kembali”

Tahap kompresi (1-2) dijelaskan sebagai berikut. Saat PAC dinyalakan kompresor mulai bekerja menaikkan tekanan refrigeran dan mengalirkannya ke kondenser pertama. Refrigeran yang keluar dari kompresor berupa gas dengan tekanan dan temperatur yang tinggi.

Tahap kondensasi (2-3) terjadi di kondenser pertama. Pada kondenser pertama temperatur refrigeran lebih tinggi dibandingkan temperatur udara di luar sehingga terjadi perpindahan panas dari refrigeran ke lingkungan luar yang menyebabkan udara yang dibuang oleh kondenser pertama menjadi lebih panas. Akibat pembuangan panas dari refrigeran ke lingkungan luar, terjadi proses kondensasi sehingga terjadi perubahan fase pada refrigeran dari gas ke cair.

Tahap ekspansi (3-4) terjadi di pipa kapiler. Refrigeran R134a cair yang keluar dari kondenser pertama kemudian mengalir ke dalam pipa kapiler. Di sini tekanan refrigeran menurun drastis karena ada efek penghambatan oleh alat ekspansi.

Tahap evaporasi (4-1) terjadi di evaporator. Refrigeran yang telah mempunyai tekanan dan temperatur rendah masuk ke dalam evaporator. Di evaporator temperatur refrigeran lebih rendah dibandingkan temperatur udara dalam PAC. Oleh karena itu, terjadi penyerapan panas oleh refrigeran yang mengakibatkan temperatur udara yang keluar dari evaporator ( $T_1$ ) lebih rendah

dibanding dengan temperatur udara ruang pusat data ( $T_{air-in}$ ) yang masuk ke dalam evaporator dengan kelembaban relatif (RH) yang tinggi ( $\phi_1 = 95\%$  s/d  $100\%$ ). Akibat penyerapan panas ini, terjadi proses evaporasi sehingga refrigeran berubah fase dari cair ke gas (lihat gambar 2.6).

Refrigeran dalam bentuk gas yang keluar dari evaporator kemudian masuk kembali ke kompresor untuk dikompresi, dan begitu seterusnya. Siklus ini dikenal dengan siklus refrigerasi atau *vapor-compression cycle* dan terjadi di setiap mesin refrigerasi pada umumnya. Namun, PAC yang digunakan di sini memiliki kondenser tambahan sehingga refrigeran yang keluar dari kompresor tidak hanya mengalir ke kondenser pertama, tapi juga ke kondenser kedua jika katup kondenser kedua dibuka (lihat gambar 2.5).

Sama halnya seperti pada tahap kondensasi (2-3) yang terjadi di kondenser pertama, di kondenser kedua temperatur refrigeran lebih tinggi dibandingkan temperatur udara di dalam PAC sehingga terjadi pembuangan panas dari refrigeran ke udara luar. Hal ini menyebabkan temperatur udara yang keluar dari kondenser kedua ( $T_2$ ) lebih tinggi dibanding temperatur udara yang masuk ke kondenser kedua ( $T_1$ ) dengan kelembaban relatif yang lebih rendah ( $\phi_2 < \phi_1$ ) (lihat gambar 2.4). Adanya kenaikan temperatur tersebut sesuai dengan fungsi kondenser kedua sebagai kondenser *reheat*, yaitu memanaskan kembali udara yang keluar dari evaporator karena umumnya udara yang keluar dari evaporator memiliki temperatur yang rendah dan kelembaban yang tinggi. Diharapkan dengan adanya kondenser *reheat* ini, temperatur dan kelembaban udara yang masuk ke kabinet dapat disesuaikan dengan mengatur bukaan katup kondenser kedua agar diperoleh nilai seperti spesifikasi yang diinginkan, yaitu temperatur  $20-22^\circ\text{C}$  dan RH 45-55%.

### 2.3.2 Persamaan Matematis Sistem Tata Udara Presisi

Untuk membahas persamaan matematis, sebaiknya asumsi-asumsi yang berlaku pada sistem tata udara presisi ini diperhatikan, yaitu:

- Campuran udara terjadi di dalam evaporator, kondenser dan lingkungan
- Temperatur evaporasi di evaporator dianggap konstan

- Sisi udara di evaporator meliputi daerah kering (*dry region*) dan daerah basah (*wet region*)
- Perbandingan volume sisi udara *dry region* terhadap volume sisi udara *wet region* adalah 1:4
- Temperatur kondensasi di kondenser dianggap konstan
- Sisi udara di kondenser hanya meliputi daerah kering (*dry region*) saja
- Aliran refrigeran yang mengalir ke kondenser sekunder diasumsi 10% dari *mass flow refrigerant* total, yaitu aliran refrigeran yang keluar dari kompresor atau aliran refrigeran yang masuk ke/keluar dari evaporator
- Tekanan di kompresor dianggap konstan
- Beban dianggap konstan
- Kecepatan aliran udara volumetris (*air volumetric flow*) dalam sistem dianggap konstan
- Rugi-rugi panas pada daerah aliran udara diabaikan.

Asumsi di atas juga digunakan dalam proses identifikasi pada penelitian seminar terdahulu.

Karena pada sistem ini keluaran yang akan dikendalikan nanti adalah temperatur kabinet ( $T_{cab}$ ) dan kelembaban relatif kabinet ( $\phi_{cab}$ ), maka persamaan matematis yang dipakai adalah persamaan-persamaan dari model kompresor, model evaporator, model kondenser kedua, model udara masuk atau *supply air* (udara yang masuk ke kabinet) dan model kabinet.

Persamaan matematis sistem yang tersedia adalah berupa persamaan aliran udara pada sistem PAC, dan pada gambar 2.4 terlihat bahwa aliran udara pada sistem tata udara presisi sudah mencapai kabinet sebelum melewati kondenser pertama. Oleh karena itu, persamaan matematis model kondenser pertama tidak dibutuhkan karena informasi keluaran sistem berupa temperatur dan RH udara dalam kabinet ( $T_{cab}$ ,  $\phi_{cab}$ ) sudah didapatkan tanpa memerlukan informasi mengenai udara yang dibuang keluar oleh kondenser pertama.

### 2.3.2.1 Model Kompresor

Persamaan matematisnya [5]:

$$M_{ref} = \frac{sV_{com}}{v_s} \left( 1 - 0,015 \left[ \left( \frac{P_c}{P_e} \right)^{\frac{1}{\beta}} - 1 \right] \right) \quad (2.34)$$

di mana

$M_{ref}$  : aliran massa refrigeran total keluaran kompresor (kg/s)

$s$  : kecepatan kompresor (rps)

$V_{com}$  : *swept volume* kompresor ( $m^3$ )

$v_s$  : volume spesifik dari *superheat refrigerant* ( $m^3/kg$ )

$P_c$  : tekanan kondensasi (kPa)

$P_e$  : tekanan evaporasi (kPa)

$\beta$  : indeks kompresi

*Swept volume* dapat dicari dengan [7]:

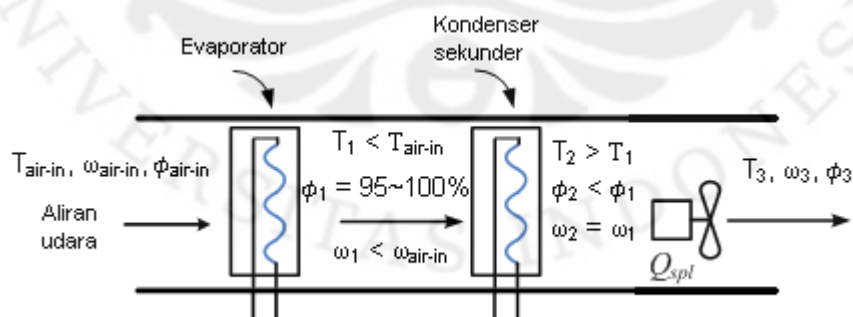
$$V_{com} = \frac{V_d}{n_c} \quad (2.35)$$

di mana

$V_d$  : *displacement volume* kompresor ( $m^3$ )

$n_c$  : jumlah silinder pada kompresor

### 2.3.2.2 Model Evaporator

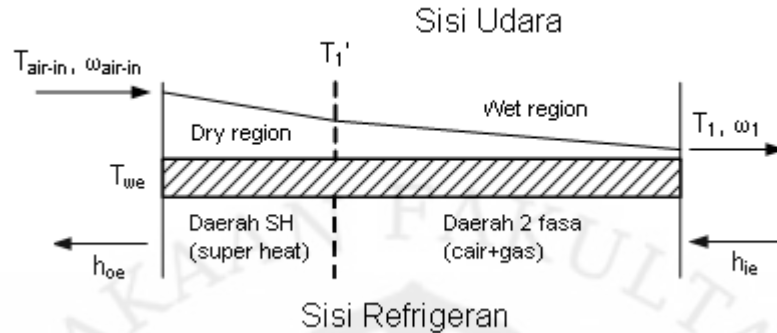


Gambar 2.7. Skema Aliran Udara di Evaporator dan Kondenser Kedua [4]

“telah diolah kembali”

Gambar di atas adalah skema aliran udara yang melewati evaporator dan kondenser kedua beserta keterangan perubahan temperatur dan kelembaban

selama melewati kedua komponen tersebut. Evaporator sendiri memiliki skema aliran udara yang terbagi menjadi *dry region* dan *wet region*.



Gambar 2.8. Skema Aliran Udara dan Refrigeran di Evaporator [4]

“telah diolah kembali”

Persamaan matematis di *dry region* [5]:

$$C_{pu}\rho_u V_1 \frac{dT_1'}{dt} = C_{pu}\rho_u f (T_{air-in} - T_1') + UA_1 \left( T_{we} - \frac{T_{air-in} + T_1'}{2} \right) \quad (2.36)$$

Persamaan matematis di *wet region* [5]:

$$C_{pu}\rho_u V_2 \frac{dT_1}{dt} + \rho_u V_2 h_{fg} \frac{d\omega_1}{dt} = C_{pu}\rho_u f (T_1' - T_1) + \rho_u f h_{fg} (\omega_{air-in} - \omega_1) + UA_2 \left( T_{we} - \frac{T_1' + T_1}{2} \right) \quad (2.37)$$

$$\omega_1 = (0,0198T_1^2 + 0,085T_1 + 4,4984)/1000 \quad (2.38a)$$

$$\frac{d\omega_1}{dt} = (2 \times 0,0198T_1 + 0,085) \frac{dT_1}{dt} / 1000 \quad (2.38b)$$

Persamaan matematis dinding evaporator [5]:

$$C_{pw}\rho_w V_{we} \frac{dT_{we}}{dt} = UA_1 \left( \frac{T_{air-in} + T_1'}{2} - T_{we} \right) + UA_2 \left( \frac{T_1' + T_1}{2} - T_{we} \right) - M_{ref1} (h_{oe} - h_{ie}) \quad (2.39)$$

di mana

$C_{pu}$  : kalor spesifik udara (kJ/kg°C)

$C_{pw}$  : kalor spesifik dinding evaporator/kondenser (kJ/kg°C)

$\rho_u$  : kerapatan udara (kg/m<sup>3</sup>)

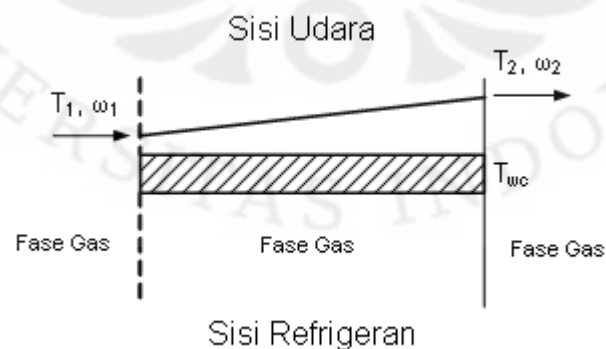
$\rho_w$  : kerapatan dinding evaporator/kondenser (kg/m<sup>3</sup>)



- $V_1$  : volume sisi udara evaporator di *dry region* ( $m^3$ )  
 $V_2$  : volume sisi udara evaporator di *wet region* ( $m^3$ )  
 $V_{we}$  : volume sisi udara evaporator total ( $m^3$ )  
 $f$  : kecepatan aliran udara volumetris ( $m^3/s$ )  
 $h_{fg}$  : kalor laten dari vaporasi udara (kJ/kg)  
 $UA_1$  : perpindahan kalor keseluruhan di *dry region* evaporator ( $kW/^\circ C$ )  
 $UA_2$  : perpindahan kalor keseluruhan di *wet region* evaporator ( $kW/^\circ C$ )  
 $\omega_{air-in}$  : kelembaban spesifik udara di ruang pusat data (kg/kg)  
 $\omega_1$  : kelembaban spesifik udara keluaran evaporator (kg/kg)  
 $T_{air-in}$  : temperatur udara di ruang pusat data ( $^\circ C$ )  
 $T_1'$  : temperatur udara di antara *dry region* dan *wet region* evaporator ( $^\circ C$ )  
 $T_1$  : temperatur udara keluaran evaporator ( $^\circ C$ )  
 $T_{we}$  : temperatur dinding evaporator ( $^\circ C$ )  
 $M_{ref1}$  : aliran massa refrigeran di evaporator (kg/s) ( $M_{ref1} = M_{ref}$ )  
 $h_{ie}$  : entalpi di input evaporator (kJ/kg)  
 $h_{oe}$  : entalpi di output evaporator (kJ/kg)

### 2.3.2.3 Model Kondenser Kedua

Persamaan matematis model kondenser kedua mirip seperti persamaan matematis *dry region* evaporator.



Gambar 2.9. Skema Aliran Udara di Kondenser Kedua [4]

“telah diolah kembali”

Gambar di atas adalah skema diagram aliran udara di kondenser kedua, berbeda dari skema aliran udara evaporator karena kondenser hanya terdiri dari *dry region*.

Persamaan matematis kondenser kedua [5]:

$$C_{pu}\rho_u V_{wc2} \frac{dT_2}{dt} = C_{pu}\rho_u f(T_1 - T_2) + UA_3 \left( T_{wc2} - \frac{T_1 + T_2}{2} \right) \quad (2.40)$$

Persamaan matematis dinding kondenser kedua [5]:

$$C_{pw}\rho_w V_{wc2} \frac{dT_{wc2}}{dt} = UA_3 \left( \frac{T_1 + T_2}{2} - T_{wc2} \right) - M_{ref2}(h_{oc2} - h_{ic2}) \quad (2.41)$$

di mana

$V_{wc2}$  : volume sisi udara kondenser kedua ( $m^3$ )

$UA_3$  : perpindahan kalor keseluruhan di kondenser kedua ( $kW/^\circ C$ )

$T_2$  : temperatur udara keluaran kondenser kedua ( $^\circ C$ )

$T_{wc2}$  : temperatur dinding kondenser kedua ( $^\circ C$ )

$M_{ref2}$  : aliran massa refrigeran di kondenser kedua ( $kg/s$ ) ( $M_{ref2} = 10\%M_{ref}$ )

$h_{ic2}$  : entalpi di input kondenser kedua ( $kJ/kg$ )

$h_{oc2}$  : entalpi di output kondenser kedua ( $kJ/kg$ )

#### 2.3.2.4 Model Udara Masuk (*Supply Air*)

Pada gambar 2.7 terlihat bahwa udara yang keluar dari kondenser kedua melewati *fan* sebelum masuk ke kabinet. *Fan* memiliki panas yang dapat menyebabkan temperatur udara naik sedikit walaupun tidak signifikan dan dapat diabaikan. Akan tetapi, karena diasumsikan tidak terdapat beban kelembaban yang dihasilkan oleh *fan*, maka kelembaban spesifik setelah melewati *fan* ( $\omega_3$ ) dianggap sama dengan kelembaban spesifik keluaran kondenser sekunder ( $\omega_2$ ), yaitu

$$\omega_3 = \omega_2 = \omega_1 \quad (2.42)$$

Persamaan matematis model udara masuk (*supply air*) [4]:

$$T_3 = \frac{C_{pu}\rho_u f T_2 + \dot{Q}_{spl}}{C_{pu}\rho_u f} \quad (2.43)$$

di mana

- $T_3$  : temperatur udara setelah melewati *fan* (°C)  
 $\dot{Q}_{spl}$  : *heat* dari *fan* (kW)

### 2.3.2.5 Model Kabinet

Persamaan matematis temperatur kabinet [5]:

$$C_{pu}\rho_u V_{cab} \frac{dT_{cab}}{dt} = C_{pu}\rho_u f(T_3 - T_{cab}) + \dot{Q}_{load} \quad (2.44)$$

Persamaan matematis kelembaban kabinet [5]:

$$\rho_u V_{cab} \frac{d\omega_{cab}}{dt} = \rho_u f(\omega_3 - \omega_{cab}) + M \quad (2.45)$$

di mana

- $V_{cab}$  : volume kabinet (m<sup>3</sup>)  
 $T_{cab}$  : temperatur udara kabinet (°C)  
 $\omega_{cab}$  : kelembaban spesifik udara kabinet (kg/kg)  
 $\dot{Q}_{load}$  : beban *heat sensible* dari peralatan IT (kW)  
 $M$  : beban kelembaban di kabinet (kg/s)

Karena informasi keluaran yang dibutuhkan berupa kelembaban relatif, maka dibutuhkan persamaan umum yang dapat mengkonversi nilai kelembaban spesifik ( $\omega$ ) menjadi kelembaban relatif ( $\phi$ ) seperti berikut [4].

$$\phi = \frac{\omega P}{(0,622 + \omega) P_g} \quad (2.46)$$

dengan

$$P_g = 0,6108 \exp\left(\frac{17,27 T}{T + 237,3}\right) \quad (2.47)$$

di mana

- $P$  : tekanan atmosfer (kPa)  
 $P_g$  : tekanan uap saturasi (kPa)  
 $T$  : temperatur udara (°C)

## BAB 3 PERANCANGAN PENGENDALI MPC UNTUK SISTEM TATA UDARA PRESISI

### 3.1 Model Linier PAC

Model proses dari sistem tata udara presisi yang dipakai pada perancangan pengendali merupakan model ruang keadaan linier diskrit yang didapat dari hasil identifikasi sistem menggunakan metode N4SID pada penelitian seminar. Model ini didapat berdasarkan data diskrit masukan dan keluaran sistem dengan waktu pencuplikan lima detik. Berikut adalah matriks parameter sistemnya.

$$A = \begin{bmatrix} 0,9816 & 0,0274 & -0,0269 & 0,0011 & 3,9473 \times 10^{-6} & 1,5316 \times 10^{-4} & -6,9321 \times 10^{-5} & -3,1097 \times 10^{-4} \\ -0,0385 & 0,9105 & 0,0321 & -0,0677 & -9,8802 \times 10^{-4} & 3,2553 \times 10^{-4} & 8,44 \times 10^{-5} & -3,5519 \times 10^{-4} \\ 0,0827 & 0,2083 & 0,8197 & -0,1046 & 0,0019 & -3,7666 \times 10^{-4} & -2,8977 \times 10^{-4} & 3,1024 \times 10^{-4} \\ -0,0085 & -0,0064 & 9,377 \times 10^{-5} & 0,7924 & -0,001 & 6,8015 \times 10^{-4} & -6,7989 \times 10^{-5} & -0,0011 \\ 4,9635 \times 10^{-4} & 0,0012 & -0,0014 & -0,0031 & -0,0226 & 0,1276 & 0,0927 & 0,0499 \\ 2,4594 \times 10^{-4} & 4,9171 \times 10^{-4} & -3,9594 \times 10^{-4} & 0,0014 & -0,0801 & 0,3535 & 0,1349 & 0,1594 \\ -2,5528 \times 10^{-4} & -5,0208 \times 10^{-4} & 3,789 \times 10^{-4} & -0,0013 & -0,0225 & 0,4692 & -0,0607 & 0,2536 \\ -1,0567 \times 10^{-4} & -2,1003 \times 10^{-4} & 1,6487 \times 10^{-4} & -5,3548 \times 10^{-4} & -0,0383 & 0,0988 & 0,0436 & 0,0435 \end{bmatrix}$$

$$B = \begin{bmatrix} 1,6999 \times 10^{-4} & -0,0039 \\ 2,6895 \times 10^{-4} & -0,001 \\ -6,7814 \times 10^{-4} & 0,0128 \\ 1,409 \times 10^{-4} & -0,0093 \\ -3,7894 \times 10^{-6} & 4,3718 \times 10^{-5} \\ -2,3685 \times 10^{-6} & 7,7393 \times 10^{-5} \\ 2,4307 \times 10^{-6} & -7,7205 \times 10^{-5} \\ 1,0105 \times 10^{-6} & -3,2459 \times 10^{-5} \end{bmatrix}$$

$$C = \begin{bmatrix} -3,5504 & 0,2574 & -0,2214 & 0,0047 & 5,1729 \times 10^{-4} & -5,0581 \times 10^{-4} & 3,016 \times 10^{-4} & 6,8965 \times 10^{-4} \\ -0,1579 & -0,2073 & -0,1357 & -0,0626 & -4,8337 \times 10^{-5} & -9,3579 \times 10^{-5} & 4,3492 \times 10^{-5} & 1,9571 \times 10^{-4} \end{bmatrix}$$

$$D = \begin{bmatrix} -3,4789 \times 10^{-4} & 0,011 \\ 1,2083 \times 10^{-6} & -3,8263 \times 10^{-5} \end{bmatrix}$$

Model tersebut didapat saat kondisi sistem diberi beban sebesar 300 Watt ( $\dot{Q}_{load} = 300 W$ ). Saat proses identifikasi dilakukan, kondisi alat sebenarnya masih belum mencapai temperatur dan kelembaban yang diinginkan sebesar 20-22°C dan 45-55%. Berikut ini adalah data eksperimen yang menunjukkan kondisi sistem saat identifikasi dilakukan.

Tabel. Hasil Eksperimen Sistem PAC tanpa Beban dengan Kecepatan Kompresor 3.100 rpm dan Katup Kondenser Kedua Tertutup Penuh [6]

	temperatur	RH (%)
fresh air	29.3	42
suplay air	21.4	56,2

*Fresh air* adalah udara yang dibuang oleh kondenser pertama ke lingkungan dan *supply air* adalah udara yang masuk ke kabinet setelah melewati kondenser kedua. Berdasarkan tabel di atas, terlihat bahwa saat sistem tidak diberi beban, udara yang masuk ke kabinet sudah memiliki temperatur cukup tinggi dan kelembabannya juga sudah melewati spesifikasi ideal yang diinginkan. Oleh karena itu, wajar jika saat diberi beban, temperatur kabinet yang didapat sudah melewati spesifikasi yang diinginkan ( $T_{cab} > 22^{\circ}C$ ), begitu pula dengan kelembabannya.

Berdasarkan simulasi, model ruang keadaan yang didapat untuk beban 300 W dan dengan kecepatan kompresor sebesar 3.100 rpm, serta katup kondenser kedua terbuka penuh menghasilkan temperatur kabinet sebesar  $25,3842^{\circ}C$  dan RH kabinet sebesar 60,853%. Model tersebut memiliki nilai eigen sebagai berikut.

$$Ev = \begin{bmatrix} 0,9751 \\ 0,9497 \\ -0,2029 \\ 0,4909 \\ 0,0256 \\ 8,5633 \times 10^{-9} \\ 0,7897 \\ 0,7897 \end{bmatrix}$$

Terlihat bahwa semua nilai eigen dari sistem terletak di dalam *unit circle* yang menunjukkan bahwa sistem stabil. Selain itu, model linier ini juga *fully controllable* dan *fully observable*.

### 3.2 Uji Interaksi Antarvariabel Masukan-Keluaran

Uji interaksi ini dilakukan dengan mencari nilai RGA dari sistem untuk memilih pasangan masukan-keluaran yang tepat. Pertama, model ruang keadaan yang dipakai diubah ke dalam bentuk fungsi alih diskrit dengan bentuk  $G =$

$\begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$ . Mengubah model ruang keadaan menjadi fungsi alih bisa dilakukan dengan Matlab menggunakan fungsi  $tf()$ . Fungsi alih yang didapat adalah sebagai berikut:

$$G_{11} = (-0,0003479 + 0,0009447z^{-1} - 0,001002z^{-2} + 0,0005315z^{-3} - 0,0001267z^{-4} - 9,249 \times 10^{-6}z^{-5} + 8,76 \times 10^{-6}z^{-6} - 2,156 \times 10^{-7}z^{-7} + 1,576 \times 10^{-15}z^{-8}) / (1 - 3,818z^{-1} + 5,596z^{-2} - 3,777z^{-3} + 0,9804z^{-4} + 0,07616z^{-5} - 0,06004z^{-6} + 0,00147z^{-7} - 1,259 \times 10^{-11}z^{-8})$$

$$G_{12} = (0,01102 - 0,03137z^{-1} + 0,03337z^{-2} - 0,01575z^{-3} + 0,002417z^{-4} + 0,0005092z^{-5} - 0,0001755z^{-6} + 4,121 \times 10^{-6}z^{-7}) / (1 - 3,818z^{-1} + 5,596z^{-2} - 3,777z^{-3} + 0,9804z^{-4} + 0,07616z^{-5} - 0,06004z^{-6} + 0,00147z^{-7} - 1,259 \times 10^{-11}z^{-8})$$

$$G_{21} = (1,208 \times 10^{-6} - 4,002 \times 10^{-6}z^{-1} - 7,77 \times 10^{-6}z^{-2} + 2,385 \times 10^{-5}z^{-3} - 1,53 \times 10^{-5}z^{-4} + 1,072 \times 10^{-6}z^{-5} + 9,376 \times 10^{-7}z^{-6} - 2,443 \times 10^{-8}z^{-7}) / (1 - 3,818z^{-1} + 5,596z^{-2} - 3,777z^{-3} + 0,9804z^{-4} + 0,07616z^{-5} - 0,06004z^{-6} + 0,00147z^{-7} - 1,259 \times 10^{-11}z^{-8})$$

$$G_{22} = (-3,826 \times 10^{-5} - 0,0001726z^{-1} + 0,0005953z^{-2} - 0,0005575z^{-3} + 0,0001722z^{-4} + 9,343 \times 10^{-6}z^{-5} - 9,725 \times 10^{-6}z^{-6} + 2,398 \times 10^{-7}z^{-7}) / (1 - 3,818z^{-1} + 5,596z^{-2} - 3,777z^{-3} + 0,9804z^{-4} + 0,07616z^{-5} - 0,06004z^{-6} + 0,00147z^{-7} - 1,259 \times 10^{-11}z^{-8})$$

Selanjutnya nilai *gain* pada keadaan tunak dicari dengan perumusan seperti di bawah ini:

$$K_{11} = \lim_{z \rightarrow 1} G_{11} = -0,0333$$

$$K_{12} = \lim_{z \rightarrow 1} G_{12} = 0,8484$$

$$K_{21} = \lim_{z \rightarrow 1} G_{21} = -8,0842 \times 10^{-4}$$

$$K_{22} = \lim_{z \rightarrow 1} G_{22} = -0,0307$$

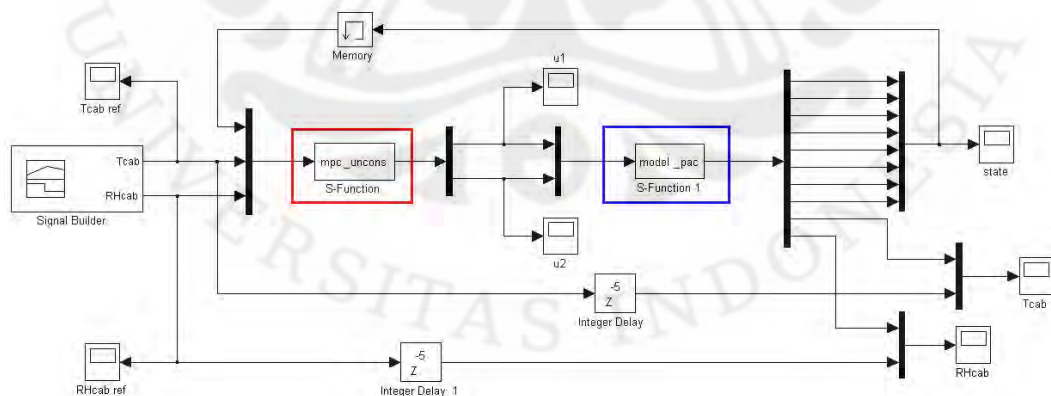
sehingga didapat matriks  $\mathbf{K} = \begin{bmatrix} -0,0333 & 0,8484 \\ -8,0842 \times 10^{-4} & -0,0307 \end{bmatrix}$ . Dengan menggunakan persamaan (2.33), nilai RGA:

$$\Lambda = \mathbf{K} .* (\mathbf{K}^T)^{-1} = \begin{bmatrix} 0,598 & 0,402 \\ 0,402 & 0,598 \end{bmatrix}$$

Nilai  $0,5 < \lambda_{11} < 1$  menunjukkan pasangan  $u_1$  dengan  $y_1$  sudah tepat, begitu pula  $u_2$  dengan  $y_2$ . Akan tetapi, jika melihat nilai kedua pasang *gain* relatif yang hampir sama, artinya  $u_2$  juga mempunyai pengaruh yang hampir sama besarnya dengan  $u_1$  terhadap  $y_1$  dan sebaliknya. Hal ini mengindikasikan terdapat interaksi atau efek *coupling* yang kuat antara pasangan  $u_1 - y_1$  dengan  $u_2 - y_2$ .

### 3.3 Perancangan MPC tanpa *Constraint* Menggunakan C-MEX *S-function*

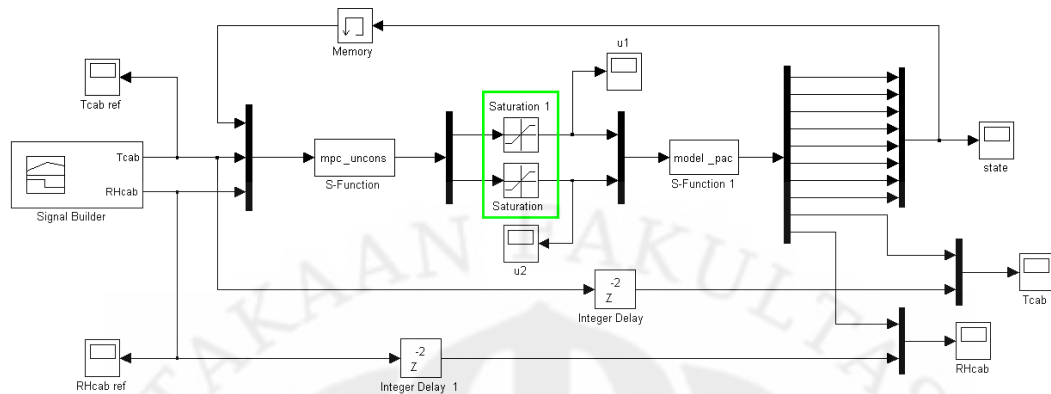
Pengendali MPC tanpa *constraint* dirancang untuk mengendalikan sistem tata udara presisi menggunakan *simulink* C-MEX *s-function*. Blok *s-function* yang digunakan ada dua, yaitu satu blok untuk pengendali MPC tanpa *constraint* dan satu blok untuk model sistem tata udara presisi. Untuk lebih jelasnya dapat dilihat pada gambar di bawah ini.



Gambar 3.1. Diagram Blok *Simulink* MPC tanpa *Constraint* tanpa Blok Saturasi

Pada gambar di atas, blok yang berwarna merah merupakan blok C-MEX *s-function* pengendali MPC tanpa *constraint* dan blok warna biru adalah blok C-MEX *s-function* untuk model sistem tata udara presisi. Perancangan di atas belum sesuai dengan kondisi fisik sebenarnya karena sinyal kendali yang dihasilkan

tidak dibatasi. Agar sesuai dengan kondisi fisik sebenarnya, perlu diberi blok saturasi untuk membatasi nilai sinyal kendali seperti gambar di bawah ini.



Gambar 3.2. Diagram Blok *Simulink* MPC tanpa *Constraint* dengan Blok Saturasi Blok yang dikotaki hijau adalah blok saturasi yang dipakai. Berikut ini adalah penjelasan algoritmanya dengan lebih lengkap.

### 3.3.1 Blok *S-function* MPC tanpa *Constraint*

Blok *s-function* ini berisi algoritma pengendali MPC tanpa *constraint* sampai menghasilkan nilai  $\Delta \underline{U}(k)_{opt}$  dan mengirim sinyal kendali  $\hat{u}(k|k)$  ke blok *s-function* model sistem tata udara presisi. Untuk perhitungan sampai mendapatkan  $\Delta \underline{U}(k)_{opt}$ , dibutuhkan informasi nilai keadaan  $\underline{x}(k)$  dan sinyal referensi  $\underline{T}(k)$  yang selalu di-*update* di setiap pencuplikan.

*S-function* yang dipilih dalam perancangan adalah C-MEX *s-function*. Bahasa C dipilih karena waktu pemrogramannya lebih cepat dibanding *m-file s-function*. Format C-MEX *s-function* terdiri dari beberapa fungsi dan fungsi yang dipakai untuk algoritma MPC tanpa *constraint* adalah *mdlInitializeSizes*, *mdlInitializeSampleTimes*, *mdlInitializeConditions*, *mdlOutputs*, *mdlUpdate* dan *mdlTerminate*.

Sebelum masuk ke setiap fungsi, *header* dan definisi awal perlu dimasukkan terlebih dahulu seperti layaknya bahasa C di antaranya definisi level *s-function*, nama *s-function*, memasukkan *header* yang diperlukan, definisi variabel tertentu seperti nilai  $H_p$ ,  $H_u$ , orde ( $n$ ), jumlah keluaran sistem ( $p$ ) dan jumlah masukan sistem ( $m$ ).



Pada pengendali MPC tanpa *constraint* ini, nilai  $H_p = H_u = 2$ ,  $n = 8$ ,  $p = 2$  dan  $m = 2$ . Selain itu, matriks ruang keadaan sistem juga didefinisi di bagian awal. Selanjutnya akan dibahas algoritma MPC tanpa *constraint* di setiap fungsi.

### 3.3.1.1 Fungsi *MdlInitializeSizes*

Fungsi ini merupakan inisialisasi awal dari blok *s-function* yang akan digunakan sebagai pengendali MPC tanpa *constraint*. Di fungsi ini perlu dimasukkan jumlah *port* masukan dan *port* keluaran blok, jumlah variabel keadaan yang akan dipakai pada blok, lebar tiap-tiap *port* masukan dan keluaran, jumlah waktu pencuplikan dan lain-lain.

Pada blok *s-function* MPC tanpa *constraint* ini, jumlah *port* masukan dan keluarannya adalah satu, dengan lebar *port* masukan sebesar sepuluh (delapan untuk nilai keadaan  $\underline{x}(k)$  dan dua untuk sinyal referensi  $\underline{T}(k)$ ) dan lebar *port* keluaran sebesar dua untuk sinyal kendali  $\hat{\underline{u}}(k|k)$ . Jumlah variabel keadaan blok yang dipakai bergantung pada nilai  $H_p$  dan  $H_u$  yang dipilih dengan urutan pada

array variabel keadaan blok sebagai berikut: 
$$\begin{bmatrix} \underline{x}(k) \\ \underline{T}(k) \\ \underline{u}(k-1) \\ \hat{\underline{u}}(k+i|k) \end{bmatrix} \text{ untuk } i = 0 \dots H_u - 1.$$

Jumlah waktu pencuplikan yang dipakai sebanyak satu.

### 3.3.1.2 Fungsi *MdlInitializeSampleTimes*

Di fungsi ini dimasukkan waktu pencuplikan yang akan dipakai pada blok *s-function* yang bersangkutan. Waktu pencuplikan yang dipakai bisa lebih dari satu tetapi harus sesuai dengan jumlah waktu pencuplikan yang telah diinisialisasi pada fungsi *mdlInitializeSizes* sebelumnya. Karena model proses didapat dari data diskrit dengan waktu pencuplikan lima detik, maka pada blok *s-function* ini waktu pencuplikannya juga lima detik ( $T_s = 5$  s) sesuai dengan waktu pencuplikan yang akan dipakai pada simulasi.

### 3.3.1.3 Fungsi *MdlInitializeConditions*

Fungsi ini digunakan untuk menginisialisasi kondisi-kondisi awal yang akan digunakan oleh blok *s-function*, yaitu variabel-variabel yang nilainya tetap dan tidak berubah setiap pencuplikan. Pada algoritma MPC tanpa *constraint* ini, variabel-variabel yang diinisialisasi dalam fungsi ini adalah matriks  $\underline{Q}$ ,  $\underline{R}$ ,  $\underline{C}_y\Psi$ ,  $\underline{C}_y\Gamma$ ,  $\underline{C}_y\Theta$  dan  $\underline{H}$ . Matriks-matriks tersebut tidak berubah di setiap pencuplikan karena tidak mengandung  $\underline{x}(k)$  dan  $\underline{T}(k)$  yang merupakan masukan blok *s-function* MPC tanpa *constraint* ini dan nilainya selalu di-update setiap pencuplikan.

Secara umum, algoritma MPC tanpa *constraint* pada fungsi *mdlInitializeConditions* dapat dilihat seperti di bawah ini.

1. Variabel keadaan blok *s-function* yang akan dipakai diisi nol untuk inisialisasi awal.
2. Membuat matriks  $\underline{Q}$  dan  $\underline{R}$ . Pada pengendali ini,  $\underline{Q}$  dan  $\underline{R}$  yang dipilih adalah

$$\underline{Q} = \begin{bmatrix} 0,5 & 0 & 0 & 0 \\ 0 & 0,5 & 0 & 0 \\ 0 & 0 & 0,5 & 0 \\ 0 & 0 & 0 & 0,5 \end{bmatrix} \text{ dan } \underline{R} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}.$$

3. Membuat matriks  $\underline{C}_y\Psi$ . Nilai  $\underline{C}_y\Psi$  yang didapat dengan  $H_p = 2$  adalah:

$$\underline{C}_y\Psi = \begin{bmatrix} -3,5134 & 0,0911 & -0,0778 & 0,0057 & -7,0418e-4 & -2,7654e-4 & 3,2309e-4 & 9,9048e-4 \\ -0,1577 & -0,2209 & -0,1136 & -0,0215 & 6,7601e-6 & -8,2621e-5 & 2,5803e-5 & 1,5082e-4 \\ -3,4588 & -0,0295 & 0,0336 & 0,0027 & -2,6659e-4 & -4,1351e-4 & 1,9432e-4 & 0,0011 \\ -0,1556 & -0,2289 & -0,096 & 0,0096 & 2,0766e-5 & -6,9251e-5 & 2,1163e-5 & 1,1593e-4 \end{bmatrix}$$

4. Membuat matriks  $\underline{C}_y\Gamma$ . Nilai  $\underline{C}_y\Gamma$  yang didapat dengan  $H_p = 2$  adalah:

$$\underline{C}_y\Gamma = \begin{bmatrix} -7,3138e-4 & 0,0217 \\ 1,8199e-6 & -3,5693e-4 \\ -0,0013 & 0,0343 \\ -1,0378e-5 & -7,6405e-4 \end{bmatrix}$$

5. Membuat matrik  $\underline{C}_y\Theta$ . Nilai  $\underline{C}_y\Theta$  yang didapat dengan  $H_p = 2$  dan  $H_u = 2$  adalah:

$$\underline{C}_y\Theta = \begin{bmatrix} -7,3138e-4 & 0,0217 & -3,4789e-4 & 0,011 \\ 1,8199e-6 & -3,5693e-4 & 1,2083e-6 & -3,8263e-5 \\ -0,0013 & 0,0343 & -7,3138e-4 & 0,0217 \\ -1,0378e-5 & -7,6405e-4 & 1,8199e-6 & -3,5693e-4 \end{bmatrix}$$

$\underline{C}_y\Theta$  yang sudah didapat kemudian ditranspos menghasilkan  $(\underline{C}_y\Theta)^T$ .

6. Mencari matriks  $\underline{H}$  dengan persamaan  $\underline{H} = (\underline{C}_y \underline{\Theta})^T \underline{Q} \underline{C}_y \underline{\Theta} + \underline{R}$  sehingga didapat:

$$\underline{H} = \begin{bmatrix} 2 & -2,9355e-5 & 5,8453e-7 & -1,7601e-5 \\ -2,9355e-5 & 2,0008 & -1,6304e-5 & 4,9156e-4 \\ 5,8453e-7 & -1,6304e-5 & 2 & -9,8556e-6 \\ -1,7601e-5 & 4,9156e-4 & -9,8556e-6 & 2,0003 \end{bmatrix}$$

Setelah itu, mencari invers  $\underline{H}$ :

$$\underline{H}^{-1} = \begin{bmatrix} 0,5 & 7,3346e-6 & -1,4605e-7 & 4,3979e-6 \\ 7,3346e-6 & 0,4998 & 4,0737e-6 & -1,2282e-4 \\ -1,4605e-7 & 4,0737e-6 & 0,5 & 2,4625e-6 \\ 4,3979e-6 & -1,2282e-4 & 2,4625e-6 & 0,4999 \end{bmatrix}$$

### 3.3.1.4 Fungsi *MdlUpdate*

Fungsi ini diisi oleh variabel-variabel yang nilainya berubah dan selalu di-*update* setiap waktu pencuplikan. Dalam algoritma MPC tanpa *constraint*, matriks-matriks  $\underline{x}(k)$ ,  $\underline{T}(k)$ ,  $\underline{u}(k-1)$ ,  $\underline{E}(k)$ ,  $\underline{G}(k)$ ,  $\Delta \hat{\underline{u}}_{opt}(k+i|k)$  dan  $\hat{\underline{u}}(k+i|k)$  (untuk  $i = 0, \dots, Hu-1$ ) dimasukkan ke dalam fungsi ini karena nilainya selalu di-*update* setiap pencuplikan.

Pada pencuplikan pertama ( $k = 0$ ), yaitu saat  $t = 0$ , semua nilai variabel yang dimasukkan ke fungsi ini masih bernilai nol. Untuk itu perhitungan yang diperlihatkan selanjutnya adalah pada pencuplikan kedua ( $k = 1$ ), yaitu saat  $t = 5$ . Urutan algoritmanya adalah sebagai berikut.

1. Mengambil nilai keadaan sekarang  $\underline{x}(k)$  dan nilai referensi  $\underline{T}(k)$  dari *port* masukan blok *s-function* pengendali MPC tanpa *constraint* ini. Nilai  $\underline{x}(k)$  dan  $\underline{T}(k)$  dimasukkan ke dalam variabel keadaan blok yang telah diinisialisasi pada *mdlInitializeSizes* dalam bentuk *array*. Nilai  $\underline{x}(k)$  didapat dari keluaran blok *s-function* model PAC dan nilai referensi didapat dari sumber sinyal berupa *pulse*.

Pada  $k = 1$  ini, nilai  $\underline{x}(1) = 0$  karena terdapat *delay* antara keluaran nilai keadaan  $\underline{x}(k)$  dari blok *s-function* model PAC dengan masukan  $\underline{x}(k)$  pada blok *s-function* MPC tanpa *constraint* akibat menggunakan blok *memory*.

Blok ini digunakan untuk mencegah terjadinya *algebraic loop*. Nilai  $\underline{T}(k)$

$$\text{adalah: } \underline{T}(1) = \begin{bmatrix} 25,681 \\ 0,6646 \\ 0 \\ 0 \end{bmatrix}.$$

2. Nilai referensi  $\underline{T}(k)$  dikurangi dengan simpangan keluaran dari sistem nonlinier dengan sistem liniernya sebesar  $y_0^1 = 25,3842$  dan  $y_0^2 = 0,60853$

$$\text{sehingga nilai } \underline{T}(1) = \begin{bmatrix} 0,2968 \\ 0,05607 \\ 0 \\ 0 \end{bmatrix}.$$

3. Meng-*update* nilai  $\underline{u}(k-1)$  yang merupakan nilai  $\hat{\underline{u}}(k|k)$  pada pencuplikan sebelumnya. Saat  $k = 1$ , nilai  $\underline{u}(0) = 0$ .
4. Menghitung matriks  $\underline{E}(k) = \underline{T}(k) - \underline{C}_y \underline{\Psi} \underline{x}(k) - \underline{C}_y \underline{\Gamma} \underline{u}(k-1)$ . Karena nilai  $\underline{x}(1) = 0$  dan nilai  $\underline{u}(0) = 0$ , maka:

$$\underline{E}(1) = \underline{T}(1) = \begin{bmatrix} 0,2968 \\ 0,05607 \\ 0 \\ 0 \end{bmatrix}.$$

5. Menghitung matriks  $\underline{G}(k)$  dengan persamaan  $\underline{G}(k) = 2(\underline{C}_y \underline{\Theta})^T \underline{Q} \underline{E}(k)$ . Matriks  $\underline{G}$  yang didapat pada pencuplikan ini:

$$\underline{G}(1) = \begin{bmatrix} -0,000217 \\ 0,006423 \\ -0,000103 \\ 0,003268 \end{bmatrix}.$$

6. Menghitung nilai  $\Delta \hat{\underline{u}}(k+i|k)$  optimal untuk  $i = 0, \dots, H_u - 1$  dengan  $\Delta \underline{U}(k)_{opt} = \frac{1}{2} \underline{H}^{-1} \underline{G}$  pada persamaan (2.19). nilai perubahan sinyal kendali optimal yang didapat:

$$\Delta \underline{U}(1) = \begin{bmatrix} -0,000054 \\ 0,001605 \\ -0,000026 \\ 0,000816 \end{bmatrix}.$$

7. Mendapatkan nilai sinyal kendali prediksi  $\hat{\underline{u}}(k+i|k)$  untuk  $i = 0, \dots, H_u - 1$  dengan menggunakan  $\underline{u}(k) = \Delta \underline{u}(k) + \underline{u}(k-1)$  pada persamaan (2.3) sehingga:

$$\underline{U}(1) = \Delta \underline{U}(1) = \begin{bmatrix} -0,000054 \\ 0,001605 \\ -0,000026 \\ 0,000816 \end{bmatrix}$$

karena nilai  $\underline{u}(0) = 0$ .

8. Memasukkan nilai sinyal kendali  $\hat{\underline{u}}(k|k)$  yang telah didapat ke *array* variabel keadaan blok di posisi  $\underline{u}(k - 1)$  untuk dijadikan sinyal kendali lampau pada pencuplikan berikutnya.
9. Nilai sinyal kendali  $\hat{\underline{u}}(k|k)$  ditambah dengan besar simpangan masukan antara sistem nonlinier dengan sistem liniernya sebesar  $u_0^1 = 51,667$  dan  $u_0^2 = 1$  sehingga  $\hat{\underline{u}}(1|1) = \begin{bmatrix} 51,666946 \\ 1,001605 \end{bmatrix}$ .
10. Menggeser nilai referensi ke elemen di bawahnya untuk membuat vektor sinyal referensi yang akan datang  $\underline{T}(k)$  yang akan dipakai pada pencuplikan

berikutnya sehingga nilai  $\underline{T}(k)$  menjadi  $\begin{bmatrix} 0,2968 \\ 0,05607 \\ 0,2968 \\ 0,05607 \end{bmatrix}$ .

Pada pencuplikan berikutnya, nilai referensi yang baru akan dimasukkan pada

elemen teratas sehingga bentuk matriks  $\underline{T}(k) = \begin{bmatrix} \underline{r}(k) \\ \underline{r}(k - 1) \\ \vdots \\ \underline{r}(k - (Hp - 1)) \end{bmatrix}$  berbeda

dengan  $\underline{T}(k)$  yang dijelaskan pada dasar teori. Strategi ini dipakai karena pada simulasi, nilai yang akan datang seperti  $\underline{r}(k + 1)$  sampai  $\underline{r}(k + Hp)$  tidak bisa diketahui. Oleh karena itu, pada model *simulink* ditambahkan blok *delay* sebesar  $Hp$  untuk menyesuaikan sinyal referensi dari sumber sinyal dengan keluaran sistem.

Algoritma MPC tanpa *constraint* pada fungsi *mdlUpdate* di atas diulang secara terus-menerus di setiap pencuplikan sampai simulasi berakhir. Berikutnya akan dibahas algoritma pada fungsi *mdlOutputs*.

### 3.3.1.5 Fungsi *MdlOutputs*

Fungsi ini berisi variabel-variabel yang nilainya akan dikeluarkan dari blok *s-function* pada setiap pencuplikan. Pada blok MPC tanpa *constraint* ini, nilai

yang akan dikeluarkan adalah sinyal kendali  $\underline{\hat{u}}(k|k)$  sebanyak dua nilai. Oleh karena itu, pada fungsi *mdlInitializeSizes* lebar *port* keluaran dari blok diisi senilai dua, yaitu dengan jumlah yang sama dengan jumlah masukan sistem tata udara presisi (*m*). Nilai ini nantinya akan digunakan sebagai sinyal masukan pada blok *s-function* model PAC.

### 3.3.1.6 Fungsi *MdlTerminate*

Fungsi ini wajib dimasukkan dalam setiap algoritma C-MEX *s-function* untuk menghentikan simulasi dari blok *s-function* yang bersangkutan. Setelah ini akan dibahas blok *s-function* untuk model PAC.

### 3.3.2 Blok *S-function* Model PAC

Sama halnya seperti blok *s-function* MPC tanpa *constraint*, blok *s-function* model PAC ini berisi persamaan (2.1) model proses untuk menghasilkan nilai keluaran sistem  $\underline{y}(k)$  dan mengirim nilai keadaan sekarang  $\underline{x}(k)$  ke blok *s-function* MPC tanpa *constraint*. Untuk perhitungan sampai mendapatkan  $\underline{y}(k)$  dan  $\underline{x}(k)$ , dibutuhkan sinyal masukan sistem berupa sinyal kendali  $\underline{u}(k)$  yang selalu di-update di setiap pencuplikan.

Fungsi-fungsi yang dipakai pada blok *s-function* model PAC sama seperti blok *s-function* MPC tanpa *constraint*, yaitu *mdlInitializeSizes*, *mdlInitializeSampleTimes*, *mdlInitializeConditions*, *mdlOutputs* dan *mdlTerminate*. Sebelum masuk ke dalam tiap fungsi, inisialisasi awal seperti mendefinisikan level C-MEX *s-function*, nama *s-function*, memasukkan *header*, definisi awal variabel yang akan digunakan seperti nilai *m*, *p* dan lain-lain, mengalokasikan *array* yang akan digunakan pada fungsi-fungsi *s-function* serta mendefinisikan model ruang keadaan sistem perlu dilakukan terlebih dahulu. Setelah itu, algoritma dalam tiap fungsi baru akan dibahas.

#### 3.3.2.1 Fungsi *MdlInitializeSizes*

Pada fungsi ini, jumlah *port* masukan dan *port* keluaran blok *s-function* model PAC ditentukan sebanyak satu dengan lebar masing-masing *port* adalah dua untuk *port* masukan (untuk sinyal kendali  $\underline{u}(k)$ ) dan sepuluh untuk *port*

keluaran (delapan untuk  $\underline{x}(k)$  dan dua untuk  $\underline{y}(k)$ ). Jumlah variabel keadaan blok ini ditentukan sebesar sepuluh dengan pembagian yang sama seperti pembagian lebar *port* keluaran blok dengan urutan elemen pada *array* variabel keadaan blok sebagai berikut:  $\begin{bmatrix} \underline{x}(k) \\ \underline{y}(k) \end{bmatrix}$ . Jumlah waktu pencuplikan yang dipakai adalah sebanyak satu.

### 3.3.2.2 Fungsi *MdlInitializeSampleTimes*

Waktu pencuplikan yang ditentukan pada fungsi ini adalah sebesar lima detik untuk blok *s-function* model PAC ini. Waktu pencuplikan ini sama dengan blok *s-function* MPC tanpa *constraint*, simulasi dan model proses yang dipakai.

### 3.3.2.3 Fungsi *MdlInitializeConditions*

Dalam fungsi ini, hanya dimasukkan inisialisasi awal dari variabel keadaan blok, yaitu dengan mengisi *array* variabel keadaan blok dengan nilai nol. Hal ini dimaksudkan agar nilai awal  $\underline{x}(k) = 0$  dan  $\underline{y}(k) = 0$ .

### 3.3.2.4 Fungsi *MdlUpdate*

Persamaan (2.1) model proses dimasukkan dalam fungsi ini. Variabel yang dihitung setiap pencuplikan adalah  $\underline{x}(k + 1)$  dan  $\underline{y}(k)$ . Pada pencuplikan pertama ( $k = 0$ ), yaitu saat  $t = 0$ , semua nilai variabel pada blok ini masih bernilai nol. Oleh karena itu, contoh algoritma perhitungan selanjutnya akan diambil pada saat pencuplikan kedua ( $k = 1$ ), yaitu saat  $t = 5$ .

1. Menghitung nilai  $\underline{x}(k + 1) = \underline{A} \underline{x}(k) + \underline{B} \underline{u}(k)$ . Nilai  $\underline{u}(k)$  terlebih dahulu dikurangi dengan besar simpangan masukan antara sistem nonlinier dengan sistem liniernya, yaitu  $u_0^1 = 51,667$  dan  $u_0^2 = 1$ . Nilai keadaan saat  $k = 1$  untuk  $\underline{x}(k + 1)$  adalah:

$$\underline{x}(2) = \begin{bmatrix} -0,004887 \\ -0,012862 \\ 0,022265 \\ 0,002066 \\ 0,000152 \\ 0,000045 \\ -0,000048 \\ -0,00002 \end{bmatrix}$$

dengan  $\underline{x}(1) = 0$ , yang didapat pada pencuplikan sebelumnya ( $k = 0$ ). Nilai

$\underline{u}(1)$  setelah dikurangi simpangan adalah  $\begin{bmatrix} -0,000054 \\ 0,001605 \end{bmatrix}$ .

2. Menghitung nilai  $\underline{y}(k) = \underline{C} \underline{x}(k) + \underline{D} \underline{u}(k)$  dengan nilai  $\underline{x}(1) = 0$  dan  $\underline{u}(1) = \begin{bmatrix} -0,000054 \\ 0,001605 \end{bmatrix}$ . Maka nilai  $\underline{y}(k) = \begin{bmatrix} 0,006957 \\ -0,000024 \end{bmatrix}$ .
3. Menambah nilai  $\underline{y}(k)$  dengan besar simpangan keluaran antara sistem nonlinier dengan sistem linier, yaitu  $y_0^1 = 25,3842$  dan  $y_0^2 = 0,60853$  sehingga  $\underline{y}(k) = \begin{bmatrix} 25,391157 \\ 0,608506 \end{bmatrix}$ .
4. Memasukkan nilai  $\underline{x}(k + 1)$  yang didapat ke dalam *array* variabel keadaan blok di posisi elemen  $\underline{x}(k)$  sehingga nilai  $\underline{x}(k + 1)$  di pencuplikan sekarang akan menjadi  $\underline{x}(k)$  di pencuplikan selanjutnya.

Algoritma di atas dilakukan berulang-ulang di setiap pencuplikan.

### 3.3.2.5 Fungsi *MdlOutputs*

Dalam fungsi ini, dimasukkan variabel dari nilai yang akan dikeluarkan di setiap pencuplikan, yaitu  $\underline{x}(k)$  dan  $\underline{y}(k)$ . Nilai  $\underline{x}(k)$  yang dikeluarkan merupakan hasil perhitungan  $\underline{x}(k + 1)$  karena nilai ini akan digunakan pada pencuplikan selanjutnya sebagai salah satu masukan blok *s-function* MPC tanpa *constraint* sebagai  $\underline{x}(k)$  pada pencuplikan selanjutnya. Nilai  $\underline{y}(k)$  dikeluarkan untuk dibandingkan dengan sinyal referensi  $\underline{T}(k)$ .

### 3.3.2.6 Fungsi *MdlTerminate*

Fungsi *mdlTerminate* merupakan fungsi yang wajib pada algoritma C-MEX *s-function* untuk menghentikan simulasi blok *s-function*.



### 3.4 Perancangan MPC dengan *Constraint* Menggunakan *M-file*

Pengendali MPC dengan *constraint* dirancang menggunakan *m-file* Matlab. *M-file* Matlab yang digunakan pada perancangan ini ada dua macam: satu *m-file* berisi fungsi kriteria yang dapat dipanggil pada program utama dan satu *m-file* lagi merupakan program utama yang berisi algoritma MPC dengan *constraint*.

Algoritma MPC dengan *constraint* pada dasarnya tidak jauh berbeda dengan tanpa *constraint*. Perbedaannya terletak pada matriks pertidaksamaan dan persamaan yang dibuat pada MPC dengan *constraint*. Berikut ini akan dibahas algoritma perancangan MPC dengan *constraint* yang diterapkan pada setiap *m-file*.

#### 3.4.1 Algoritma MPC dengan *Constraint* pada *M-file* 'mpc\_cons\_pac.m'

Program utama yang berisi algoritma MPC dengan *constraint* disimpan dalam *m-file* dengan nama 'mpc\_cons\_pac.m'. Algoritma MPC pada *m-file* ini adalah sebagai berikut.

1. Inisialisasi simulasi, yaitu menentukan waktu awal simulasi, waktu pencuplikan dan waktu akhir simulasi. Selain itu, jumlah masukan sistem ( $m$ ), jumlah keluaran sistem ( $p$ ) dan orde ( $n$ ) juga didefinisi di sini. Nilai  $\underline{x}(k)$  dan  $\underline{u}(k)$  mula-mula diset nol. Pada perancangan pengendali untuk PAC,  $t_{start} = 0$ ,  $T_s = 5$ ,  $t_f = 10.000$ ,  $n = 8$ ,  $m = 2$  dan  $p = 2$ .
2. Mendefinisi model linier diskrit sistem yang akan dipakai dan besar simpangan masukan dan keluaran antara sistem nonlinier dengan sistem linier PAC, yaitu  $u_0^1 = 51,667$  dan  $u_0^2 = 1$ , serta  $y_0^1 = 25,3842$  dan  $y_0^2 = 0,60853$ .
3. Menentukan nilai parameter pengendali MPC dengan *constraint* seperti  $H_p$ ,  $H_u$ ,  $\underline{Q}$ ,  $\underline{R}$  dan batasan-batasan untuk perubahan sinyal kendali (*slew rate*) dan nilai sinyal kendali. Pada perancangan MPC dengan *constraint* ini, nilai  $H_p$  dan  $H_u$ -nya sama dengan perancangan MPC tanpa *constraint* sebelumnya karena  $H_p = H_u = 2$  merupakan nilai *horizon* yang menghasilkan keluaran terbaik pada sistem PAC, sedangkan  $\underline{Q}$  dan  $\underline{R}$  akan disesuaikan untuk mendapat keluaran terbaik. Selain itu, batasan-batasan yang ditentukan adalah sebagai berikut:

$$-10 \leq \Delta u_1 \leq 10 \quad \text{dan} \quad -0,1 \leq \Delta u_2 \leq 0,1$$

$$41,667 \leq u_1 \leq 73,333 \quad \text{dan} \quad 0 \leq u_2 \leq 1$$

Batas nilai sinyal kendali  $u_1$  dan  $u_2$  kemudian dikurangi oleh  $u_0^1$  dan  $u_0^2$  sehingga batas nilai sinyal kendali tersebut menjadi:

$$-10 \leq u_1 \leq 21,666 \quad \text{dan} \quad -1 \leq u_2 \leq 0$$

Pada perancangan kali ini, keluaran sistem tidak dibatasi karena model dan alat sebenarnya saat diidentifikasi belum mencapai spesifikasi yang diinginkan, yaitu  $T_{cab}$  sebesar 20-22°C dan  $RH_{cab}$  sebesar 45-55%.

4. Vektor  $\underline{x}_{aug}$  yang dibuat dengan nilai  $Hu = 2$  adalah:

$$\underline{x}_{aug} = \begin{bmatrix} \Delta u_1(k|k) \\ \Delta u_2(k|k) \\ \Delta u_1(k+1|k) \\ \Delta u_2(k+1|k) \\ u_1(k|k) \\ u_2(k|k) \\ u_1(k+1|k) \\ u_2(k+1|k) \\ u_1(k-1) \\ u_2(k-1) \end{bmatrix}$$

dan diisi  $\underline{x}_{aug} = 0$  sebagai nilai awal.

5. Sinyal referensi  $T(k)$  diinisialisasi, kemudian nilai  $T(k)$  dikurangi simpangan keluaran  $y_0^1$  dan  $y_0^2$ .
6. Matriks  $\underline{C}_y\Psi$ ,  $\underline{C}_y\Gamma$  dan  $\underline{C}_y\Theta$  dibuat untuk dipakai pada perhitungan fungsi kriteria di *m-file* yang lainnya.
7. Membuat matriks persamaan dan pertidaksamaan dari batasan-batasan yang dipakai seperti pada persamaan (2.22) dan (2.23) sehingga didapat:

$$\underline{F} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \underline{F}_1 = \begin{bmatrix} 10 \\ 0,1 \\ 10 \\ 0,1 \\ 10 \\ 1 \\ 10 \\ 1 \\ 10 \\ 0,1 \\ 10 \\ 0,1 \\ 21,666 \\ 0 \\ 21,666 \\ 0 \end{bmatrix}$$

$$\underline{F}_{eq} = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 & 0 & 0 \end{bmatrix}, \underline{F}_{1eq} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

8. Memulai simulasi sistem lup tertutup dengan waktu simulasi seperti yang sudah didefinisi di no.1. Simulasi ini merupakan perhitungan algoritma yang dijalankan dalam lup selama 10.000 detik dengan pencuplikan setiap lima detik. Setiap pencuplikan, nilai variabel yang ditempatkan di dalam lup ini dihitung dan di-update terus-menerus. Algoritma dalam lup simulasi ini adalah:

- Menghitung keluaran dengan persamaan  $\underline{y}(k) = \underline{C} \underline{x}(k) + \underline{D} \underline{u}(k)$ .
- Menghitung nilai keadaan dengan persamaan  $\underline{x}(k + 1) = \underline{A} \underline{x}(k) + \underline{B} \underline{u}(k)$ .
- Minimasi fungsi kriteria menggunakan fungsi *fmincon* ( ). Untuk menggunakan *fmincon* ( ), salah satu argumen dari *fmincon* ( ) adalah fungsi kriteria yang dipakai. Oleh karena itu, *m-file* yang memuat persamaan fungsi kriteria dipanggil di sini. Dari sini didapatkan  $\Delta \underline{U}(k)_{opt}$ .
- Menghitung nilai sinyal kendali  $\underline{u}(k)$  menggunakan persamaan (2.3).
- Memasukkan nilai  $\underline{x}(k + 1)$  menjadi  $\underline{x}(k)$  untuk pencuplikan selanjutnya.

- Memasukkan nilai  $\underline{u}(k)$  menjadi  $\underline{u}(k - 1)$  untuk pencuplikan selanjutnya.

Algoritma dalam lup dimulasi tersebut dihitung mulai dari  $t_{start}$  sampai  $t_f$  setiap  $T_s$ .

### 3.4.2 Fungsi Kriteria pada *M-file* 'lossfct\_pac.m'

Fungsi kriteria yang dimasukkan pada *m-file* merupakan fungsi yang dibuat dengan memasukkan persamaan fungsi kriteria dan disimpan dalam *m-file* dengan nama 'lossfct\_pac.m' untuk dipanggil pada program utama 'mpc\_cons\_pac.m' sebagai argumen masukan dari fungsi *fmincon* ( ).

Fungsi ini mengambil nilai  $\underline{x}_{aug}$  sebagai argumen masukan dan nilai  $V(k)$  sebagai keluaran. Berikut akan dibahas algoritma pada *m-file* ini.

1. Mengambil nilai  $\Delta\underline{u}(k)$  dari  $\underline{x}_{aug}$ .
2. Menghitung nilai vektor  $\underline{E}(k) = \underline{T}(k) - \underline{C}_y \underline{\Psi} \underline{x}(k) - \underline{C}_y \underline{\Gamma} \underline{u}(k - 1)$ .
3. Menghitung nilai  $V(k) = (\underline{C}_y \underline{\Theta} \Delta\underline{U}(k) - \underline{E}(k))^T \underline{Q} (\underline{C}_y \underline{\Theta} \Delta\underline{U}(k) - \underline{E}(k)) + \Delta\underline{U}^T(k) \underline{R} \Delta\underline{U}(k)$ .

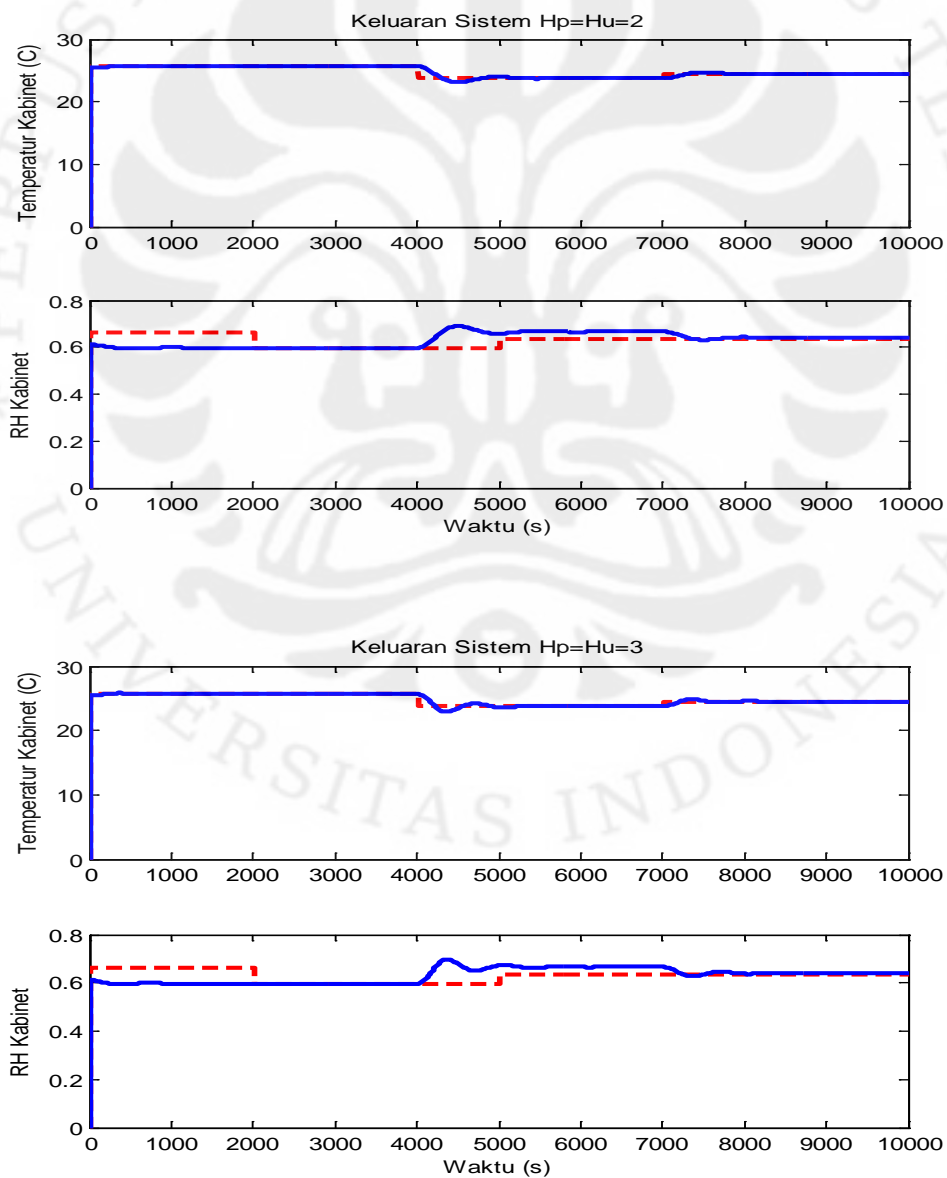
## BAB 4 HASIL SIMULASI DAN ANALISIS

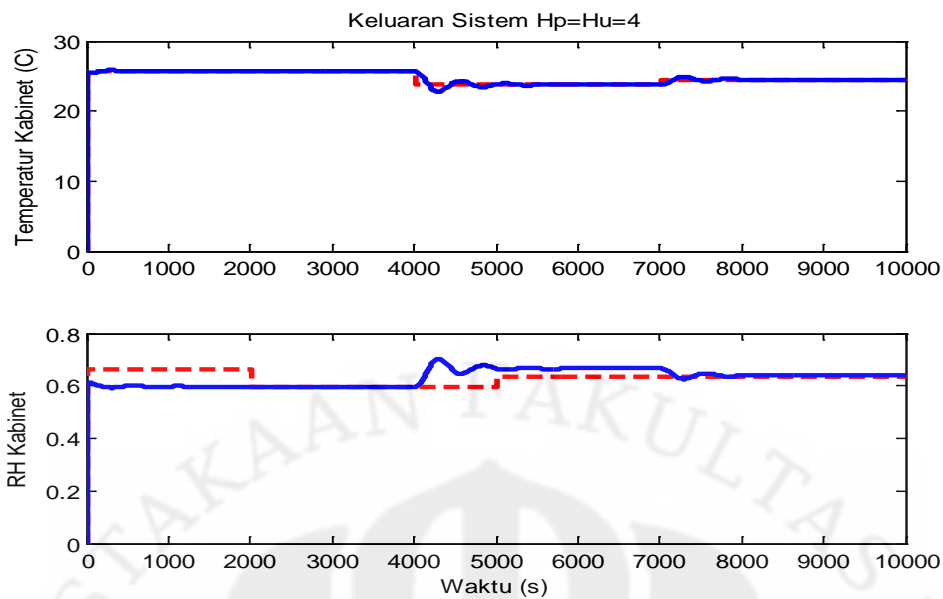
### 4.1 Pengendali MPC tanpa *Constraint*

#### 4.1.1 Hasil Pengendali tanpa Blok Saturasi

##### 4.1.1.1 Hasil Simulasi dan Analisis Variasi Nilai *Horizon*

Di bawah ini adalah hasil pengendalian MPC tanpa *constraint* dengan variasi nilai *horizon*  $H_p = H_u$  di mana matriks faktor bobot  $\underline{Q}$  dan  $\underline{R}$  dijaga tetap, yaitu dengan bobot  $Q = R = 1$ .





Gambar 4.1. Keluaran Sistem Hasil Pengendali MPC tanpa *Constraint* tanpa Blok Saturasi dengan Variasi Nilai *Horizon*

Pada gambar di atas, terlihat bahwa keluaran RH kabinet ( $y_2$ ) yang didapat tidak mengikuti sinyal referensi yang diberikan. Hal ini disebabkan karena model yang dipakai didapat berdasarkan persamaan matematis sistem PAC yang telah dijabarkan di subbab 2.3.2, dan jika ditinjau kembali, persamaan matematis untuk nilai RH kabinet adalah persamaan (2.45) sampai (2.47):

$$\rho_u V_{cab} \frac{d\omega_{cab}}{dt} = \rho_u f (\omega_3 - \omega_{cab}) + M$$

$$\phi_{cab} = \frac{\omega_{cab} P}{(0,622 + \omega_{cab}) P_g} \text{ dan } P_g = 0,6108 \exp\left(\frac{17,27 T_{cab}}{T_{cab} + 237,3}\right)$$

dengan nilai  $P$ ,  $\rho_u$ ,  $V_{cab}$ ,  $f$  dan  $M$  konstan. Terlihat bahwa nilai RH kabinet ( $\phi_{cab}$ ) bergantung pada nilai temperatur kabinet ( $T_{cab}$ ) dan kelembaban spesifik kabinet ( $\omega_{cab}$ ). Kelembaban spesifik kabinet sendiri bergantung pada nilai  $\omega_3$  di mana  $\omega_3$  didapat dari persamaan (2.42) dan (2.38a):

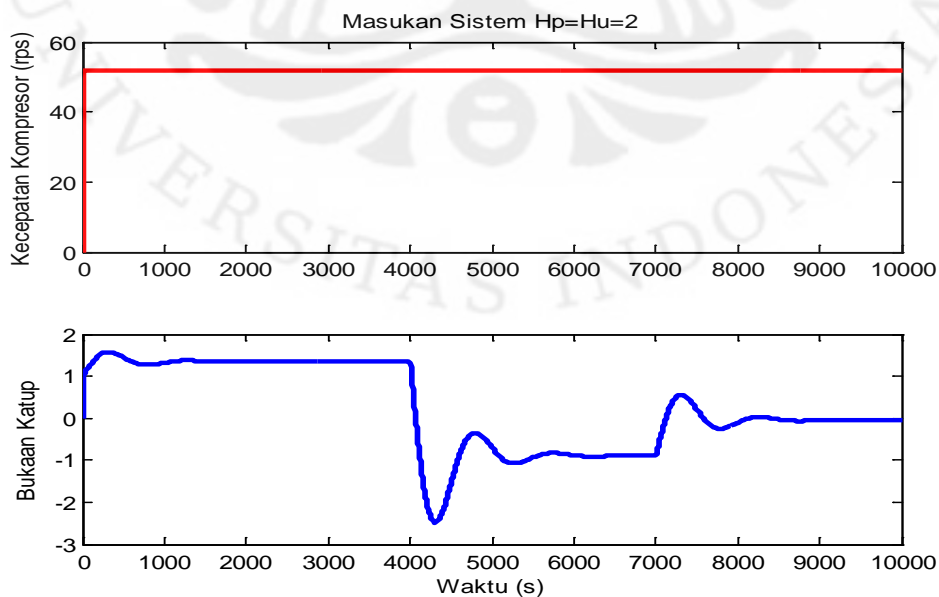
$$\omega_3 = \omega_2 = \omega_1$$

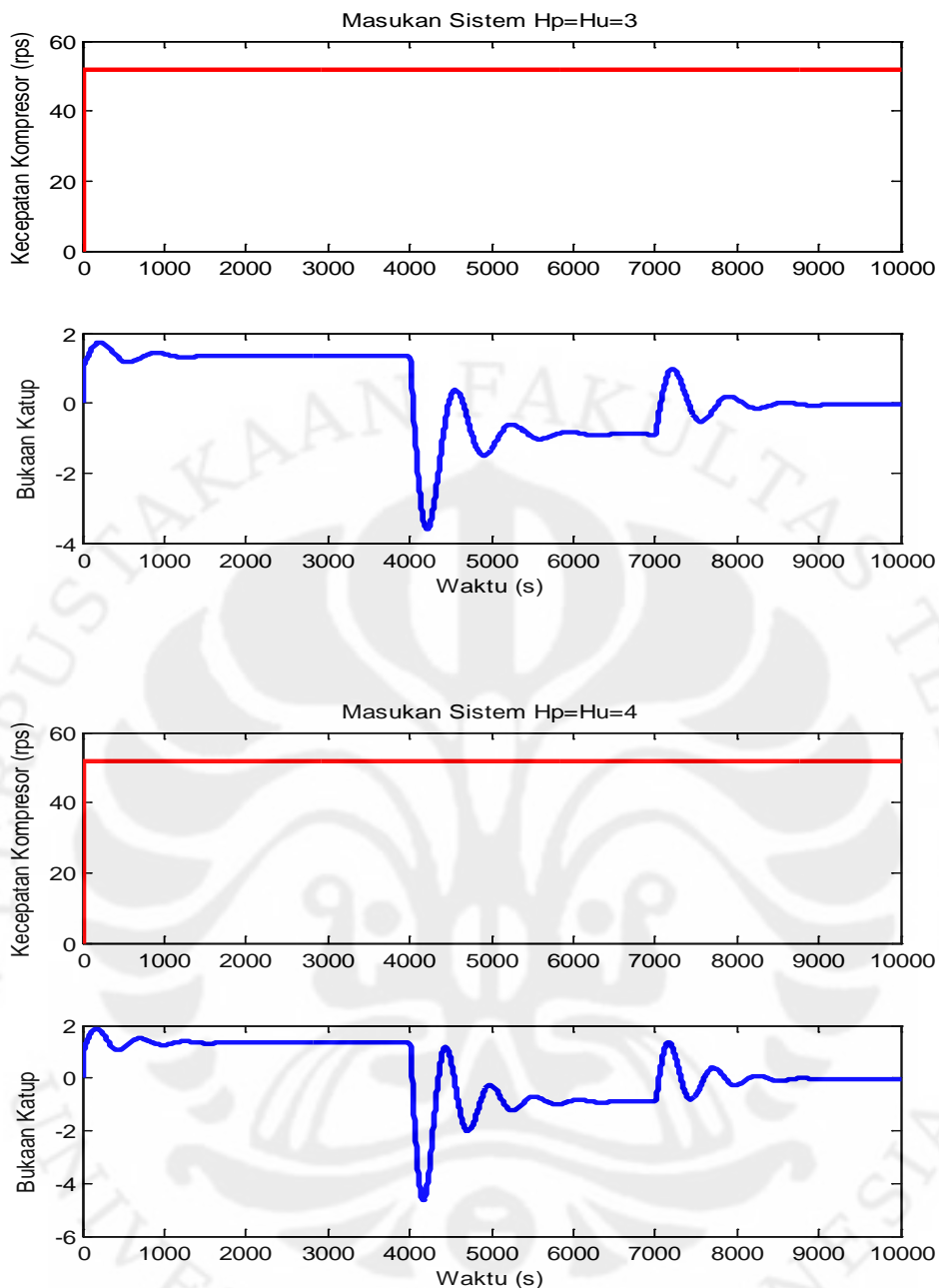
$$\omega_1 = (0,0198T_1^2 + 0,085T_1 + 4,4984)/1000$$

Terlihat bahwa nilai  $\omega_3$  juga bergantung pada nilai temperatur  $T_1$ .

Dari penjelasan persamaan di atas, jelas terbukti bahwa nilai RH kabinet sangat bergantung pada nilai temperatur kabinet, bukan pada nilai masukan sistem, sehingga keluaran sistem RH tidak mengikuti sinyal referensi yang diberikan, tetapi malah mengikuti keluaran temperatur kabinet ( $T_{cab}$ ). Jika temperatur kabinet sudah mencapai kondisi tunak dan mengikuti sinyal referensinya, maka nilai RH kabinet juga tidak akan berubah lagi mengikuti sinyal referensinya. Jika nilai  $T_{cab}$  berubah pada  $t = 4000$  dan  $t = 7000$  seperti gambar di atas, maka  $RH_{cab}$  juga ikut berubah di  $t = 4000$  dan  $t = 7000$  walaupun sinyal referensi  $RH_{cab}$  diset berubah pada  $t = 2000$  dan  $t = 5000$ . Hal ini juga mengakibatkan tidak terlihatnya efek *coupling* pada  $T_{cab}$  maupun  $RH_{cab}$  karena efek *coupling* hanya terlihat jika nilai salah satu keluaran berubah, sementara nilai keluaran lainnya tetap. Akan tetapi, pada kasus ini nilai  $RH_{cab}$  selalu berubah saat  $T_{cab}$  berubah dan konstan saat  $T_{cab}$  konstan sehingga efek *coupling* tidak terlihat.

Gambar 4.1 di atas juga menunjukkan bahwa nilai *horizon* yang lebih kecil menghasilkan sinyal keluaran dengan perubahan yang lebih mulus dan kecil mengikuti nilai referensinya sehingga *overshoot* yang terjadi lebih kecil. Oleh karena itu, nilai *horizon* yang paling baik untuk sistem tata udara presisi ini adalah sebesar  $H_p = H_u = 2$ . Hal ini bisa dilihat lebih jelas pada hasil sinyal kendalinya di bawah ini.





Gambar 4.2. Masukan Sistem Hasil Pengendali MPC tanpa *Constraint* tanpa Blok Saturasi dengan Variasi Nilai *Horizon*

Hasil sinyal kendali sistem di atas memperlihatkan bahwa semakin besar nilai *horizon*, semakin besar pula amplitudo dan perubahan sinyal kendali yang dihasilkan. Hal inilah yang menyebabkan  $H_p = H_u = 2$  menghasilkan keluaran sistem dengan perubahan yang lebih mulus dan *overshoot* yang lebih kecil karena amplitudo sinyal kendali yang dihasilkan dan perubahannya juga lebih kecil. Oleh karena itu,  $H_p = H_u = 2$  merupakan nilai *horizon* terbaik untuk sistem PAC ini.

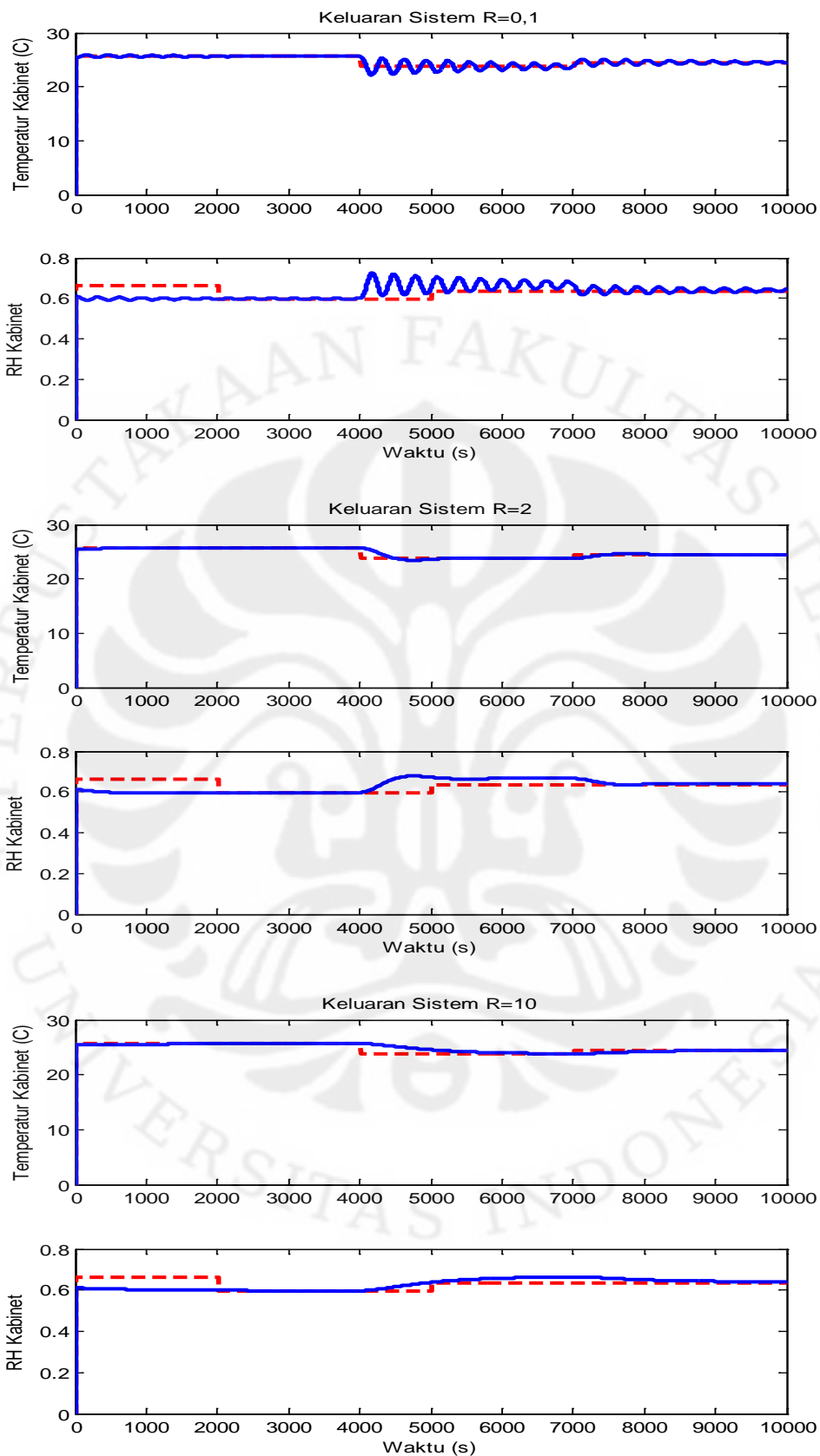


Selain itu, jika diperhatikan pada gambar 4.2 di atas, sinyal kendali kecepatan kompresor ( $u_1$ ) seolah-olah terlihat konstan sementara sinyal kendali bukaan katup ( $u_2$ ) terus berubah-ubah. Padahal sebenarnya, kecepatan kompresor juga berubah, hanya saja perubahannya kecil. Untuk mengetahui penyebab sinyal kendali yang dihasilkan adalah seperti pada gambar 4.2, hasil uji interaksi menggunakan RGA perlu diingat kembali.

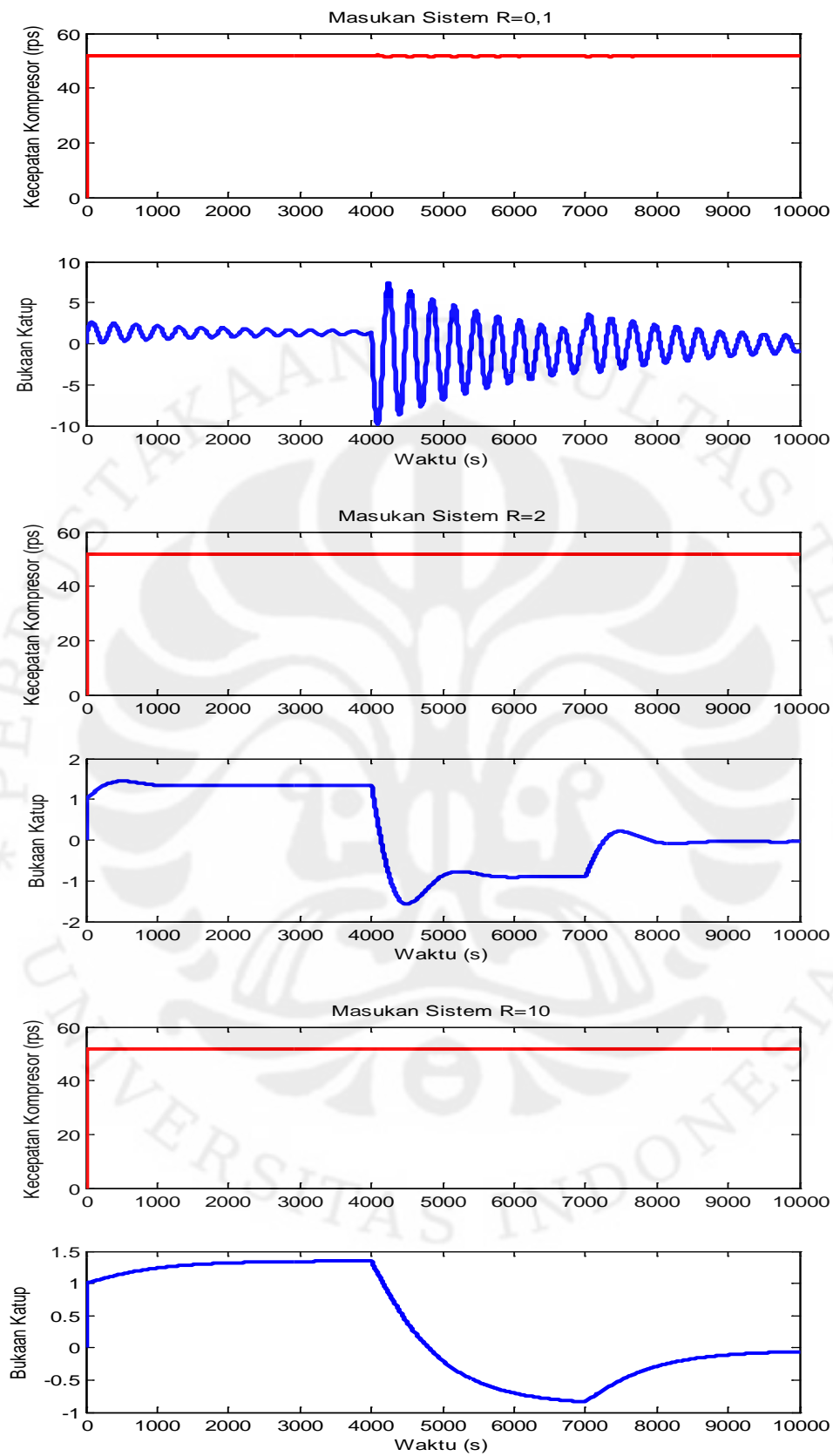
Hasil RGA menunjukkan pasangan masukan-keluaran yang dipakai adalah  $u_1 - y_1$  dan  $u_2 - y_2$ , tetapi masukan  $u_1$  dan  $u_2$  mempunyai pengaruh yang hampir sama kuatnya pada setiap keluaran sistem. Oleh karena itu, saat kecepatan kompresor ( $u_1$ ) dimasukkan untuk mengendalikan  $T_{cab}$  ( $y_1$ ) dan bukaan katup ( $u_2$ ) dimasukkan untuk mengendalikan  $RH_{cab}$  ( $y_2$ ), nilai  $u_2$  juga memberikan pengaruh untuk  $y_1$  dalam mengikuti sinyal referensinya karena  $u_2$  memiliki pengaruh yang hampir sama kuat dengan  $u_1$ .  $RH_{cab}$  yang mengikuti nilai  $T_{cab}$  juga menyebabkan  $u_2$  memberikan pengaruh kepada  $T_{cab}$  karena  $RH_{cab}$  bergantung pada  $T_{cab}$ . Hal ini terlihat dari perubahan sinyal bukaan kompresor yang berubah pada  $t = 4000$  dan  $t = 7000$  seperti  $T_{cab}$  karena  $RH_{cab}$  mengikuti  $T_{cab}$ . Karena itulah, saat  $u_2$  terus berubah-ubah karena  $RH_{cab}$  masih memiliki error terhadap sinyal referensinya, nilai  $u_2$  ini juga memberikan pengaruh pada  $T_{cab}$  dalam mengikuti sinyal referensinya sehingga perubahan nilai  $u_1$  yang dibutuhkan menjadi tidak besar.

#### 4.1.1.2 Hasil Simulasi dan Analisis Variasi Bobot Matriks R

Setelah memilih nilai *horizon* terbaik untuk sistem tata udara presisi ini, yaitu  $H_p = H_u = 2$ , selanjutnya yang harus dilakukan adalah memilih nilai bobot  $Q$  dan  $R$  terbaik. Nilai bobot  $R$  terbaik akan dicari terlebih dahulu dengan menentukan  $Q = 1$ .



Gambar 4.3. Keluaran Sistem Hasil Pengendali MPC tanpa *Constraint* tanpa Blok Saturasi dengan Variasi Nilai Bobot R

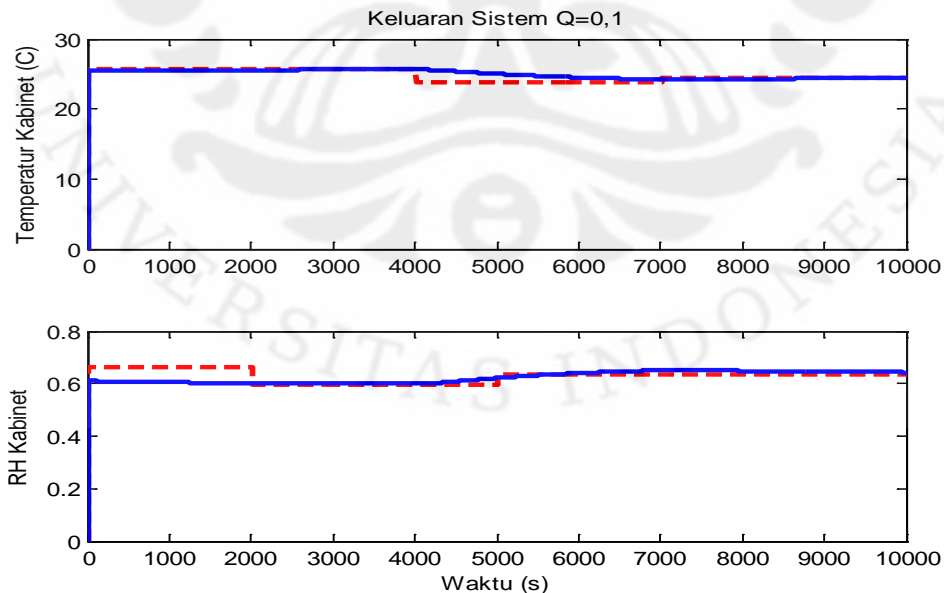


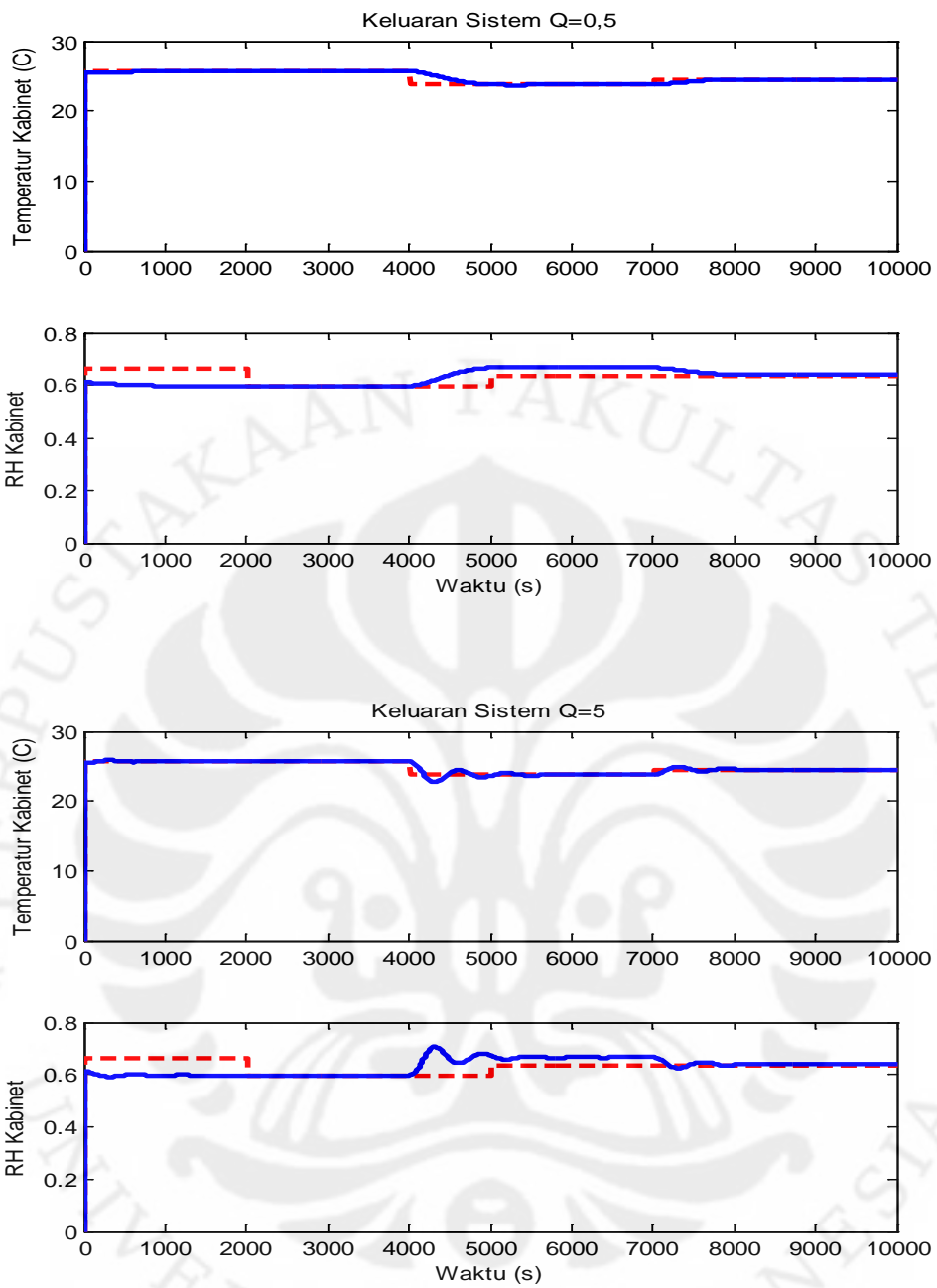
Gambar 4.4. Masukan Sistem Hasil Pengendali MPC tanpa *Constraint* tanpa Blok Saturasi dengan Variasi Nilai Bobot R

Hasil gambar di atas menunjukkan semakin besar nilai bobot  $R$  yang dipakai, semakin kecil perubahan nilai sinyal kendalinya di setiap pencuplikan. Hasil keluaran sistem sebelumnya dengan  $H_p = H_u = 2$ ,  $Q = 1$  dan  $R = 1$  menunjukkan keluaran sistem sedikit mengalami *overshoot* terhadap sinyal referensinya. Artinya, untuk meredam *overshoot* tersebut, nilai perubahan sinyal kendalinya harus dikecilkan agar setiap ada perubahan referensi keluaran sistem, perubahan nilai sinyal kendalinya tidak drastis sehingga tidak menghasilkan *overshoot*. Untuk itu, nilai  $R$  harus dipilih lebih besar dari satu dan berdasarkan gambar di atas,  $R = 2$  menghasilkan keluaran lebih baik dibandingkan  $R = 10$  karena keluaran sistem  $R = 10$  responnya lambat sehingga sebelum mencapai nilai referensi, nilainya sudah berubah lagi untuk mengikuti referensi baru. Oleh karena itu, nilai bobot  $R$  yang dipilih untuk sistem PAC ini adalah  $R = 2$ .

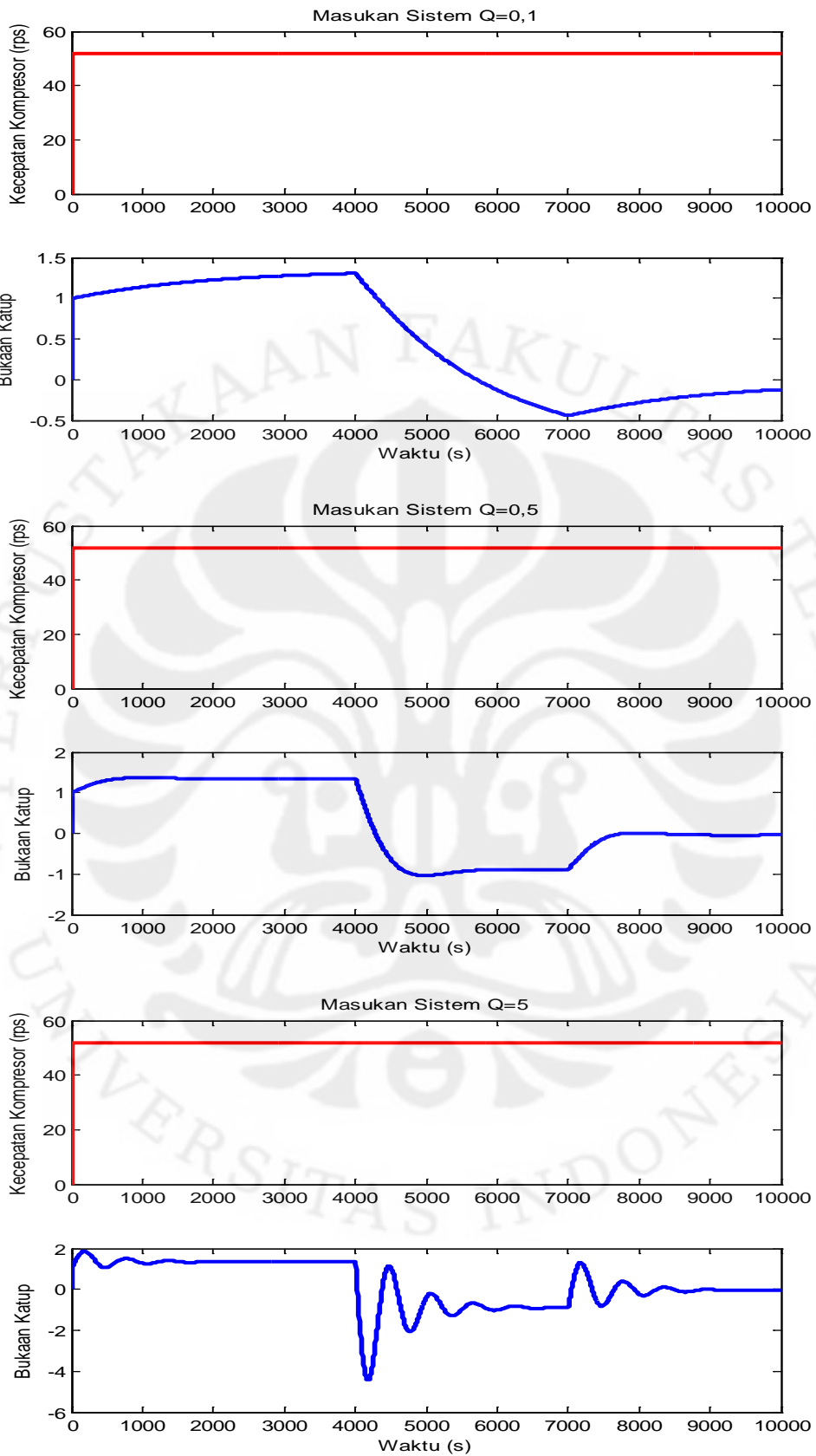
#### 4.1.1.3 Hasil Simulasi dan Analisis Variasi Bobot Matriks $Q$

Untuk pemilihan bobot matriks  $Q$  kali ini, nilai  $H_p = H_u = 2$  dan nilai  $R = 2$  dipilih sebagai parameter pengendali yang sudah ditentukan. Berikut adalah hasil keluaran dan masukan sistem berdasarkan variasi nilai bobot  $Q$ .





Gambar 4.5. Keluaran Sistem Hasil Pengendali MPC tanpa *Constraint* tanpa Blok Saturasi dengan Variasi Nilai Bobot Q



Gambar 4.6. Masukan Sistem Hasil Pengendali MPC tanpa *Constraint* tanpa Blok Saturasi dengan Variasi Nilai Bobot  $Q$

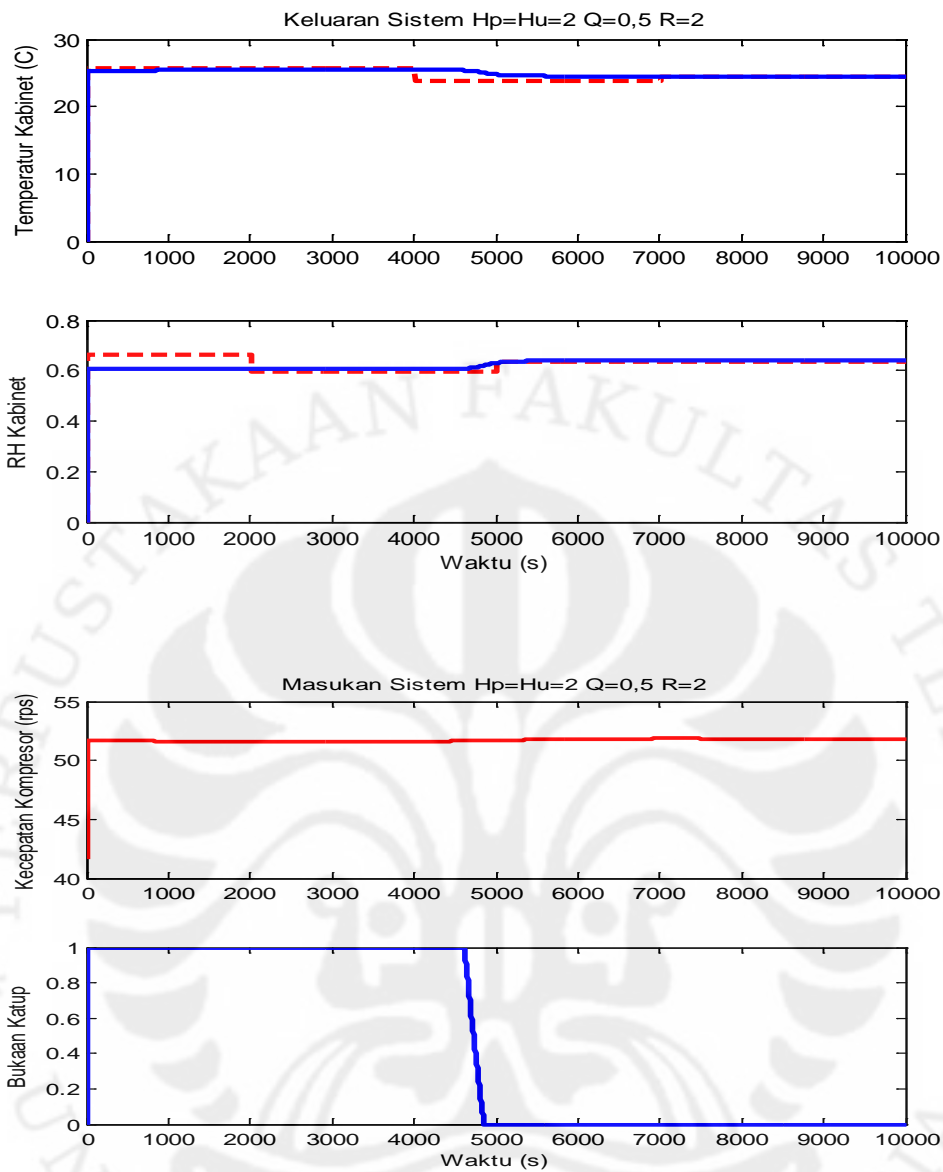
Berdasarkan gambar di atas, terlihat bahwa semakin besar nilai bobot  $Q$ , semakin besar pula perubahan nilai sinyal kendalinya. Menangkap hal ini, nilai bobot  $Q$  mempunyai pengaruh kebalikan dari nilai bobot  $R$  yang semakin besar nilainya malah semakin memperkecil nilai perubahan sinyal kendali di setiap pencuplikan.

Pada saat kondisi  $H_p = H_u = 2$ ,  $Q = 1$  dan  $R = 2$ , keluaran sistem masih sedikit memiliki *overshoot* terhadap sinyal referensinya. Untuk itu, dibutuhkan nilai bobot  $Q < 1$  untuk meredam *overshoot* tersebut. Pada gambar di atas, saat  $Q = 0,1$  respon keluaran menjadi sangat lambat untuk mengikuti sinyal referensinya karena perubahan sinyal kendali yang terlalu kecil.

Di sisi lain, saat  $Q = 0,5$  terlihat bahwa keluaran sistem sudah tidak memiliki *overshoot* pada setiap perubahan sinyal referensi. Oleh karena itu, nilai bobot  $Q$  yang dipilih di sini adalah  $Q = 0,5$ . Secara keseluruhan, nilai parameter pengendali MPC tanpa *constraint* terbaik untuk sistem tata udara presisi ini, yaitu  $H_p = H_u = 2$ ,  $Q = 0,5$  dan  $R = 2$ .

#### **4.1.2 Hasil Pengendali dengan Blok Saturasi**

Keluaran sistem hasil pengendali pada subbab 4.1.1 tanpa blok saturasi di atas terlihat bagus karena sinyal kendalinya tidak sesuai dengan kondisi fisik sebenarnya. Hal ini terlihat dari nilai bukaan katup yang mencapai nilai negatif. Padahal, bukaan katup tidak mungkin bernilai negatif. Untuk melihat hasil yang sesuai dengan kondisi fisik sebenarnya, sinyal kendali perlu dibatasi dengan menggunakan blok saturasi. Di halaman selanjutnya adalah hasil pengendali dengan blok saturasi menggunakan parameter pengendali terbaik, yaitu  $H_p = H_u = 2$ ,  $Q = 0,5$  dan  $R = 2$ .



Gambar 4.7. Keluaran dan Masukan Sistem Hasil Pengendali MPC tanpa *Constraint* dengan Blok Saturasi pada  $H_p = H_u = 2$ ,  $Q = 0,5$  dan  $R = 2$

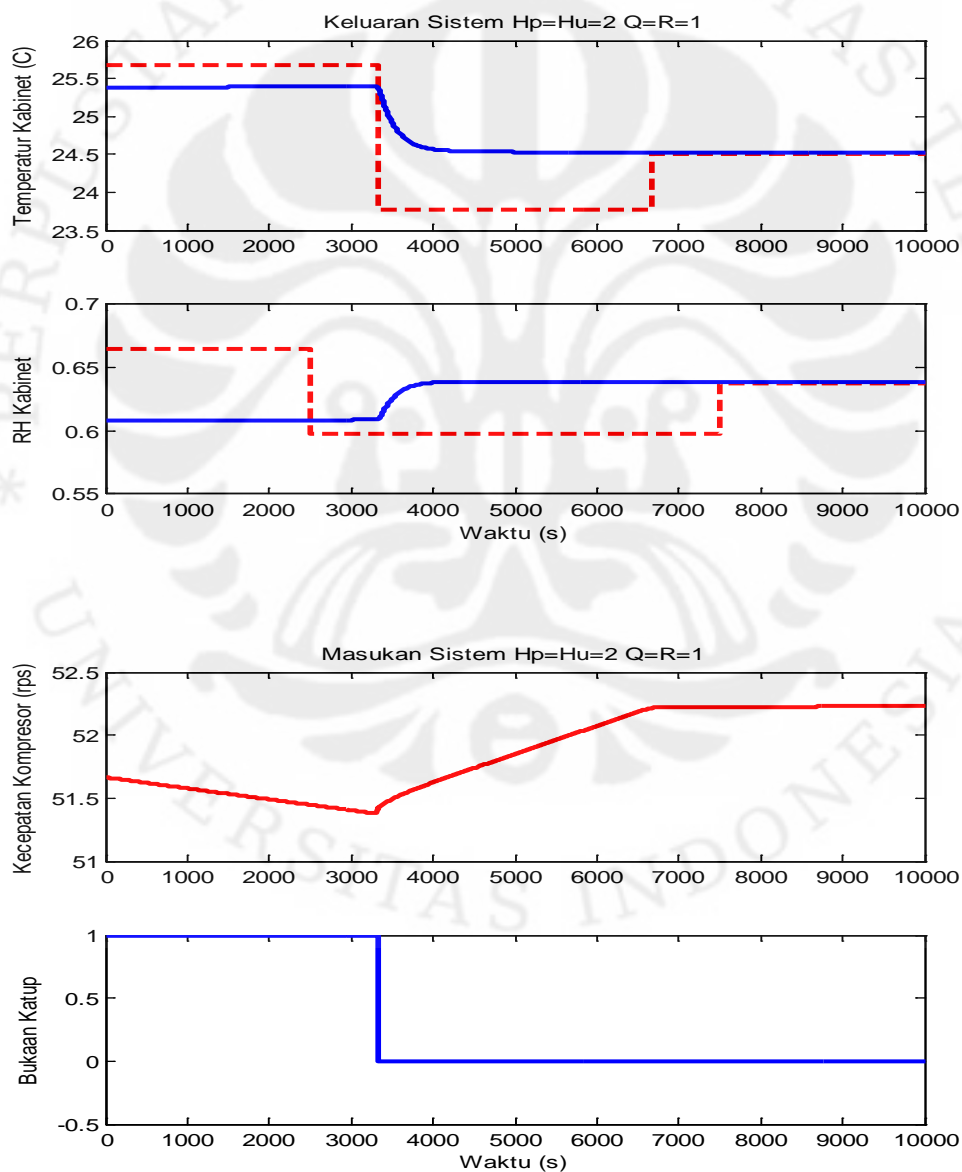
Gambar di atas menunjukkan bahwa keluaran temperatur kabinet tidak bisa mengikuti sinyal referensi jika sinyal kendali dibatasi sesuai kondisi fisik sebenarnya, yaitu 2.500-4.400 rpm untuk kecepatan kompresor dan 0-100% untuk bukaan katup. Hasil ini berbeda dengan saat sinyal kendali tidak diberi batasan sesuai kondisi fisik sebenarnya, di mana keluaran temperatur kabinet dapat mengikuti sinyal referensi. Pada kondisi pengendali menggunakan blok saturasi, terlihat bahwa hasil sinyal kendalinya sudah sesuai dengan kondisi fisik sebenarnya.



## 4.2 Pengendali MPC dengan *Constraint*

### 4.2.1 Perbandingan Hasil Pengendali MPC tanpa dan dengan *Constraint*

Sebelum melangkah lebih jauh melihat hasil pengendali MPC dengan *constraint*, sebagai permulaan akan diperlihatkan hasil pengendali MPC dengan *constraint* pada nilai  $H_p = H_u = 2$  dan  $Q = R = 1$ . Hasil MPC dengan *constraint* pada nilai parameter pengendali ini akan dianalisis dan dibandingkan dengan pengendali MPC tanpa *constraint* tanpa blok saturasi dengan nilai parameter pengendali yang sama. Di bawah ini adalah gambar hasil simulasinya.



Gambar 4.8. Keluaran dan Masukan Sistem Hasil Pengendali MPC dengan *Constraint* pada  $H_p = H_u = 2$  dan  $Q = R = 1$

Pada hasil pengendali MPC dengan *constraint* di atas, terlihat bahwa temperatur kabinet ( $T_{cab}$ ) tidak bisa mengikuti sinyal referensi dengan baik, berbeda dengan hasil pengendali MPC tanpa *constraint* dengan nilai parameter pengendali yang sama. Hal ini disebabkan karena amplitudo sinyal kendali pada MPC dengan *constraint* dibatasi dengan ketentuan sebagai berikut:

$$41,667 \leq u_1 \leq 73,333 \quad \text{dan} \quad 0 \leq u_2 \leq 1.$$

Oleh karena itu, tidak seperti pada pengendali tanpa *constraint* yang nilai bukaan katupnya mencapai 200% dan -200% sehingga nilai temperatur kabinetnya bisa mengikuti referensi, nilai bukaan katup pada pengendali dengan *constraint* terbatas antara 0-100% seperti terlihat pada gambar di atas.

Selain itu, karena nilai bukaan katup dibatasi, maka pengaruh bukaan katup terhadap temperatur kabinet juga berkurang dibandingkan pada pengendali tanpa *constraint* sehingga masukan kecepatan kompresor juga lebih bekerja untuk mengendalikan nilai temperatur kabinet agar sesuai dengan nilai referensinya. Hal ini menyebabkan perubahan sinyal masukan kecepatan kompresor lebih signifikan dibanding kecepatan kompresor pada pengendali tanpa *constraint* yang terlihat seolah-olah konstan.

Pada pengendali dengan *constraint* ini, nilai yang dibatasi bukan hanya amplitudo sinyal kendali, tapi juga perubahan sinyal kendalinya (*slew rate*) seperti yang ditentukan di bawah ini:

$$-10 \leq \Delta u_1 \leq 10 \quad \text{dan} \quad -0,1 \leq \Delta u_2 \leq 0,1.$$

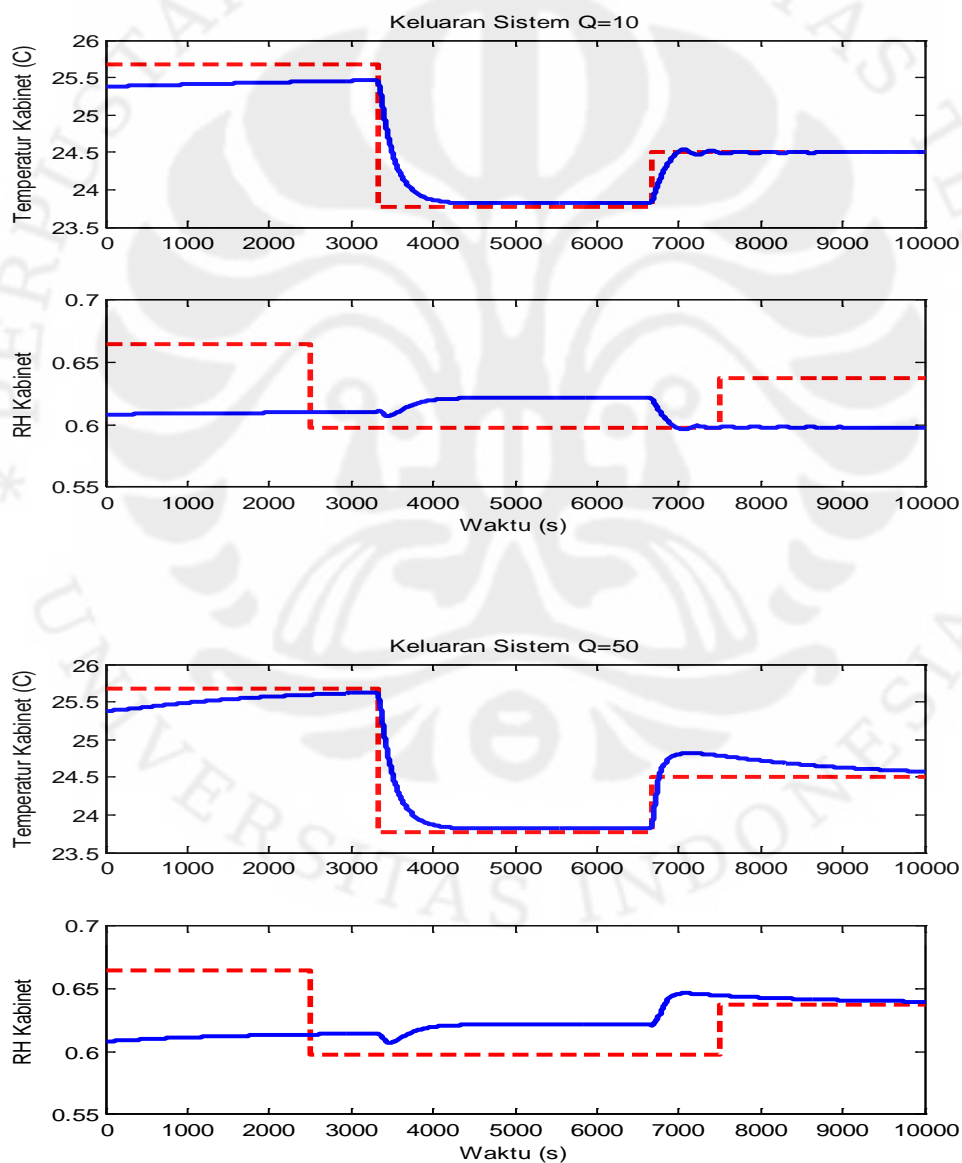
Akan tetapi, pada gambar masukan sistem di atas, perubahan bukaan katup pada  $t = 3340$  adalah sebesar satu melebihi batas  $\Delta u_2$  yang sudah ditentukan. Artinya, tidak terdapat solusi yang *feasible* untuk batasan  $\Delta u_2$ .

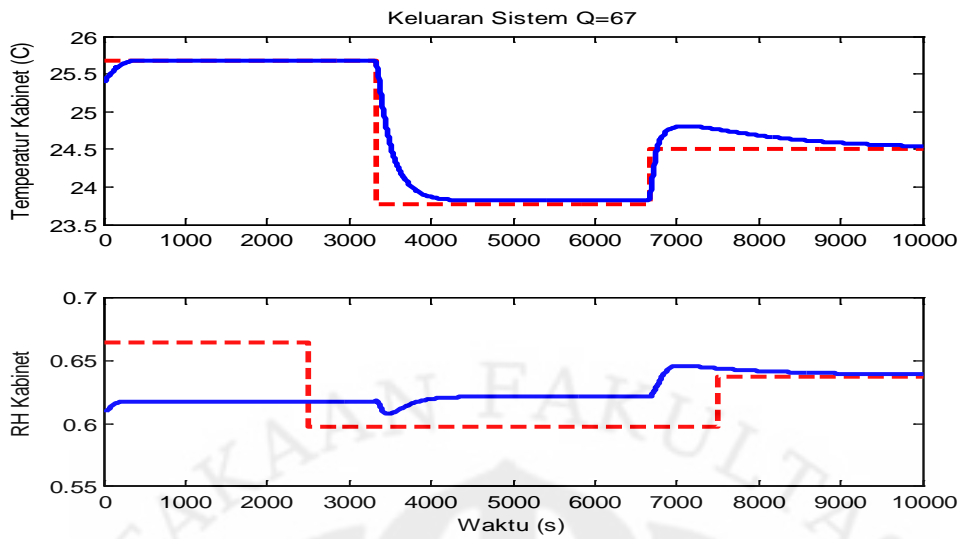
#### 4.2.2 Hasil Simulasi dan Analisis Variasi Bobot Matriks Q

Pada hasil MPC tanpa *constraint* sebelumnya, telah didapatkan nilai *horizon* terbaik sebesar  $H_p = H_u = 2$ . Oleh karena itu, pada pengendali MPC tanpa

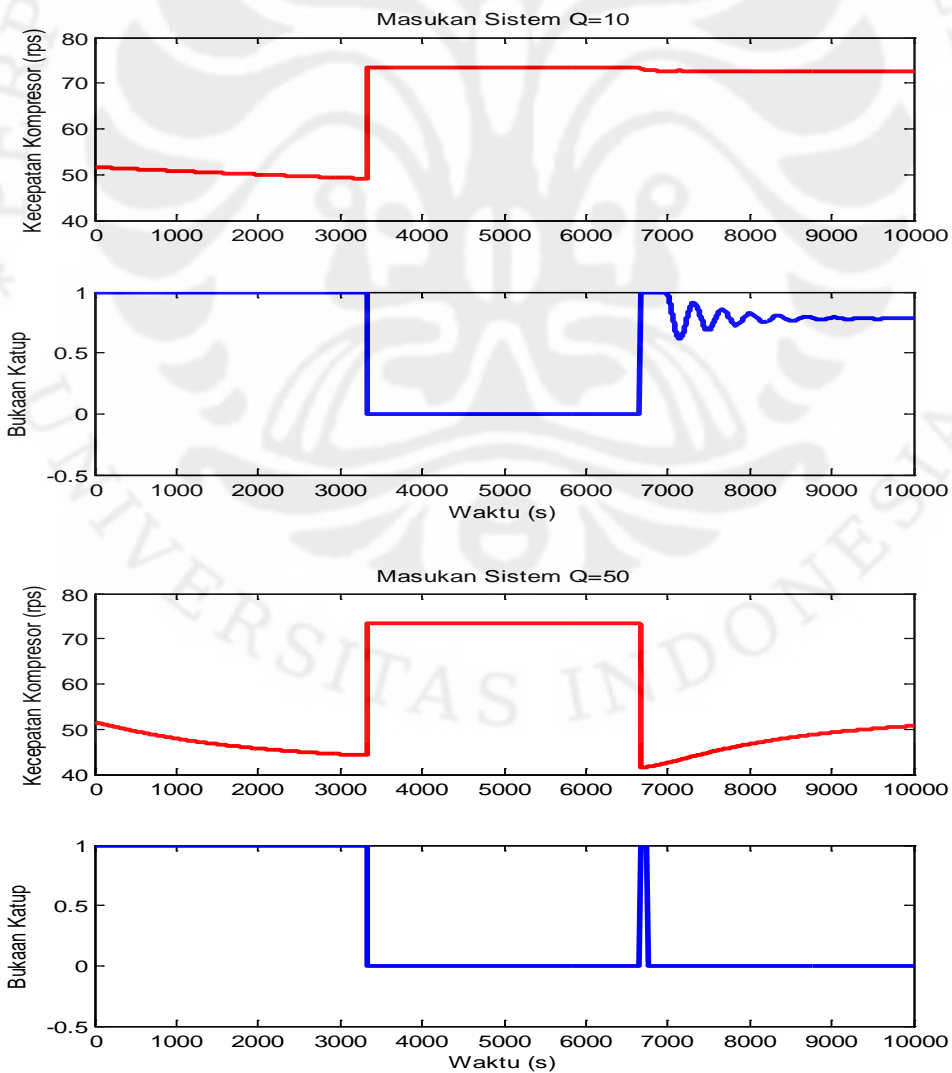
*constraint* ini, nilai *horizon* yang dipakai juga sama. Yang akan dicari selanjutnya adalah nilai bobot  $Q$  dan  $R$ .

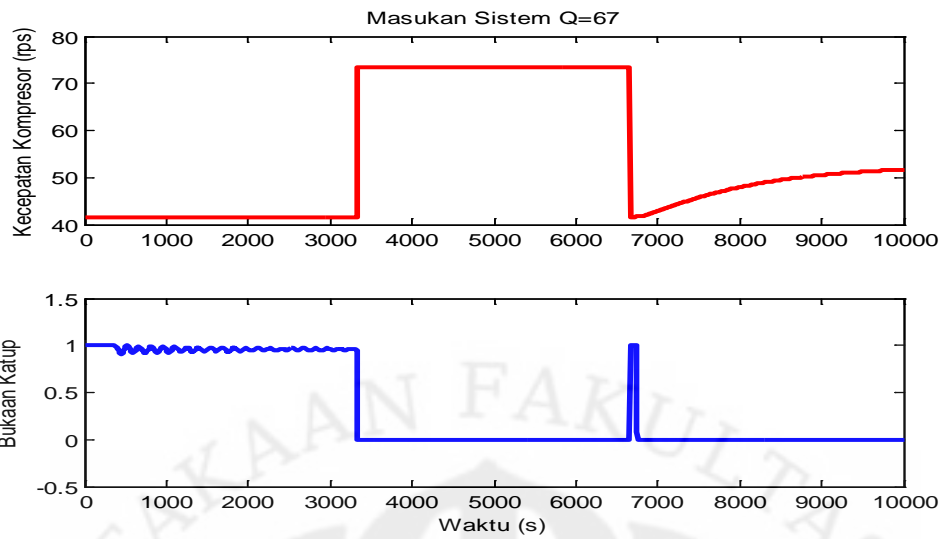
Dengan nilai bobot  $Q = R = 1$  di atas, terlihat bahwa sinyal keluaran sistem masih belum bisa mengikuti sinyal referensi yang diberikan. Artinya, perubahan sinyal kendali yang diberikan harus lebih besar dibanding sebelumnya. Untuk memperbesar perubahan sinyal kendali, dapat dilakukan dengan memperbesar bobot  $Q$  atau mengecilkan bobot  $R$ . Pertama, yang akan dilakukan adalah menjaga nilai  $R = 1$  dan mengubah nilai bobot  $Q$ .





Gambar 4.9. Keluaran Sistem Hasil Pengendali MPC dengan *Constraint* dengan Variasi Nilai Bobot Q

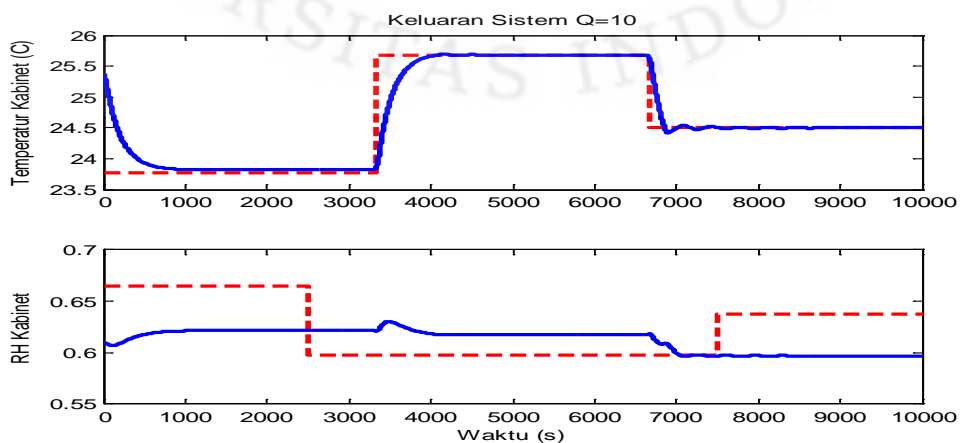


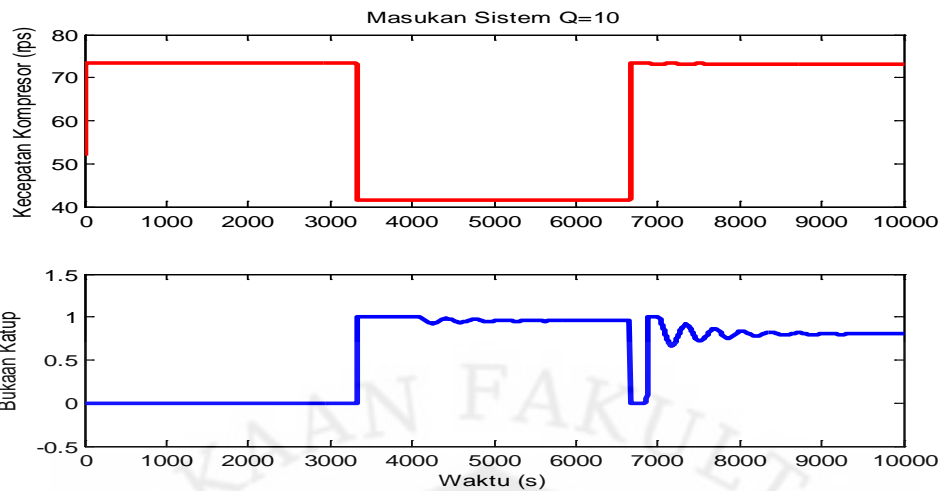


Gambar 4.10. Masukan Sistem Hasil Pengendali MPC dengan *Constraint* dengan Variasi Nilai Bobot  $Q$

Pada hasil simulasi dengan variasi nilai bobot  $Q$  di atas, terlihat bahwa saat  $Q = 10$  dan  $Q = 50$ , nilai keluaran  $T_{cab}$  masih belum mengikuti sinyal referensi di bagian awal karena responnya yang lambat. Nilai  $Q$  terbaik adalah  $Q = 67$  di mana keluaran sudah bisa mengikuti sinyal referensi di bagian awal dan respon keluaran sudah lebih cepat.

Nilai  $Q$  yang didapat sangat besar karena pada dasarnya sistem tata udara presisi ini memiliki respon yang lambat dengan perubahan nilai hanya setiap lima detik, sedangkan di simulasi ini keluaran temperatur kabinet diharapkan sudah mencapai nilai  $25,681^{\circ}\text{C}$  sesuai sinyal referensi dalam waktu sekitar 3.000 detik. Jika nilai awal referensi yang diinginkan lebih rendah, maka nilai bobot  $Q$  yang diperlukan juga tidak besar. Hal tersebut bisa dilihat pada gambar di bawah ini.





Gambar 4.11. Keluaran dan Masukan Sistem Hasil Pengendali MPC dengan *Constraint* dengan Nilai Awal Referensi  $r_1 = 23,778$

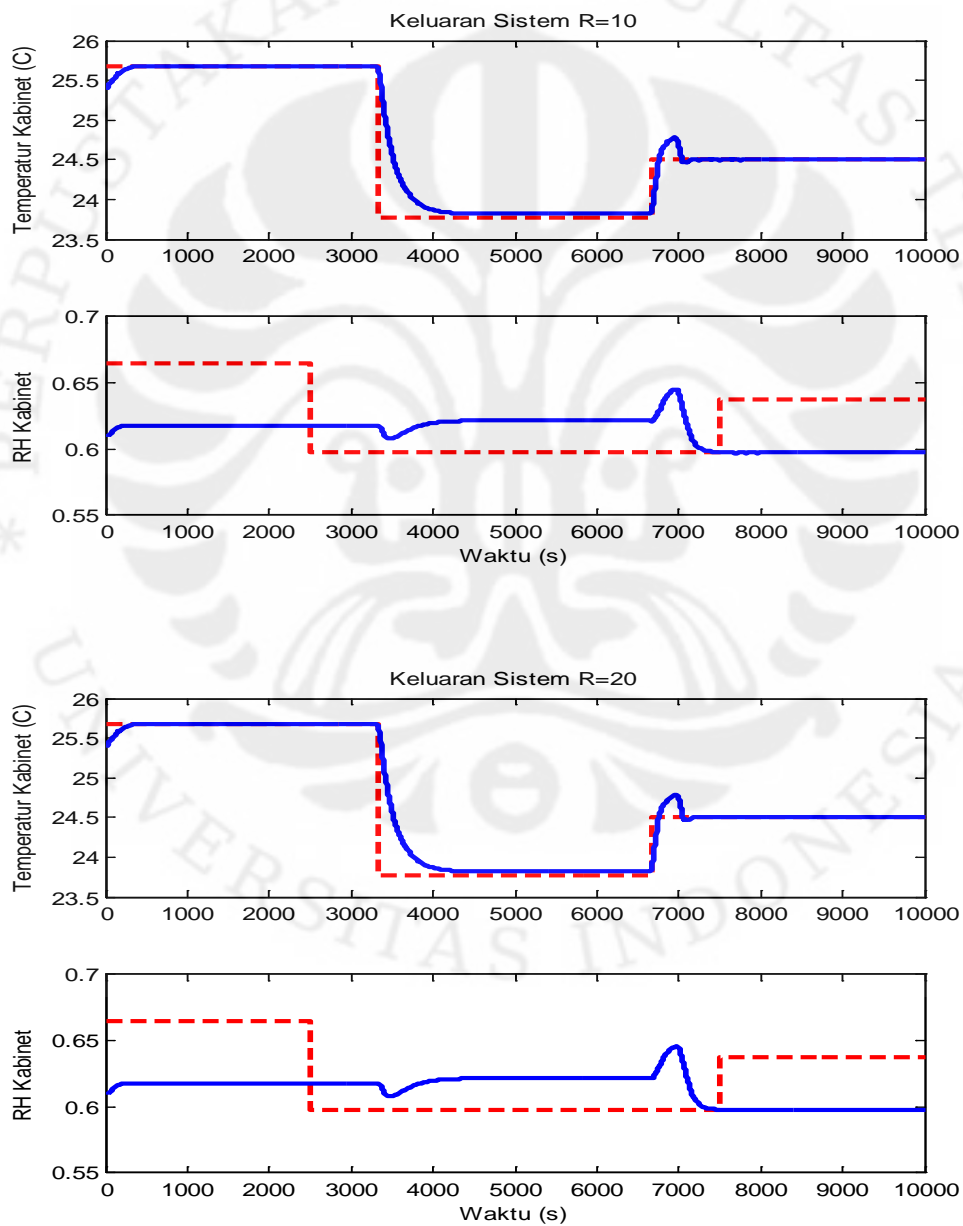
Gambar 4.11 di atas menunjukkan bahwa dengan nilai parameter  $H_p$ ,  $H_u$  dan  $R$  yang sama, dengan nilai awal referensi yang berbeda, maka nilai bobot  $Q$  yang dibutuhkan juga berbeda. Nilai bobot  $Q$  yang dibutuhkan saat nilai awal referensi  $r_1 = 23,778$  adalah  $Q = 10$ , sedangkan saat nilai awal referensi  $r_1 = 25,681$  adalah  $Q = 67$ . Namun, nilai bobot  $Q$  yang dipilih untuk sistem ini tetap  $Q = 67$  untuk mengantisipasi semua nilai sinyal referensi yang masih mungkin diikuti oleh keluaran sistem.

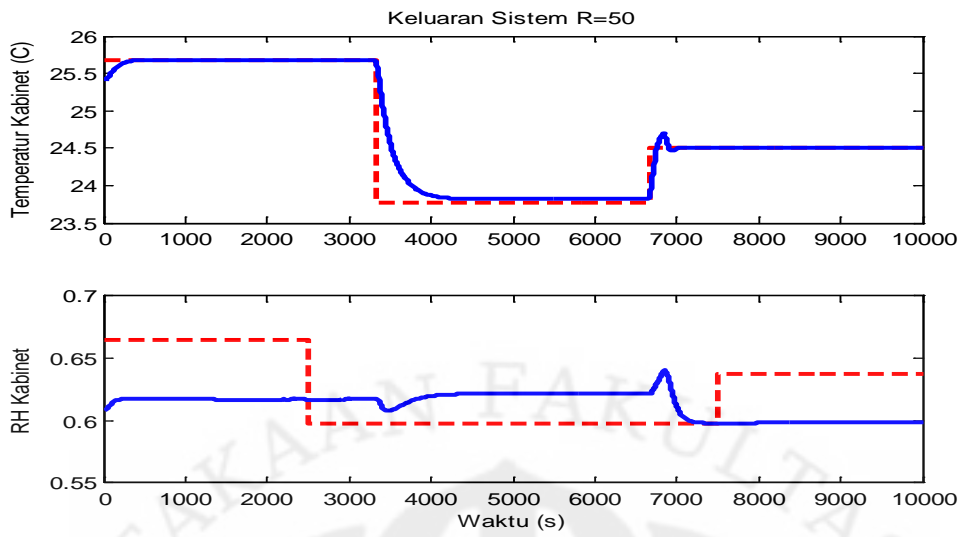
Selain itu, jika hasil simulasi di atas diperhatikan secara seksama, terlihat bahwa keluaran sistem  $T_{cab}$  tidak bisa mencapai nilai  $23,778^\circ\text{C}$  seperti sinyal referensi dan memiliki selisih terhadap referensinya. Hal ini berarti kondisi sistem dengan beban 300 W memang belum dapat mencapai nilai temperatur tersebut. Temperatur terendah yang bisa dicapai adalah sebesar  $23,8149^\circ\text{C}$  sehingga terdapat error antara nilai keluaran tunak dengan nilai referensi.

Jika melihat masukan sistem hasil pengendali di atas, tampak jelas bahwa batasan perubahan sinyal kendali (*slew rate*) untuk kecepatan kompresor  $u_1$  di sini juga sudah tidak memiliki solusi yang *feasible*. Hal ini disebabkan besarnya nilai bobot  $Q$  yang dipakai menuntut perubahan sinyal kendali yang juga besar sehingga batasan-batasan  $\Delta u_1$  dan  $\Delta u_2$  menjadi tidak berlaku lagi.

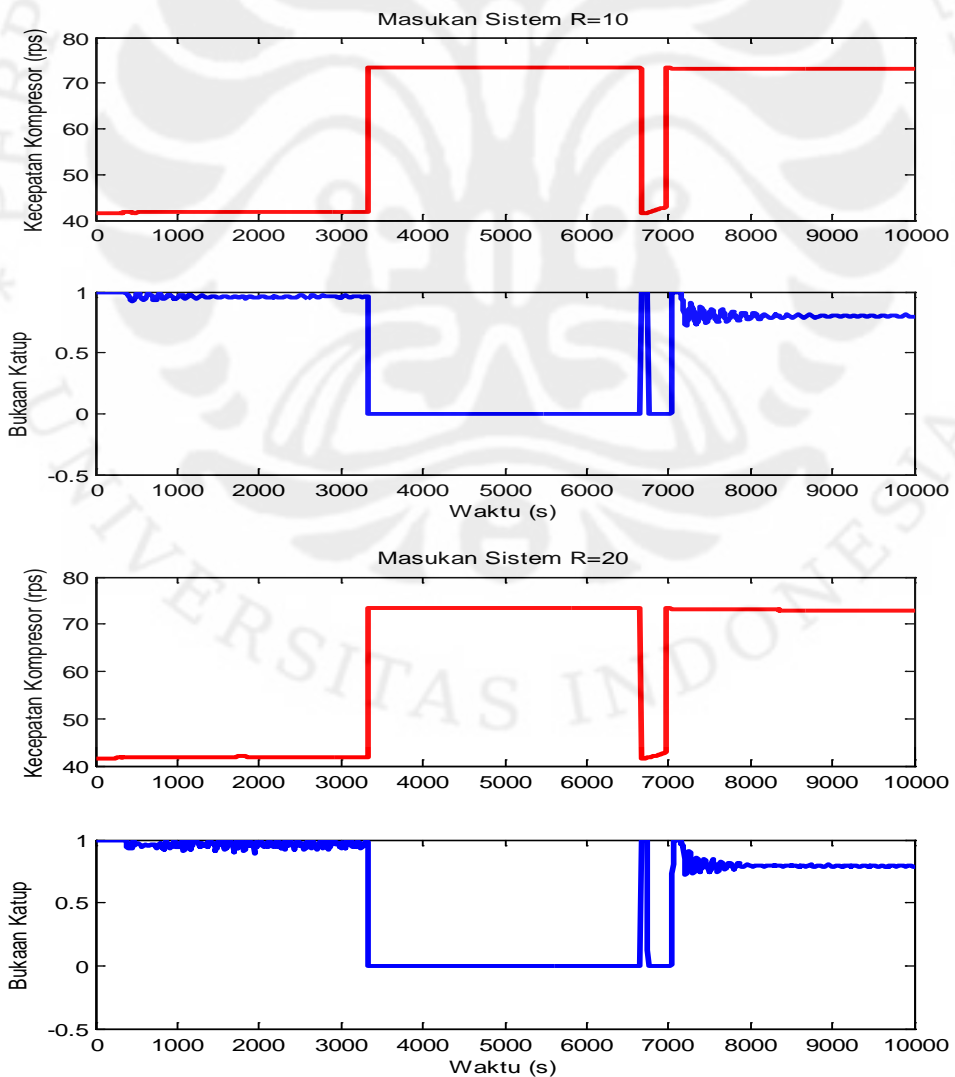
### 4.2.3 Hasil Simulasi dan Analisis Variasi Bobot Matriks R

Pada hasil keluaran pengendali pada nilai parameter pengendali  $H_p = H_u = 2$ ,  $Q = 67$  dan  $R = 1$  yang ditunjukkan oleh gambar 4.9, terlihat masih terdapat *overshoot* pada keluaran temperatur kabinet di sekitar  $t = 6700$ . Untuk itu perlu dicari nilai bobot  $R > 1$  untuk memperkecil perubahan sinyal dan meredam *overshoot* yang terjadi. Di bawah ini akan diperlihatkan hasil simulasi dengan variasi nilai bobot  $R$ .

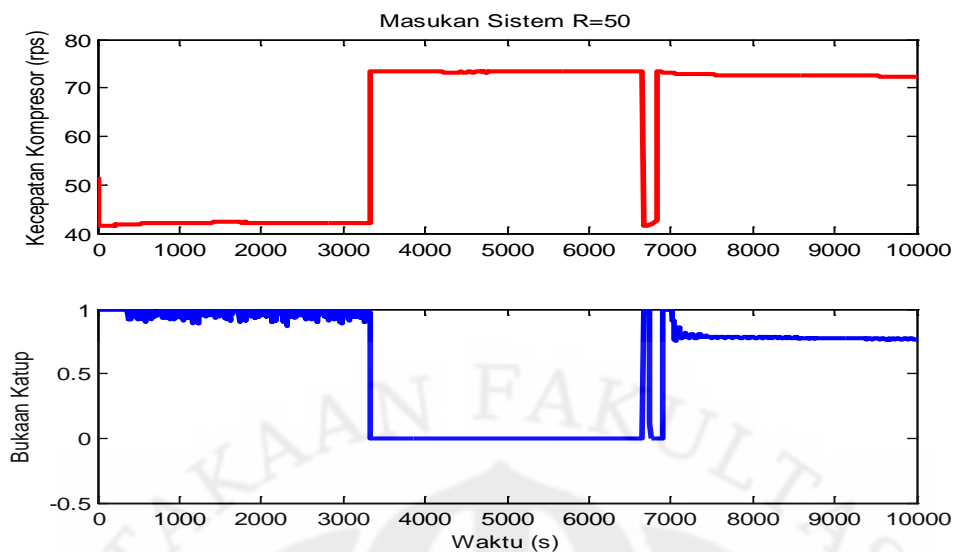




Gambar 4.12. Keluaran Sistem Hasil Pengendali MPC dengan *Constraint* dengan Variasi Nilai Bobot R







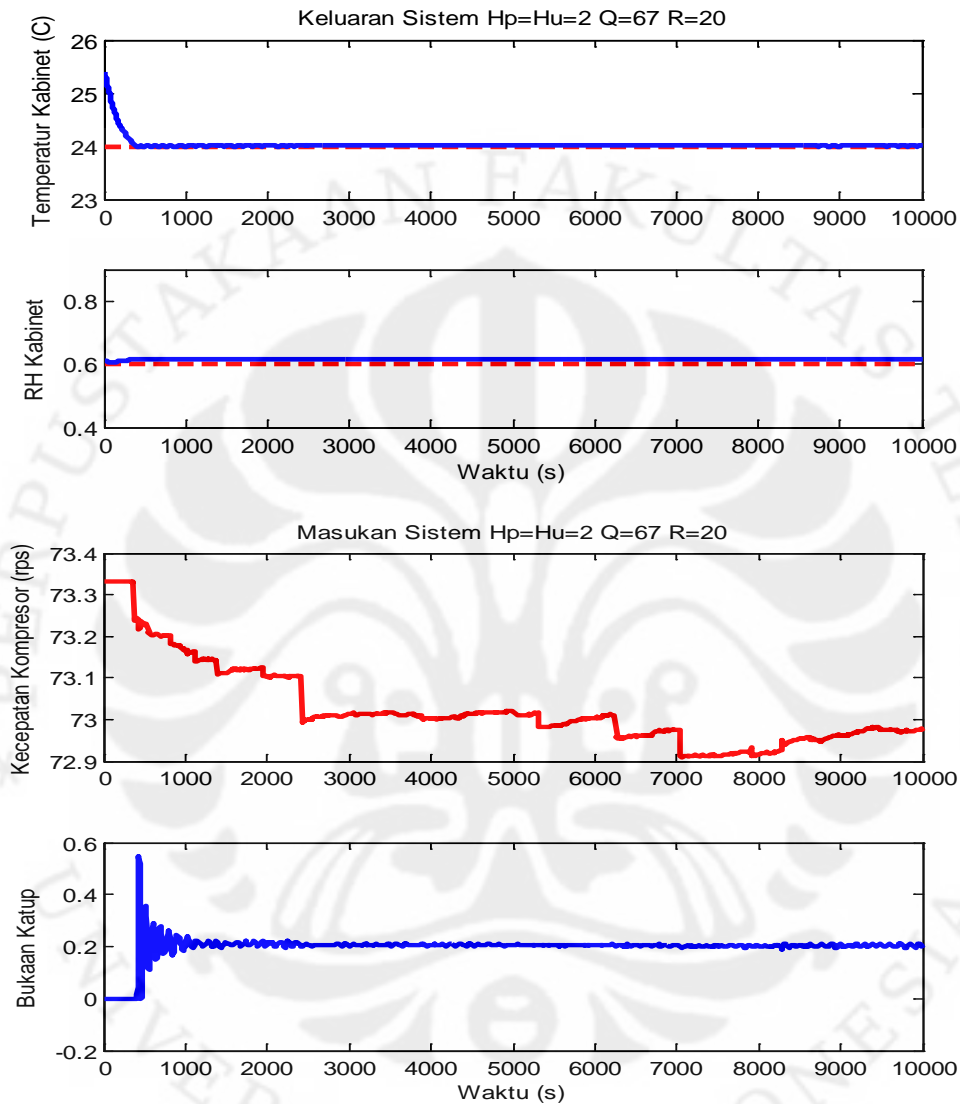
Gambar 4.13. Masukan Sistem Hasil Pengendali MPC dengan *Constraint* dengan Variasi Nilai Bobot  $R$

Hasil simulasi dengan variasi nilai bobot  $R$  pada gambar di atas menunjukkan semakin besar nilai bobot  $R$ , maka *overshoot*-nya semakin kecil. Akan tetapi, jika nilai bobot  $R$ -nya terlalu besar, maka keluaran yang dihasilkan jadi tidak mulus seperti yang terlihat pada keluaran RH kabinet dengan  $R = 50$  saat  $t = 1000$  sampai  $t = 3000$ . Oleh karena itu, nilai  $R$  yang dipilih untuk sistem ini adalah  $R = 20$ .

Dari variasi nilai bobot  $Q$  dan  $R$  yang telah dilakukan, terlihat perbedaan antara  $Q$  dan  $R$  walaupun keduanya sama-sama mempengaruhi nilai perubahan sinyal kendali. Matriks bobot  $Q$  memiliki pengaruh terhadap error antara sinyal referensi dengan keluaran dan respon keluaran sistem. Berdasarkan hasil simulasi variasi nilai bobot  $Q$  sebelumnya, dapat disimpulkan bahwa semakin besar nilai bobot  $Q$ , semakin kecil error trayektori dan semakin cepat respon keluaran dari sistem.

Matriks bobot  $R$  di sisi lain mempunyai pengaruh terhadap perubahan sinyal kendali dan dapat meredam *overshoot* yang terjadi. Semakin besar nilai bobot  $R$  yang dipakai, semakin kecil perubahan sinyal kendalinya dan semakin kecil pula *overshoot* yang terjadi. Pencarian nilai bobot  $Q$  dan  $R$  yang sesuai menghasilkan nilai parameter pengendali terbaik untuk sistem PAC ini sebesar  $H_p = H_u = 2$ ,  $Q = 67$  dan  $R = 20$ .

Pengendali MPC dengan *constraint* dengan nilai parameter pengendali terbaik yang telah ditentukan di atas jika diberi sinyal referensi bernilai konstan berupa  $r_1 = 24$  dan  $r_2 = 0,6$  akan menghasilkan:

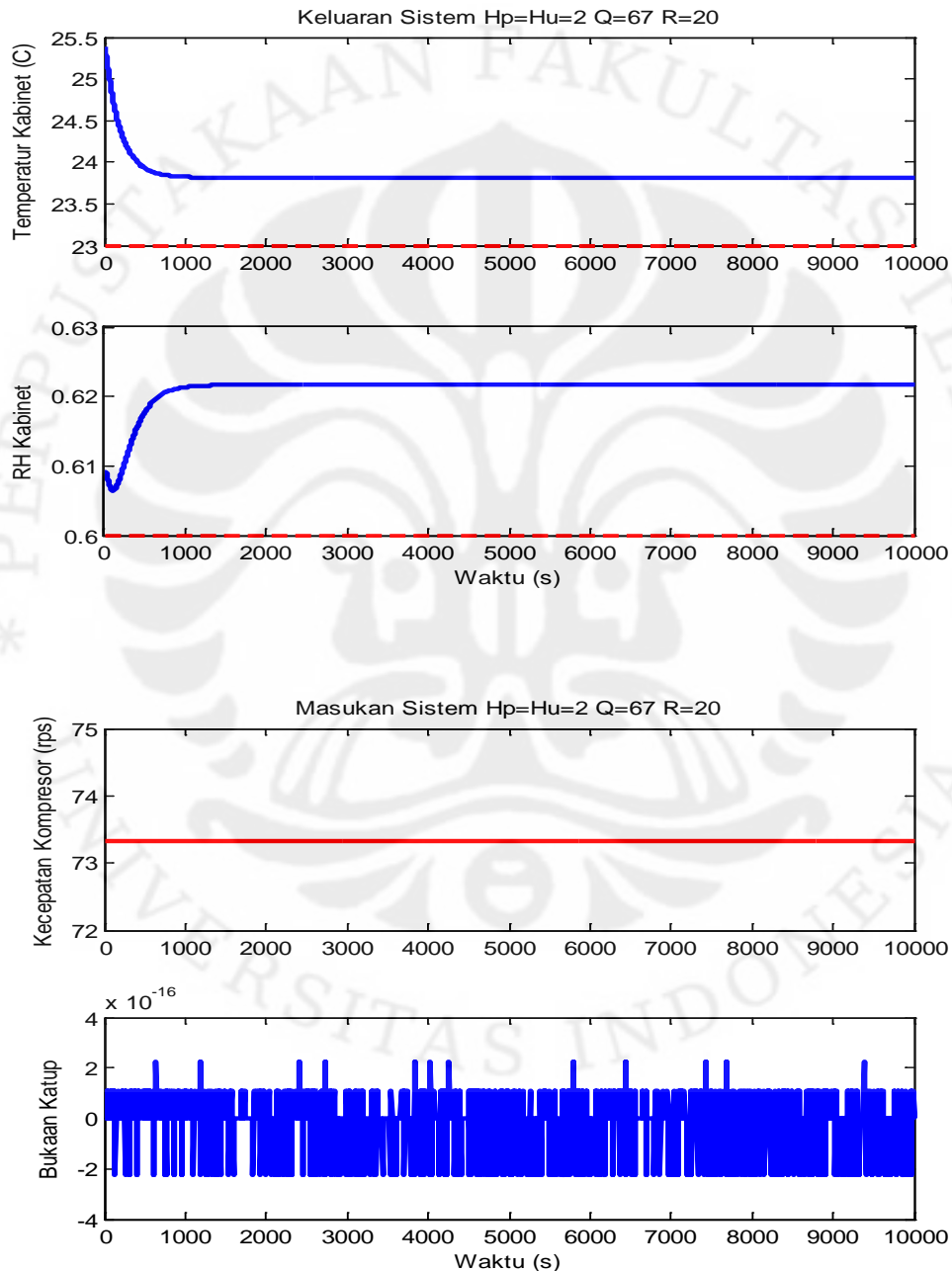


Gambar 4.14. Keluaran dan Masukan Sistem Hasil Pengendali MPC dengan *Constraint* dengan Nilai Referensi Konstan  $r_1 = 24$  dan  $r_2 = 0,6$

Gambar di atas memperlihatkan pada  $r_1 = 24$  dan  $r_2 = 0,6$ , hasil keluaran RH kabinet mengikuti sinyal referensi dengan sedikit error antara keluaran tunaknya dengan nilai referensinya. Hal ini menunjukkan saat temperatur kabinet bernilai  $24^{\circ}\text{C}$ , nilai RH kabinetnya bernilai sekitar 60% seperti terlihat pada gambar karena nilai RH kabinet yang dihasilkan mengacu pada temperatur kabinetnya sehingga hasil pengendali yang terlihat tidak jelek. Batasan  $\Delta u_1$  yang

ditentukan masih memiliki solusi yang *feasible*, sedangkan batasan  $\Delta u_2$  tidak memiliki solusi yang *feasible*.

Jika nilai referensi konstan yang dimasukkan adalah  $r_1 = 23$  dan  $r_2 = 0,6$ , maka hasil pengendali MPC dengan *constraint* dengan nilai parameter pengendali yang sama adalah sebagai berikut.



Gambar 4.15. Keluaran dan Masukan Sistem Hasil Pengendali MPC dengan *Constraint* dengan Nilai Referensi Konstan  $r_1 = 23$  dan  $r_2 = 0,6$

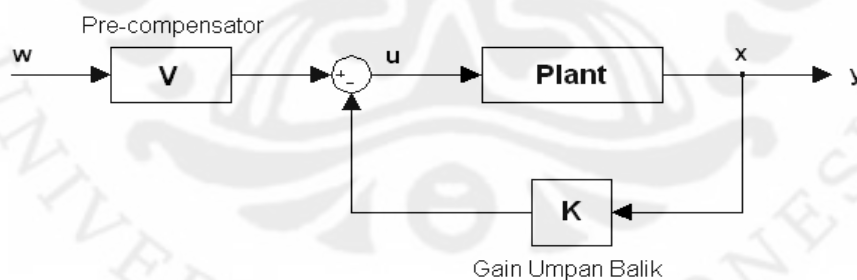
Seperti yang telah dijelaskan sebelumnya, pada kondisi model dengan beban 300 Watt yang dipakai di sini, nilai temperatur kabinet terendah yang bisa dicapai senilai  $23,8149^{\circ}\text{C}$  sehingga keluaran temperatur kabinet di sini tidak bisa mengikuti nilai referensi yang telah ditetapkan.

Nilai RH kabinet terlihat naik dan sudah tidak sesuai dengan nilai referensi RH yang ditentukan dibanding saat temperatur kabinet bernilai  $24^{\circ}\text{C}$  sesuai dengan hubungan antara temperatur dan RH, yaitu jika nilai temperatur naik, nilai RH akan turun dan sebaliknya. Sinyal masukan kecepatan kompresor sudah mencapai nilai maksimum dari batasan yang ditentukan sebesar 73,333 rps atau 4.400 rpm sehingga di sini sinyal kendali bukaan katup yang lebih aktif berubah-ubah untuk mengendalikan keluaran agar mengikuti nilai referensi.

### 4.3 Pengendali Penempatan Kutub dengan Umpan Balik Keadaan

#### 4.3.1 Penurunan Persamaan Pengendali

Untuk membandingkan hasil pengendali MPC yang telah dilakukan, maka dirancang pengendali penempatan kutub dengan umpan balik keadaan (*state-feedback pole placement*) menggunakan C-MEX *s-function*. Diagram blok pengendali ini ditunjukkan pada gambar di bawah ini.



Gambar 4.16. Diagram Blok Pengendali Penempatan Kutub dengan Umpan Balik Keadaan

Pada gambar di atas,  $\underline{K}$  merupakan matriks *gain* umpan balik yang dicari agar sinyal kendali  $\underline{u} = -\underline{K} \underline{x}$  dapat menempatkan kutub-kutub lingkaran tertutup di posisi yang diinginkan [8]. Untuk menggeser kutub-kutub sistem, sistem harus memiliki sifat *fully controllable* dan *fully observable*. Syarat lain untuk merancang pengendali ini adalah jumlah masukan sistem harus sama dengan jumlah keluaran sistem ( $m = p$ ).

Pada perancangan pengendali ini juga terdapat blok *pre-compensator* ( $V$ ) untuk mengatur agar sinyal keluaran sama dengan sinyal referensi. Berdasarkan diagram blok di atas, persamaan sinyal kendalinya:

$$\underline{u}(k) = \underline{V} \underline{w}(k) - \underline{K} \underline{x}(k) \quad (4.1)$$

dengan  $\underline{V} \in \mathbb{R}^{m \times p}$  dan  $\underline{K} \in \mathbb{R}^{m \times n}$ .

Persamaan pengendali penempatan kutub dengan umpan balik keadaan ini diturunkan pada kondisi tunak, yaitu saat nilai  $\underline{y}_s(k) = \underline{w}(k)$ . Pada kondisi tunak, nilai keadaan saat pencuplikan setelahnya sama dengan nilai keadaan sekarang, yaitu  $\underline{x}(k+1) = \underline{x}(k) = \underline{x}_s(k)$ . Dengan mensubstitusi persamaan (4.1) ke dalam persamaan model proses (2.1a):

$$\begin{aligned} \underline{x}(k+1) &= \underline{A} \underline{x}(k) + \underline{B} \underline{u}(k) \\ &= \underline{A} \underline{x}(k) + \underline{B} \underline{V} \underline{w}(k) - \underline{B} \underline{K} \underline{x}(k) \\ \underline{x}_s(k) &= (\underline{A} - \underline{B} \underline{K}) \underline{x}_s(k) + \underline{B} \underline{V} \underline{w}(k) \\ (\underline{I} - \underline{A} + \underline{B} \underline{K}) \underline{x}_s(k) &= \underline{B} \underline{V} \underline{w}(k) \\ \underline{x}_s(k) &= (\underline{I} - \underline{A} + \underline{B} \underline{K})^{-1} \underline{B} \underline{V} \underline{w}(k) \end{aligned} \quad (4.2)$$

dengan  $\underline{x}_s(k)$  merupakan nilai keadaan tunak.

Substitusi persamaan (4.1) dan (4.2) ke persamaan model proses (2.1b) dalam keadaan tunak menghasilkan:

$$\begin{aligned} \underline{y}_s(k) &= \underline{C} \underline{x}_s(k) + \underline{D} \underline{u}(k) \\ &= \underline{C} \underline{x}_s(k) + \underline{D} \underline{V} \underline{w}(k) - \underline{D} \underline{K} \underline{x}_s(k) \\ &= (\underline{C} - \underline{D} \underline{K}) \underline{x}_s(k) + \underline{D} \underline{V} \underline{w}(k) \\ &= (\underline{C} - \underline{D} \underline{K}) (\underline{I} - \underline{A} + \underline{B} \underline{K})^{-1} \underline{B} \underline{V} \underline{w}(k) + \underline{D} \underline{V} \underline{w}(k) \\ \underline{y}_s(k) &= \left( (\underline{C} - \underline{D} \underline{K}) (\underline{I} - \underline{A} + \underline{B} \underline{K})^{-1} \underline{B} + \underline{D} \right) \underline{V} \underline{w}(k) \end{aligned} \quad (4.3)$$

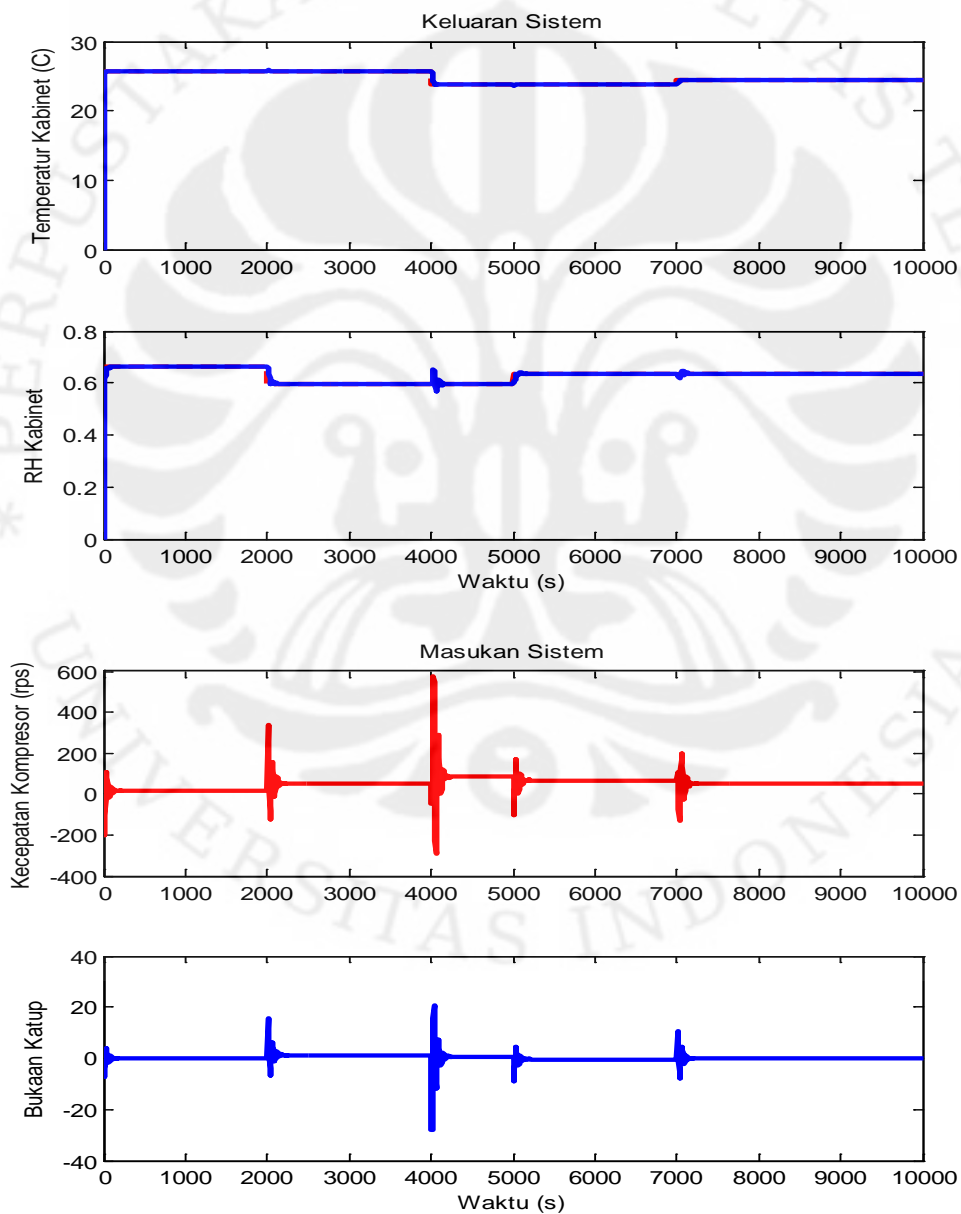
Dalam keadaan tunak, keluaran sistem diharapkan sama dengan referensi, yaitu  $\underline{y}_s(k) = \underline{w}(k)$  sehingga:

$$\underline{V} = \left( (\underline{C} - \underline{D}\underline{K})(\underline{I} - \underline{A} + \underline{B}\underline{K})^{-1}\underline{B} + \underline{D} \right)^{-1} \quad (4.4)$$

### 4.3.2 Hasil Simulasi dan Analisis Pengendali Penempatan Kutub

#### 4.3.2.1 Hasil Pengendali tanpa Blok Saturasi

Hasil simulasi pengendali penempatan kutub dengan umpan balik keadaan tanpa blok saturasi menggunakan C-MEX *s-function* dapat dilihat di bawah ini.



Gambar 4.17. Keluaran dan Masukan Sistem Hasil Pengendali Penempatan Kutub dengan Umpan Balik Keadaan tanpa Blok Saturasi

Hasil pengendali penempatan kutub dengan umpan balik keadaan memperlihatkan kedua sinyal keluaran, yaitu temperatur dan RH kabinet, mengikuti sinyal referensi dengan baik. Hal ini disebabkan persamaan pengendali yang dipakai diturunkan dengan kondisi  $\underline{y}_s(k) = \underline{w}(k)$ , sinyal keluaran tunak sama dengan sinyal referensi, sehingga keluaran sistem hasil pengendali ini tentu akan mengikuti sinyal referensinya.

Jika diperhatikan dengan lebih baik, hasil pengendalian ini tidak sebagus yang terlihat pada keluaran sistemnya. Terlihat dari sinyal kendalinya yang mengalami perubahan dan lonjakan drastis yang sangat besar setiap nilai sinyal referensinya berubah. Hal ini disebabkan matriks *gain* umpan balik yang didapat memiliki nilai yang sangat besar sehingga sangat sensitif terhadap gangguan atau perubahan pada sistem. Matriks *gain* umpan balik yang didapat dengan menggunakan fungsi *place* ( ) pada Matlab adalah sebagai berikut:

$$K = \begin{bmatrix} 426,2826 & 1,1156e3 & 200,0507 & -689,6633 & 1,2611e4 & 3,6083e4 & -4,0779e4 & 2,581e4 \\ -19,7167 & 50,8546 & 14,2769 & -33,1537 & 70,8006 & 668,4178 & -340,6807 & 405,6883 \end{bmatrix}$$

dengan kutub-kutub lup tertutup sistem:

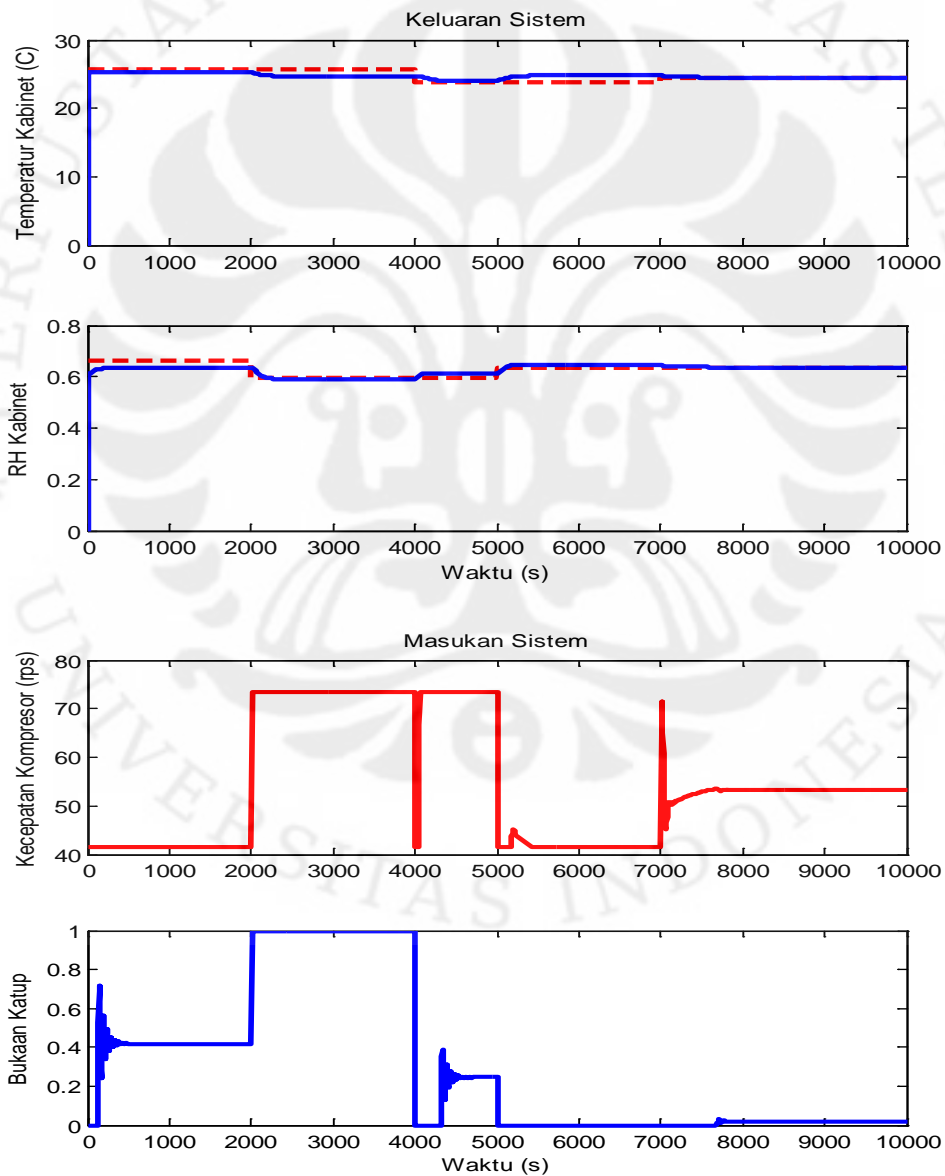
$$P = [0,8 \quad 0,8 \quad 0,1 \quad 0,4 \quad 0 \quad 0 \quad 0,6 \quad 0,6].$$

Pada dasarnya, pengendali penempatan kutub ini memang kurang cocok untuk sistem berorde tinggi. Nilai matriks *gain* umpan balik yang dihasilkan memang cenderung besar untuk sistem yang memiliki banyak keadaan seperti sistem tata udara presisi.

Meninjau gambar 4.17 kembali, jika diperhatikan secara seksama, terlihat adanya efek *coupling* pada temperatur kabinet ( $T_{cab}$ ) saat RH kabinet berubah pada  $t = 2000$  dan  $t = 5000$  dan efek *coupling* pada RH kabinet saat  $T_{cab}$  berubah pada  $t = 4000$  dan  $t = 7000$ . Berbeda dengan hasil MPC di mana RH kabinet selalu mengikuti perubahan temperatur kabinet, pada pengendali penempatan kutub ini, karena nilai keluaran mengikuti nilai referensi, saat salah satu sinyal referensi keluaran berubah, maka efek *coupling* akan terlihat pada keluaran lain yang bernilai tetap.

#### 4.3.2.2 Hasil Pengendali dengan Blok Saturasi

Hasil pengendali penempatan kutub tanpa blok saturasi di atas masih belum sesuai dengan kondisi fisik sebenarnya karena sinyal kendali berupa kecepatan kompresor dan bukaan katup yang dihasilkan melebihi batas kemampuan alat sebenarnya. Untuk melihat hasil yang sesuai dengan kondisi fisik alat sebenarnya, ditambahkan blok saturasi yang memberi batasan sebesar 41,667-73,333 rps (2.500-4.400 rpm) untuk kecepatan kompresor dan 0-100% untuk bukaan katup kondenser kedua. Berikut ini adalah hasilnya.



Gambar 4.18. Keluaran dan Masukan Sistem Hasil Pengendali Penempatan Kutub dengan Umpan Balik Keadaan dengan Blok Saturasi



Pada hasil pengendali penempatan kutub tanpa blok saturasi sebelumnya telah dijelaskan bahwa pengendali ini sangat sensitif terhadap gangguan karena memiliki nilai *gain* umpan balik yang sangat besar. Akibatnya, setiap terjadi perubahan nilai referensi, sinyal kendali yang dihasilkan juga menjadi sangat besar (terdapat lonjakan nilai sinyal kendali) seperti terlihat pada gambar 4.17.

Jika lonjakan sinyal kendali yang besar itu dipaksa dipotong sesuai dengan batasan fisik sebenarnya menggunakan blok saturasi, hasilnya terlihat pada gambar 4.18. Sinyal kendali pada gambar 4.18 ini nilainya sudah sesuai dengan kondisi fisik sebenarnya dan lonjakan nilai sinyal kendali yang besar sudah tidak terlihat lagi. Akan tetapi, keluaran sistem yang dihasilkan menjadi tidak terlalu sesuai dengan sinyal referensinya dan memiliki nilai error tunak yang lebih besar akibat pemotongan nilai sinyal kendali ini.

Efek *coupling* di sini juga lebih terlihat dan berpengaruh lebih besar dalam menambah error tunak. Terlihat saat  $T_{cab}$  yang seharusnya konstan sampai  $t = 4000$  nilainya berubah di  $t = 2000$  karena nilai referensi  $RH_{cab}$  berubah saat itu. Begitu pula nilai  $RH_{cab}$  yang seharusnya konstan dari  $t = 2000$  sampai  $t = 5000$ , nilainya berubah di  $t = 4000$  karena nilai referensi  $T_{cab}$  berubah pada waktu ini.

Dengan melihat hasil pengendali MPC dan penempatan kutub di atas, dapat dikatakan bahwa pengendali MPC lebih cocok untuk sistem berorde tinggi dibandingkan pengendali penempatan kutub. Oleh karena itu, sebenarnya untuk sistem tata udara presisi ini memang lebih tepat menggunakan MPC dibandingkan pengendali penempatan kutub. Hasil yang tidak bagus pada pengendali MPC menunjukkan model proses yang digunakan masih belum bagus dan belum merepresentasikan sistem dengan baik. Untuk menggunakan pengendali MPC ini, diperlukan model proses yang baik. Selama model proses yang digunakan dalam perancangan pengendali MPC sudah baik, maka hasil pengendalian MPC untuk sistem tata udara presisi ini akan jauh lebih baik dibandingkan hasil pengendali penempatan kutub.

## BAB 5 KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Kesimpulan yang dapat diambil dari skripsi ini, yaitu:

1. Hasil pengendali MPC pada sistem tata udara presisi masih belum bagus karena model proses yang digunakan belum merepresentasikan sistem dengan baik.
2. Keluaran RH kabinet ( $RH_{cab}$ ) tidak mengikuti sinyal referensinya, melainkan mengikuti temperatur kabinet ( $T_{cab}$ ), karena persamaan matematis model menunjukkan bahwa  $RH_{cab}$  sangat bergantung dan dipengaruhi oleh  $T_{cab}$ .
3. Hasil keluaran sistem belum mencapai spesifikasi yang diinginkan, yaitu 20-22°C untuk  $T_{cab}$  dan 45-55% untuk  $RH_{cab}$  karena kondisi alat sebenarnya saat model didapatkan juga belum bisa mencapai nilai yang sesuai dengan spesifikasi.
4. Pasangan masukan dan keluaran yang tepat untuk sistem tata udara presisi adalah kecepatan kompresor ( $u_1$ ) dengan  $T_{cab}$  ( $y_1$ ) dan bukaan katup ( $u_2$ ) dengan  $RH_{cab}$  ( $y_2$ ).
5. Pengaruh kedua sinyal masukan sistem hampir sama kuat pada setiap keluaran sistem.
6. Nilai parameter pengendali terbaik pada MPC tanpa *constraint* untuk sistem tata udara presisi adalah  $H_p = H_u = 2$ ,  $Q = 0,5$  dan  $R = 2$ .
7. Nilai parameter pengendali terbaik pada MPC dengan *constraint* untuk sistem tata udara presisi adalah  $H_p = H_u = 2$ ,  $Q = 67$  dan  $R = 20$ .
8. Pada pengendali MPC dengan *constraint*, batasan perubahan sinyal kendali (*slew rate*) tidak memiliki solusi yang *feasible* tetapi batasan amplitudo sinyal kendali memiliki solusi yang *feasible*.
9. Keluaran RH kabinet yang selalu berubah mengikuti perubahan temperatur kabinet pada pengendali MPC menyebabkan efek *coupling* menjadi tidak terlihat.

10. Pengendali MPC lebih *robust* dan tidak sensitif terhadap gangguan untuk sistem dengan orde tinggi seperti sistem tata udara presisi dibandingkan pengendali penempatan kutub.

## 5.2 Saran

Saran yang dapat diberikan berdasarkan hasil penelitian skripsi ini untuk memperbaiki hasil pengendalian sistem tata udara presisi, yaitu:

1. Mengganti persamaan matematis model untuk RH kabinet agar nilai RH kabinet tidak bergantung kepada temperatur kabinet, melainkan pada masukan sistem. Untuk itu, perlu dicari persamaan matematis RH yang mengandung nilai variabel masukan sistem (dalam hal ini berupa bukaan katup).
2. Tetap menggunakan persamaan matematis yang sama tetapi mengganti masukan sistem yang dipakai sehingga sesuai dengan persamaan matematis RH kabinet. Karena pada persamaan matematis RH kabinet tidak terdapat variabel bukaan katup, maka masukan bukaan katup ini perlu diganti dengan variabel yang ada pada persamaan matematis RH kabinet, misalnya kecepatan udara volumetris *fan*. Dengan begitu, nilai RH kabinet ini nantinya akan dipengaruhi langsung oleh masukan berupa *fan*.
3. Meng-*update* nilai parameter sistem hasil eksperimen berdasarkan kondisi alat terbaru. Hal ini perlu dilakukan jika kondisi alat sebenarnya sudah bisa mencapai nilai spesifikasi yang diinginkan.
4. Menggunakan metode pengendali yang tidak berbasiskan kepada model proses jika model proses yang didapatkan masih belum baik.

## DAFTAR REFERENSI

- [1] Subiantoro, Aries. (2007). *Diktat kuliah sistem kendali adaptif dan nonlinier*. Depok: Departemen Teknik Elektro, Universitas Indonesia.
- [2] Aryani, Dharma. (2009). *Perancangan pengendali Model Predictive Control (MPC) constrained pada sistem Proton Exchange Membrane Fuel Cell (PEM FC)*. Depok: Universitas Indonesia.
- [3] Tham, M.T. (1999). *Multivariable control: An introduction to decoupling control*. Newcastle: Dept. of Chemical and Process Engineering, University of Newcastle upon Tyne.
- [4] Sutarna, Nana. (2008, Desember). *Pemodelan sistem tata udara presisi*. Seminar dipresentasikan di Departemen Teknik Elektro, Fakultas Teknik Universitas Indonesia.
- [5] Deng, Shiming, & Qi, Qi. (2008). Multivariable control-oriented modeling of direct expansion (DX) air conditioning (A/C) system. *International Journal of Refrigeration*. September 15, 2009. <http://www.elsevier.com>
- [6] Sidi, Kuncoro. (n.d.). *Sistem tata udara untuk pengembangan AC presisi dua kondenser paralel dengan variasi kecepatan kompresor dan bukaan dua katup*. PPT dipresentasikan di Departemen Teknik Mesin, Fakultas Teknik Universitas Indonesia.
- [7] AJ Design. (n.d.). *Engine design calculator* [Perangkat lunak komputer]. Desember 30, 2009. [http://www.ajdesigner.com/phpengine/engine\\_equations\\_cubic\\_inch\\_displacement.php](http://www.ajdesigner.com/phpengine/engine_equations_cubic_inch_displacement.php)
- [8] The MathWorks, Inc. (2008). *MatLab* (Versi 7.6.0.324) [Perangkat lunak komputer]. U.S.A: The MathWorks, Inc. (MatLab Help)