



UNIVERSITAS INDONESIA

**Perancangan Pengendali MPC SISO (*Single Input Single Output*)
Adaptif pada Sistem Tata Udara Presisi**

SKRIPSI

Abdon Jonas Lumban Gaol

0806330610

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JUNI 2012**



UNIVERSITAS INDONESIA

Perancangan Pengendali MPC SISO (*Single Input Single Output*)
Adaptif pada Sistem Tata Udara Presisi

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana

Abdon Jonas Lumban Gaol

0806330610

FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JUNI 2012

HALAMAN PERNYATAAN ORISINALITAS

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

Nama : Abdon Jonas Lumban Gaol

NPM : 0806330610

Tanda Tangan :



Tanggal : 25 Juni 2012

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Abdon Jonas Lumban Gaol
NPM : 0806330610
Program Studi : Teknik Elektro
Judul Skripsi : Perancangan Pengendali MPC SISO (*Single Input
Single Output*) Adaptif 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.SEE. ()
Penguji : Dr. Ir. Feri Yusivar M.Eng. ()
Penguji : Dr. Ir. Ridwan Gunawan M.T. ()

Ditetapkan di : Depok

Tanggal : 26 Juni 2012

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan yang telah memberikan kekuatan dan semangat pribadi kepada penulis dalam pengerjaan laporan skripsi ini hingga selesai tepat waktu. Laporan skripsi ini ditulis sebagai salah satu persyaratan kelulusan untuk memperoleh gelar Sarjana Teknik dari Departemen Teknik Elektro Universitas Indonesia.

Sebagai orang yang telah mendukung, membantu dan membimbing penulis dalam pengerjaan laporan skripsi ini, maka penulis ingin berterima kasih kepada:

- (1) Ir. Aries Subiantoro M.SEE sebagai dosen pembimbing skripsi penulis yang dengan sabar telah mengajarkan dan mengarahkan penulis banyak hal.
- (2) Teman-teman satu bimbingan yang telah menyemangati dan membantu penulis hingga laporan skripsi ini selesai.
- (3) Orang tua dan keluarga saya yang telah mendukung saya dalam doa dan finansial.
- (4) Teman-teman Teknik Elektro yang sama-sama berjuang menyelesaikan skripsi masing-masing, yang juga saling memberikan semangat.

Akhir kata, penulis berharap agar laporan skripsi ini bisa bermanfaat bagi para pembaca, dan dapat berperan di dalam pengembangan ilmu pengetahuan dan teknologi di Indonesia, khususnya di Universitas Indonesia.

Depok, 13 Juni 2012

Penulis

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

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

Nama : Abdon Jonas Lumban Gaol
NPM : 0806330610
Program Studi : Teknik Elektro
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis Karya : Skripsi

demi 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 MPC SISO (*Single Input Single Output*) Adaptif pada
Sistem Tata Udara Presisi

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan 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 : 26 Juni 2012

Yang menyatakan



(Abdon Jonas Lumban Gaol)

ABSTRAK

Nama : Abdon Jonas Lumban Gaol
Program Studi : Teknik Elektro
Judul : Perancangan MPC SISO (*single input single output*) Adaptif pada Sistem Tata Udara Presisi

Penelitian ini berfokus pada perancangan pengendali *Adaptive Model Predictive Control* (MPC) atau MPC Adaptif pada sistem tata udara presisi atau *Precision Air Conditioning* (PAC) untuk mengendalikan temperatur dan kelembapan relatif, keluaran PAC tersebut, agar sesuai dengan yang diinginkan. PAC yang dikendalikan ini akan digunakan sebagai pendingin kabinet-kabinet yang ada di ruang pusat data (*data center*) sebuah perusahaan telekomunikasi. Untuk itu, diharapkan keluaran dari PAC yang sudah dikendalikan berada pada kisaran 20°C hingga 22°C untuk temperaturnya dan 40% - 55% untuk kelembapan relatif (RH)-nya. Namun, penelitian ini masih pada tahap perancangan pengendali untuk temperatur saja.

Pengendali MPC Adaptif yang dirancang adalah pengendali MPC dengan *constraint* yang memanfaatkan identifikasi rekursif. Dengan demikian, masukan MPC berupa parameter ruang keadaan akan di-*update* secara rekursif melalui tahap identifikasi sehingga mampu menghasilkan nilai keluaran sistem yang lebih baik untuk setiap *update*-annya. Identifikasi yang dilakukan adalah menggunakan algoritma PO-MOESP rekursif. Adapun pengendali MPC Adaptif yang dirancang masih menggunakan model SISO (*single input single output*), dimana hanya masukan kompresor dan keluaran temperatur yang diukur, sementara masukan kipas dijaga konstan dan keluaran kelembapan relatifnya diabaikan. Sebelum MPC Adaptif diterapkan, terlebih dahulu dirancang pengendali MPC dengan *constraint* untuk identifikasi *offline*, dan apabila hasil keluarannya sudah sesuai dengan yang diinginkan, maka pengendali MPC tersebut akan digabungkan dengan identifikasi rekursif.

Pengendali MPC Adaptif yang dirancang masih menggunakan *m-file*. Akan tetapi, program *m-file* ini bisa diaplikasikan untuk sistem MIMO (multi input multi output). Pengendali yang didesain diharapkan mampu menjamin kestabilan sistem dengan error yang cenderung lebih kecil dibandingkan pengendali MPC biasa, serta mampu menjaga kestabilan temperatur apabila direalisasikan pada PAC.

Kata kunci : MPC, *constraint*, Identifikasi sistem, PO-MOESP, Rekursif, sistem tata udara presisi

ABSTRACT

Name : Abdon Jonas Lumban Gaol
Major : Electrical Engineering
Title : The Design of Adaptive SISO (Single Input Single Output) MPC for Precision Air Conditioning (PAC)

This research focuses on Adaptive MPC Design for Precision Air Conditioning (PAC) in order to control the temperature and the relative humidity produced by PAC on desired interval. This controlled PAC would be utilized as air conditioner for cabinets in data center for telecommunication. Therefore, the controlled PAC is expected to produce temperature within 20°C - 22°C and relative humidity within 40% - 55%. Nevertheless, this research is still focusing on controlling the temperature, whereas on controlling relative humidity would be in further research.

The proposed adaptive MPC is designed by integrating recursive identification on MPC with constraint. Thus, Inputs (state variables) for MPC is updated recursively by online identification which is expected to produce better output for each update. Identification itself implements recursive PO-MOESP algorithm for SISO (single input single output) by measuring the compressor as the input and temperature as the output, whereas the fan (input) is made constant and the relative humidity (output) is neglected. MPC with constraint for offline identification is firstly designed before implementing it with online identification. Online identification would be integrated to modeled MPC only if the modeled MPC for offline identification produces desired output.

The proposed adaptive SISO MPC is designed using *m-file* program in Matlab. Nevertheless, the *m-file* program is also applicable for MIMO (multi input multi output) system. The designed controller is expected to minimize the error and to guarantee the stability of the system. Besides, the designed adaptive SISO MPC is also expected to be able to control the temperature when it's embedded to PAC system.

Keywords : MPC, constraint, System identification, PO-MOESP, Recursive, PAC

DAFTAR ISI

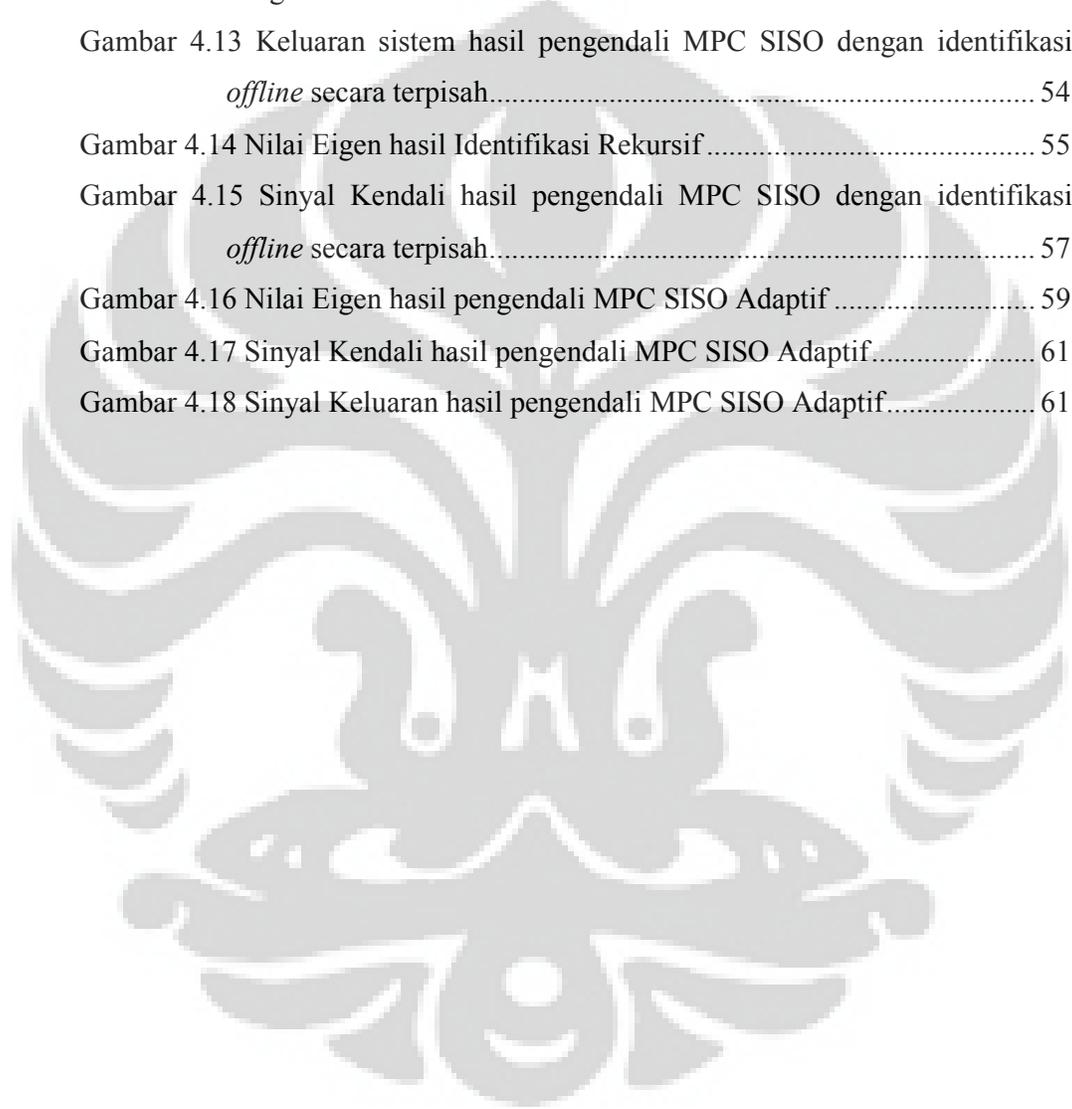
HALAMAN JUDUL	i
HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN	iii
KATA PENGANTAR	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Tujuan Penelitian	2
1.3 Pembatasan Masalah	3
1.4 Metodologi Penelitian	3
1.5 Sistematika Penulisan	4
BAB 2 DASARTEORI	6
2.1 Struktur Sistem MPC Adaptif	6
2.1 Konsep Dasar <i>Model Predictive Control</i> (MPC).....	6
2.2.1 Model Proses	8
2.2.2 Perhitungan Prediksi MPC	9
2.2.3 Fungsi Kriteria.....	12
2.2.4 Minimasi Fungsi Kriteria MPC tanpa <i>Constraint</i>	13
2.2.5 Pembentukan <i>Constraints</i> pada MPC.....	14
2.3 Identifikasi Rekursif.....	16
2.4 Sistem Tata Udara Presisi (PCA).....	16
2.4.1 Prinsip Kerja Sistem Tata Udara Presisi	17
2.4.2 Asumsi-asumsi pada Sistem Tata Udara Presisi	19
BAB 3 MERANCANG PENGENDALI.....	21
3.1 Model PAC dengan dentifikasi <i>Offline</i>	22
3.2 Pengendali MPC SISO dengan <i>constraint</i>	23
3.2.1 Algoritma MPC SISO dengan <i>Constraint</i>	24

3.2.2 Menghitung Fungsi Kriteria	28
3.3 Pengendali MPC SISO Adaptif	29
BAB 4 HASIL SIMULASI dan ANALISIS	32
4.1 Simulasi dan Analisis Hasil Pengendali SISO dengan <i>Constraint</i>	33
4.1.1 Hasil Simulasi dan Analisis untuk Variasi Nilai <i>Prediction Horizon</i> (H_p)	33
4.1.2 Hasil Simulasi dan Analisis untuk Variasi nilai <i>Control Horizon</i> (H_u)	37
4.1.3 Hasil Simulasi dan Analisis untuk Variasi Faktor Bobot Perubahan Sinyal Kendali R	42
4.1.4 Hasil Simulasi dan Analisis untuk Variasi Faktor Bobot Kesalahan Q	46
4.1.5 Pengaruh Variasi Batas Sinyal Kendali pada Hasil Pengendali MPC	49
4.2 Simulasi Pengendali MPC SISO dengan <i>Constraint</i> dengan Identifikasi Rekursif Secara Terpisah.....	53
4.3 Simulasi Hasil Pengendali MPC SISO Adaptif	57
BAB 5 KESIMPULAN dan SARAN	63
5.1 Kesimpulan	63
5.2 Saran	63
DAFTAR REFERENSI	64

DAFTAR GAMBAR

Gambar 2.1 Struktur pengendali MPC Adaptif.....	6
Gambar 2.2 Struktur dasar MPC	7
Gambar 2.3 Konsep <i>Receding Horizon</i>	8
Gambar 2.4 Bagian sistem tata udara presisi (PAC).....	17
Gambar 2.5 Diagram Pipa sistem tata udara presisi.....	17
Gambar 2.6 Diagram P-h siklus refrigasi	18
Gambar 3.1 Model PAC untuk Diidentifikasi.....	22
Gambar 3.2 Diagram alir algoritma pengendali MPC dengan <i>constraint</i>	25
Gambar 3.3 Diagram Pengendali MPC Adaptif.....	30
Gambar 3.4 Diagram alir algoritma PO-MOESP rekursif	31
Gambar 3.5 Diagram alir algoritma MPC SISO Adaptif dengan <i>constraint</i>	32
Gambar 4.1 Keluaran sistem hasil pengendali MPC SISO dengan <i>constraint</i> dengan variasi nilai H_p	35
Gambar 4.2 Sinyal Kendali hasil pengendali MPC dengan <i>constraint</i> dengan variasi nilai H_p	36
Gambar 4.3 Keluaran sistem hasil pengendali MPC SISO dengan <i>constraint</i> dengan variasi nilai H_u	38
Gambar 4.4 Sinyal Kendali hasil pengendali MPC dengan <i>constraint</i> dengan variasi nilai H_u	39
Gambar 4.5 Keluaran sistem hasil pengendali MPC SISO dengan <i>constraint</i> dengan variasi nilai H_u (bagian 2).....	41
Gambar 4.6 Keluaran sistem hasil pengendali MPC SISO dengan <i>constraint</i> dengan variasi nilai matrix R	43
Gambar 4.7 Sinyal Kendali hasil pengendali MPC SISO dengan <i>constraint</i> dengan variasi nilai matrix R	44
Gambar 4.8 Keluaran Sistem hasil pengendali MPC SISO dengan <i>constraint</i> dengan variasi nilai matrix R (bagian 2).....	46
Gambar 4.9 Keluaran sistem hasil pengendali MPC SISO dengan <i>constraint</i> dengan variasi nilai matrix Q	48

Gambar 4.10 Sinyal Kendali hasil pengendali MPC SISO dengan <i>constraint</i> dengan variasi nilai matrix Q	49
Gambar 4.11 Keluaran sistem hasil pengendali MPC SISO dengan <i>constraint</i> dengan variasi besar <i>constraint</i>	50
Gambar 4.12 Sinyal Kendali hasil pengendali MPC SISO dengan <i>constraint</i> dengan variasi besar <i>constraint</i>	52
Gambar 4.13 Keluaran sistem hasil pengendali MPC SISO dengan identifikasi <i>offline</i> secara terpisah.....	54
Gambar 4.14 Nilai Eigen hasil Identifikasi Rekursif	55
Gambar 4.15 Sinyal Kendali hasil pengendali MPC SISO dengan identifikasi <i>offline</i> secara terpisah.....	57
Gambar 4.16 Nilai Eigen hasil pengendali MPC SISO Adaptif	59
Gambar 4.17 Sinyal Kendali hasil pengendali MPC SISO Adaptif.....	61
Gambar 4.18 Sinyal Keluaran hasil pengendali MPC SISO Adaptif.....	61



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Untuk penyimpanan data dalam sistem telekomunikasi dibutuhkan sebuah ruang pusat data yang disebut sebagai ruang *data center*. Ruang pusat data ini haruslah bebas dari gangguan agar semua data telah tersimpan didalamnya tidak mengalami gangguan dan kerusakan. Ruang pusat data terdiri dari kabinet-kabinet yang harus dijaga kestabilan suhu dan kelembapan udaranya pada interval tertentu agar semua peralatan elektronik yang digunakan di dalamnya dapat berfungsi dengan baik. Salah satu alat yang digunakan untuk mempertahankan suhu dan kelembapan udara ruang pusat data adalah Sistem Tata Udara Presisi atau *Precision Air Conditioning (PAC)*.

Sistem tata udara presisi (PAC) pada awalnya dikembangkan untuk diaplikasikan pada bisnis komputer yang membutuhkan ruangan bersuhu kamar dalam proses pabrikasinya, dimana PAC ini dikembangkan untuk dapat mempertahankan suhu dan kelembapan ruangan sesuai dengan yang diinginkan. Sistem PAC ini kemudian diaplikasikan pada ruang *data center* (pusat data), ruang *Server*, *Switch center*, Peralatan medis, dan masih banyak lagi.

Penggunaan PAC sangat membutuhkan energi yang sangat besar. Pemakaian energi yang sangat besar itu membutuhkan biaya hingga mencapai triliunan rupiah hanya untuk dalam satu ruang pusat data saja. Sehingga, dikembangkan metode-metode pengendali PAC yang memungkinkan untuk menghemat pemakaian energi seefisien mungkin karena setiap penghematan energi yang dibutuhkan, misalnya hingga 10%, akan mampu menghemat pengeluaran hingga miliaran rupiah.

Metode-metode yang telah digunakan dalam mengendalikan temperatur dan kelembapan udara pada sistem tata udara presisi adalah metode Proportional-Integral-Derivate (Noor Hayate Abdul Hamid et al., 2009); Model Predictive Control (Matt Wallace et al., 2011); Generalized Predictive Control (Pedro Rodríguez and Didier Dumur, 2005); Decentralized Nonlinear Predictive Control (Zhang Huaguang and Lilong Cai, 2002); dan metode pengendali lainnya.

Pada awalnya, metode PI/PID digunakan untuk mengontrol temperatur dan kelembapan ruang Data Center, namun metode ini tidak bisa digunakan untuk model nonlinear, sedangkan sistem yang digunakan pada industri proses umumnya adalah model nonlinier. Selain itu, Metode PI/PID juga tidak bisa digunakan untuk model yang menggunakan *constraint* dan optimasi.

Setelah penelitian yang berkelanjutan, dikembangkanlah metode *Model Predictive Control* (MPC) yang bisa diterapkan untuk sistem yang menggunakan *constraint* dan teknik optimasi. Tidak hanya itu, sistem nonlinier atau sistem linier juga bisa diaplikasikan pada pengendali MPC. Kelebihan lain dari MPC adalah kemampuannya untuk mengatasi masalah MIMO (*multi input multi output*) yang umumnya dibutuhkan dalam industri proses. Akan tetapi, MPC masih rendah dalam performansi dan dalam meminimalisir gangguan yang ada. Meskipun MPC cenderung stabil untuk input dengan *constraint* hingga horizon tak terhingga, MPC masih menimbulkan beberapa masalah dalam prakteknya.

Oleh karena itu, Metode MPC kemudian dikembangkan dengan menggabungkannya dengan sebuah pengendali, yaitu pengendali adaptif (*adaptive control*). Hasil penggabungan kedua metode ini disebut sebagai pengendali MPC Adaptif. Pada penelitian ini, MPC berfungsi untuk mengendalikan sistem dan *adaptive control* berfungsi sebagai identifikasi rekursif untuk meng-*update* inputan MPC secara terus-menerus. Pada awalnya MPC Adaptif yang dikembangkan tidaklah menggunakan *constraints*, namun sekarang sudah mempertimbangkan *constraints* karena menghasilkan kestabilan yang lebih tinggi. Dalam laporan skripsi ini, MPC yang digunakan adalah SISO (*single input single output*) dengan model identifikasi yang bersifat rekursif.

1.2 Tujuan Penelitian

Penelitian ini dilakukan pada sistem tata udara presisi dengan menggunakan sistem pengendali MPC Adaptif dengan *constraints*. Penelitian ini merupakan pengembangan dari penelitian sebelumnya yang telah menggunakan pengendali MPC namun belum memanfaatkan model identifikasi rekursif.

Identifikasi rekursif yang telah dilakukan dengan menggunakan metode PO-MOESP menghasilkan nilai variable ruang keadaan (A, B, C, D) secara terus-

menerus untuk digunakan sebagai masukan dari sistem pengendali MPC yang telah dirancang. Hasil perancangan pengendali MPC Adaptif ini akan digunakan untuk mengendalikan masukan sistem tata udara presisi, yaitu masukan kompresor yang berubah-ubah dan masukan kipas yang konstan pada tegangan 24 Volt, sehingga keluaran yang dihasilkan sesuai dengan yang diinginkan, yaitu berupa temperatur. Sistem pengendali MPC ini mampu memperkecil *error* yang terjadi dan membuat sistem lebih stabil sehingga keluaran (temperatur) berada pada interval yang diinginkan.

1.3 Pembatasan Masalah

Masalah pada skripsi ini dibatasi pada perancangan MPC sistem SISO (*single input single output*) dengan *constraint* untuk sistem tata udara presisi dengan memanfaatkan masukan ruang keadaan dari identifikasi secara *offline* dan rekursif (*online*). Akan tetapi, tahap dan proses identifikasi sudah diluar batas masalah skripsi ini. Batasan atau *constraint* diberikan untuk perubahan sinyal kendali (*slew rate*) dan besar nilai sinyal kendali yang berupa kecepatan kompresor.

Model yang digunakan adalah model linier diskrit hasil identifikasi menggunakan algoritma PO-MOESP yang diperoleh dari anggota tim peneliti yang lain. Hasil pengendalian menggunakan MPC akan ditampilkan dalam bentuk grafik sinyal keluaran sistem dan sinyal kendali.

1.4 Metodologi Penelitian

Metodologi penilitan dan penulisan yang digunakan dalam menyusun skripsi ini adalah:

1. Studi Literatur

Penulis membaca buku, skripsi, jurnal, dan bahan-bahan lain yang berkaitan dengan PCA dan MPC Adaptif.

2. Konsultasi dengan dosen pembimbing

Pertemuan antara dosen pembimbing dengan penulis dan tim peneliti lain dilakukan secara rutin sekitar sekali hingga dua kali seminggu untuk membantu penulis memahami konsep, melakukan penelitian dan penulisan

laporan skripsi hingga selesai. Segala permasalahan dan kesulitan yang dihadapi selama penelitian dikonsultasikan dengan pembimbing untuk menyelesaikan permasalahan dan menjawab pertanyaan yang ada.

3. Diskusi dengan orang yang terlebih dahulu melakukan penelitian dan dengan teman-teman satu penelitian.

Penulis mendiskusikan konsep penelitian dengan tim peneliti lainnya dan menanyakan hal-hal yang belum dimengerti kepada peneliti sebelumnya. Anggot dalam satu tim penelitit juga saling membantu dalam memecahkan masalah yang ditemukan pada saat melakukan pemrograman.

1.5 Sistematikan Penelitian

Laporan skripsi ini ditulis dengan sistematika penulisan sebagai berikut:

1. Bab 1 : Pendahuluan

Pada bab ini, penulis menjelaskan Latar Belakang, Tujuan Penelitian, Pembatasan Masalah, Metodologi Penelitian, dan Sistematika Penulisan, yang diuraikan per sub bab nya.

2. Bab 2 : Dasar Teori

Pada bagian ini, penulis menjelaskan konsep Sistem Tata Udara Presisi konsep MPC (*Model Predictive Control*), dan konsep MPC Adaptif. Penulis juga menjelaskan secara singkat mengenai konsep identifikasi secara offline dan rekursif yang akan dimanfaatkan untuk perancangan MPC Adaptif.

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

Pada bab ini, penulis menjelaskan tentang perancangan pengendali MPC Adaptif yang merupakan gabungan antara pengendali MPC SISO dengan *constraint* dan identifikasi rekursif.

4. Bab 4 : Hasil Simulasi dan Analsis

Pad bagian ini, penulis menganalisis hasil simulasi dari pengendalian sistem tata udara presisi dengan menggunakan sistem pengendali MPC dengan *constraint* sebelum dan sesudah menggunakan identifikasi rekursif. Hasil simulasi ini kemudian dianalisis berdasarkan konsep MPC Adaptif yang telah dibaca oleh penulis.

5. Bab 5 : Kesimpulan dan Saran

Pada bab ini, penulis menguraikan kesimpulan yang dapat diambil dari penelitian yang telah dilakukan untuk menyelesaikan laporan skripsi ini. Saran juga diberikan untuk memperbaiki hasil penelitian selanjutnya.

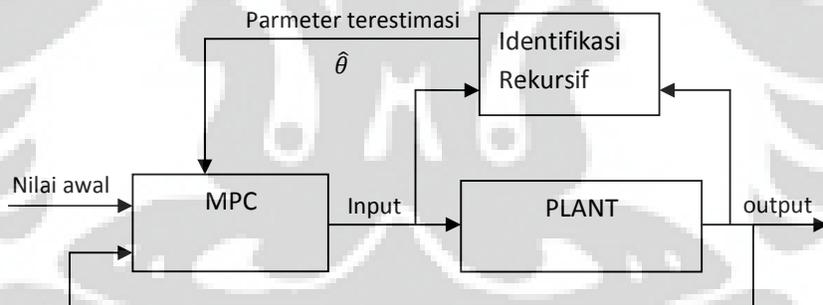


BAB 2

DASAR TEORI

2.1 Struktur Sistem MPC Adaptif

Sistem pengendali MPC Adaptif merupakan sistem pengendali MPC yang menggunakan model identifikasi rekursif. Sistem pengendali MPC membutuhkan nilai parameter ruang keadaan A, B, C dan D sebagai masukan sistem yang diperoleh dari hasil identifikasi sistem yang telah dilakukan oleh peneliti lain. Identifikasi yang telah dilakukan untuk menghasilkan nilai parameter ruang keadaan tersebut ada identifikasi dengan metode PO-MOESP untuk sistem SISO (*single input single output*). Identifikasi sistem ini didesain secara rekursif sehingga menghasilkan nilai A, B, C dan D yang *ter-update* secara *online* untuk digunakan secara rekursif pada pengendali MPC. Dengan demikian, keluaran yang dihasilkan pengendali MPC pun menjadi lebih stabil dengan *error* yang relatif kecil. Struktur pengendali MPC Adaptif ditunjukkan pada gambar 2.1 dibawah ini.



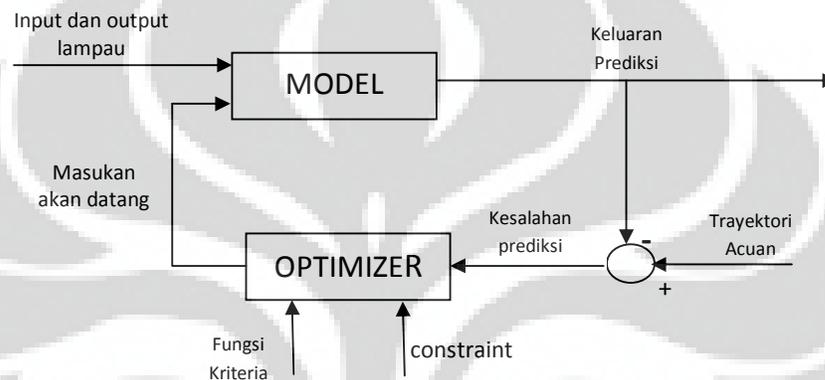
Gambar 2.1 Struktur pengendali MPC Adaptif

2.2 Konsep Dasar *Model Predictive Control* (MPC)

Sistem pengendali prediktif atau *Model Predictive Control* (MPC) merupakan metode pengendali prediktif yang berbasiskan pada 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 v dipindahkan menuju waktu pencuplikan berikutnya (pada waktu $k+1$) dengan melibatkan pemakaian sinyal kendali pertama, yaitu (k), untuk mengendalikan proses, dan prosedur di atas diulang dengan menggunakan informasi terakhir.
- Struktur dasar MPC diperlihatkan pada gambar 2.2 dibawah ini.



Gambar 2,2. Stuktur dasar MPC

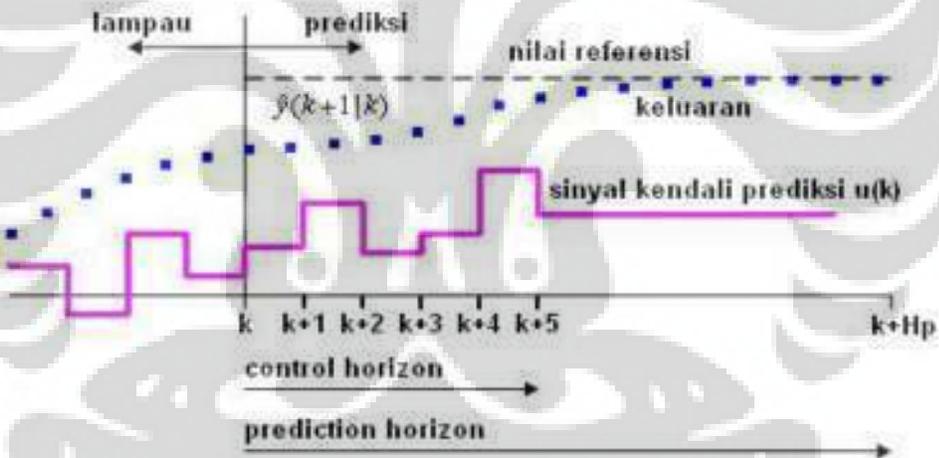
Gambar 2.2 menggambarkan metodologi pengendali yang termasuk dalam kategori MPC, yang dapat dijelaskan sebagai berikut[3]:

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 terprediksi 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 $i = 0 \dots H_u - 1$, dimana 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 sedeka tmungkin terhadap trayektori acuan $r(k + i)$. Fungsi kriteria tersebut umumnya berupa suatu fungsi kuadratik dari kesalahan antara sinyal keluaran terprediksi dengan trayektori acuan. Dalam banyak kasus, tujuan pengendalian seperti pemakaian energi yang minimum disertakan di dalam fungsi kriteria.

Solusi eksplisit dapat diperoleh jika fungsi kriteria adalah kuadratik, model linier, dan tidak ada *constraints*. Jika tidak, optimasi iteratif harus digunakan untuk memecahkannya. Langkah pertama dan kedua ini dapat diilustrasikan pada gambar 3.5 di halaman berikutnya.

3. Sinyal kendali $\hat{u}(k|k)$ dikirim ke proses, sedangkan sinyal kendali untuk prediksi berikutnya dibuang, karena pada pencuplikan berikutnya $y = (y(k + 1))$ sudah diketahui nilainya. Sehingga, langkah pertama diulang dengan nilai keluaran proses yang baru dan semua prosedur perhitungan yang diperlukan bisa diperbaiki. Sinyal kendali yang baru $\hat{u} = (k + 1|k + 1)$ mempunyai nilai yang berbeda dari $\hat{u} = (k + 1|k)$ dan dihitung dengan menggunakan konsep *receding horizon*.

Konsep dari *receding horizon* ini dapat digambarkan pada gambar 2.1. Konsep ini merupakan konsep dasar dari metode MPC yang akan digunakan.



Gambar 2.3 Konsep *Receding Horizon*^[3]

2.2.1 Model Proses

Solusi eksplisit dari metode MPC dapat diperoleh dengan menggunakan asumsi model proses yang bersifat linier. Maka persamaan model proses yang digunakan pada metode ini merupakan persamaan model ruang keadaan berikut.

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

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

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 merupakan 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.

2.2.2 Perhitungan Prediksi MPC

Persamaan model proses (2.1) menyatakan bahwa nilai keluaran sistem dapat diperoleh dengan menghitung nilai keadaan (\underline{x}) dan nilai masukan sistem (\underline{u}) terlebih dahulu. Nilai keluaran prediksi sistem juga dapat diperoleh dengan cara yang sama karena estimasi keluaran prediksi bergantung pada nilai keadaan prediksi dan nilai masukan prediksi. Oleh karena itu, nilai estimasi keadaan prediksi $\hat{\underline{x}}(k+i|k)$ selama rentang *prediction horizon* (H_p) dapat diperoleh terlebih dahulu dengan melakukan perhitungan sebagai berikut:

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

Besaran $\hat{\underline{u}}(k|k)$ menyatakan nilai estimasi sinyal kendali prediksi saat k untuk pencuplikan k . Pada saat estimasi prediksi, nilai $\underline{u}(k)$ yang merupakan sinyal kendali saat k belum diketahui. Untuk mencari nilai keluaran prediksi $\hat{\underline{y}}(k+i|k)$, dibutuhkan nilai masukan prediksi $\hat{\underline{u}}(k+i|k)$ dan nilai sinyal kendali yang dihitung selama rentang *control horizon* adalah perubahan sinyal kendalnya ($\Delta \hat{\underline{u}}(k+i|k)$). Akan tetapi, nilai sinyal kendali prediksi dapat diperoleh dengan menggunakan persamaan:

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

Sehingga estimasi sinyal kendali prediksi $\hat{\underline{u}}(k+i|k)$ selama rentang *control horizon* (H_u) dengan $i = 0, \dots, H_u - 1$ adalah:

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

Nilai sinyal kendali yang telah didapat pada persamaan (2.4) disubstitusi ke dalam persamaan (2.2) sehingga estimasi keadaan prediksi menjadi:

$$\begin{aligned}
\hat{\underline{x}}(k+1|k) &= \underline{A}\underline{x}(k) + \underline{B} \underbrace{(\Delta\hat{\underline{u}}(k|k) + \underline{u}(k-1))}_{\hat{\underline{u}}(k|k)} \\
\hat{\underline{x}}(k+2|k) &= \underline{A}^2\underline{x}(k) + \underline{A}\underline{B} \underbrace{(\Delta\hat{\underline{u}}(k|k) + \underline{u}(k-1))}_{\hat{\underline{u}}(k|k)} \\
&\quad + \underline{B} \underbrace{(\Delta\hat{\underline{u}}(k+1|k) + \Delta\hat{\underline{u}}(k|k) + \underline{u}(k-1))}_{\hat{\underline{u}}(k+1|k)} \\
&= \underline{A}^2\underline{x}(k) + (\underline{A} + \underline{I})\underline{B}\Delta\hat{\underline{u}}(k|k) + \underline{B}\Delta\hat{\underline{u}}(k+1|k) \\
&\quad + (\underline{A} + \underline{I})\underline{B}\underline{u}(k-1) \\
&\quad \vdots \\
\hat{\underline{x}}(k+Hu|k) &= \underline{A}^{Hu}\underline{x}(k) + (\underline{A}^{Hu-1} + \dots + \underline{A} + \underline{I})\underline{B}\Delta\hat{\underline{u}}(k|k) + \dots \\
&\quad + \underline{B}\Delta\hat{\underline{u}}(k+Hu-1|k) + (\underline{A}^{Hu-1} + \dots + \underline{A} + \underline{I})\underline{B}\underline{u}(k-1)
\end{aligned} \tag{2.5}$$

Jika $\underline{D} = 0$ dan $Hp \neq Hu$, maka estimasi keadaan prediksi di persamaan (2.5) dilanjutkan di mana $\Delta\hat{\underline{u}}(k+i|k) = 0$ untuk $i \geq Hu$ sesuai dengan prinsip *receding horizon* pada gambar 2.3 sehingga $\hat{\underline{u}}(k+i|k) = \hat{\underline{u}}(k+Hu-i|k)$.

Maka persamaan (2.5) dapat dilanjutkan menjadi:

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

Persamaan (2.5) dan (2.6) dapat diubah dalam bentuk matriks seperti berikut ini:

$$\begin{aligned}
 \begin{bmatrix} \underline{\hat{x}}(k+1|k) \\ \vdots \\ \underline{\hat{x}}(k+Hu|k) \\ \underline{\hat{x}}(k+Hu+1|k) \\ \vdots \\ \underline{\hat{x}}(k+Hp|k) \end{bmatrix} &= \underbrace{\begin{bmatrix} \underline{A} \\ \vdots \\ \underline{A}^{Hu} \\ \underline{A}^{Hu+1} \\ \vdots \\ \underline{A}^{Hp} \end{bmatrix}}_{\Psi \in \mathbb{R}^{(n \times Hp) \times n}} \underline{x}(k) + \underbrace{\begin{bmatrix} \underline{B} \\ \vdots \\ \sum_{i=0}^{Hu-1} \underline{A}^i \underline{B} \\ \sum_{i=0}^{Hu} \underline{A}^i \underline{B} \\ \vdots \\ \sum_{i=0}^{Hp-1} \underline{A}^i \underline{B} \end{bmatrix}}_{\Gamma \in \mathbb{R}^{(n \times Hp) \times m}} \underline{u}(k-1) \\
 &\quad \text{lampau} \\
 + \underbrace{\begin{bmatrix} \underline{B} & \dots & \underline{0}_{n \times m} \\ \underline{A} \underline{B} + \underline{B} & \dots & \underline{0}_{n \times m} \\ \vdots & \vdots & \vdots \\ \sum_{i=0}^{Hu-1} \underline{A}^i \underline{B} & \dots & \underline{B} \\ \sum_{i=0}^{Hu} \underline{A}^i \underline{B} & \dots & \underline{A} \underline{B} + \underline{B} \\ \vdots & \vdots & \vdots \\ \sum_{i=0}^{Hp-1} \underline{A}^i \underline{B} & \dots & \sum_{i=0}^{Hp-Hu} \underline{A}^i \underline{B} \end{bmatrix}}_{\Theta \in \mathbb{R}^{(n \times Hp) \times (m \times Hu)}} \begin{bmatrix} \Delta \underline{\hat{u}}(k|k) \\ \vdots \\ \Delta \underline{\hat{u}}(k+Hu-1|k) \end{bmatrix} \\
 &\quad \text{prediksi}
 \end{aligned} \tag{2.7}$$

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

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

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 kemudian diubah ke dalam bentuk matriks menjadi:

$$\begin{bmatrix} \underline{\hat{y}}(k+1|k) \\ \vdots \\ \underline{\hat{y}}(k+Hp|k) \end{bmatrix} = \underbrace{\begin{bmatrix} \underline{C} & \underline{0}_{p \times n} & \dots & \underline{0}_{p \times n} \\ \underline{0}_{p \times n} & \underline{C} & \dots & \underline{0}_{p \times n} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{0}_{p \times n} & \underline{0}_{p \times n} & \dots & \underline{C} \end{bmatrix}}_{\underline{C}_y \in \mathbb{R}^{(p \times Hp) \times (n \times Hp)}} \begin{bmatrix} \underline{\hat{x}}(k+1|k) \\ \vdots \\ \underline{\hat{x}}(k+Hp|k) \end{bmatrix}$$

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

Jika persamaan (2.7) disubstitusi ke dalam persamaan (2.9) akan menghasilkan:

$$\begin{bmatrix} \hat{y}(k+1|k) \\ \hat{y}(k+2|k) \\ \vdots \\ \hat{y}(k+Hp|k) \end{bmatrix} = \underbrace{\begin{bmatrix} \underline{C} \underline{A} \\ \underline{C} \underline{A}^2 \\ \vdots \\ \underline{C} \underline{A}^{Hp} \end{bmatrix}}_{C_y \Psi \in \mathbb{R}^{(p \times Hp) \times n}} \underline{x}(k) + \underbrace{\begin{bmatrix} \underline{C} \underline{B} \\ \underline{C} (\underline{A} \underline{B} + \underline{B}) \\ \vdots \\ \underline{C} (\underline{A}^{Hp-1} \underline{B} + \underline{A}^{Hp-2} \underline{B} + \cdots + \underline{A} \underline{B} + \underline{B}) \end{bmatrix}}_{C_y \Gamma \in \mathbb{R}^{(p \times Hp) \times m}} \underline{u}(k-1)$$

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

2.2.3 Fungsi Kriteria

Mendapatkan nilai perubahan sinyal kendali prediksi merupakan hal yang paling penting dalam metode MPC, yaitu dengan cara meminimasi fungsi kriteria yang ditentukan. Fungsi kriteria ini merupakan fungsi kuadrat yang merupakan selisih antara keluaran prediksi dengan trayektori acuan. Persamaan fungsi kriteria yang dipakai pada metode MPC adalah:

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

dimana:

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

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

$\Delta \hat{u}(k+i|k)$: 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 $\underline{R} \in \mathbb{R}^{(m \times Hu) \times (m \times Hu)}$

H_p : *prediction horizon* (horizon prediksi)

H_u : *control horizon* (horizon kendali)

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.

2.2.4 Minimasi Fungsi Kriteria MPC tanpa *Constraint*

Fungsi kriteria pada persamaan (2.11) dapat disederhanakan menjadi seperti berikut ini:

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

dimana:

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

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

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

Untuk memperoleh nilai perubahan sinyal kendali, fungsi kriteria diatas perlu diminimasi dengan mengubah kembali persamaan (2.12) ke dalam bentuk sederhana yang mengandung $\Delta \underline{U}(k)$. Untuk itu dibutuhkan matriks kesalahan penjejukan $\underline{E}(k)$, yang merupakan selisih antara nilai trayektori acuan yang akan datang dengan respon bebas sistem. Respon bebas merupakan nilai keluaran di sepanjang *horizon* saat tidak ada perubahan sinyal kendali ($\Delta \underline{U}(k) = 0$).

Persamaan matematis dari $\underline{E}(k) \in \mathbb{R}^{(p \times Hp)}$ dapat dituliskan sebagai:

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

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

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

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 terlihat di bawah ini:

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

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

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:

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

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

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

2.2.5 Pembentukan *Constraint* pada MPC

Pada dasarnya, fungsi kriteria yang digunakan pada MPC dengan *constraints* sama dengan fungsi kriteria yang digunakan pada MPC tanpa *constraints* [2]. Hanya saja pada MPC dengan *constraints*, terdapat kriteria tambahan berupa batasan-batasan (*constraints*) yang diberikan dalam bentuk pertidaksamaan dan/atau persamaan. Batasan-batasan ini bisa diberlakukan pada perubahan sinyal (*slew rate*), nilai sinyal kendali dan pada keluaran sistem.

Sebelum meminimasi fungsi kriteria, batasan-batasan yang akan dipakai pada perancangan MPC perlu ditentukan terlebih dahulu. Sebagai contoh adalah

constraints untuk perubahan sinyal kendali, dimana nilai sinyal kendali 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 menjadi:

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

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

Mengingat persamaan (3.3), bentuk pertidaksamaan matriks di atas dapat diubah ke dalam bentuk persamaan matriks sebagai berikut:

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

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

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.

2.3 Identifikasi Rekursif

Dalam perancangan MPC Adaptif ini digunakan identifikasi rekursif dengan menggunakan algoritma PO-MOESP. PO MOESP adalah MOESP biasa dengan konstruksi *instrumental variables* dari pengukuran masukan lampau dan keluaran lampau. Dimensi sistem LTI yang akan diidentifikasi diasumsikan memiliki representasi sebagai berikut:

$$x_{k+1} = \mathbf{A}x_k + \mathbf{B}u_k \quad (2.24)$$

$$y_k = \mathbf{C}x_k + \mathbf{D}u_k \quad (2.25)$$

di mana $x_k \in^n$, $u_k \in^r$. Dengan asumsi matriks \mathbf{A} harus stabil dan pasangan matriks (\mathbf{A}, \mathbf{C}) harus dapat diobservasi.

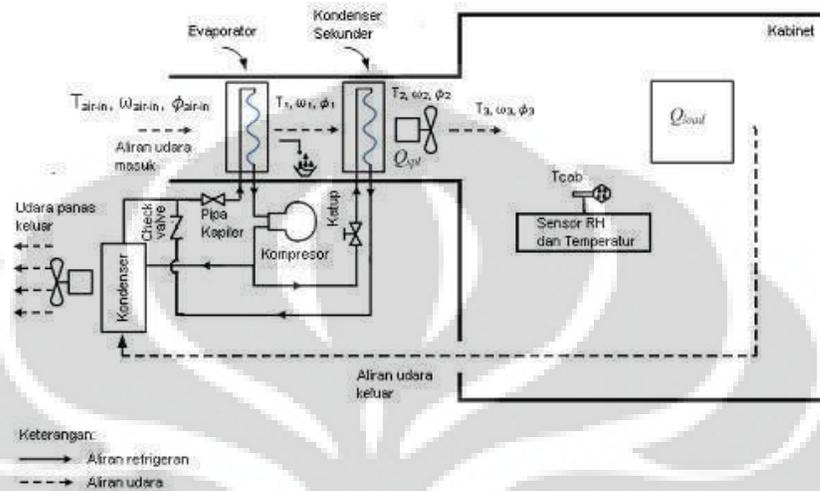
Algoritma PO MOESP memperkirakan matriks koefisien $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ secara rekursif dari data I/O yang didapat.

2.4 Sistem Tata Udara Presisi (PCA)

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

PAC banyak digunakan di ruang pusat data yang membutuhkan pengendalian suhu dan kelembapan agar peralatan dalam ruang atau kabinet

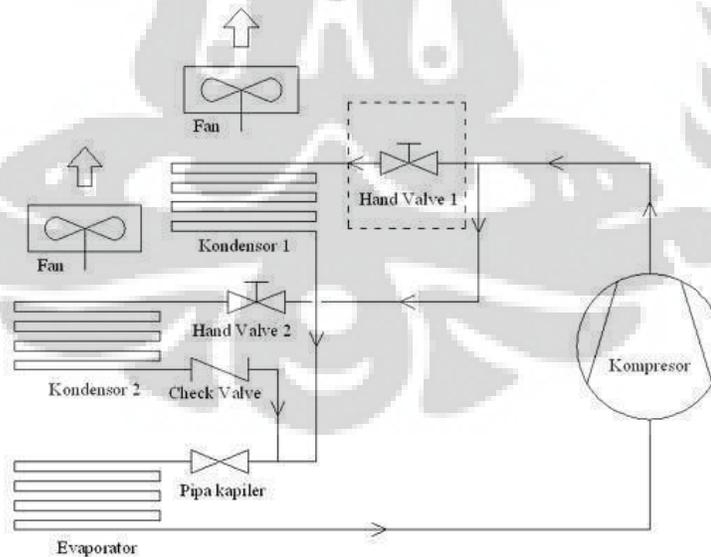
tersebut tidak cepat rusak. Untuk ruang pusat data (data center), temperature ideal di didalam kabinet berkisar antara 20-22° C, sedangkan kelembapan relatif (RH) yang ideal di dalam kabinet berkisar antara 45-55 %. Sistem PAC dapat diilustrikan pada gambar 2.5 berikut.



Gambar 2.4. Bagian sistem tata udara presisi (PAC)^[3]

2.4.1 Prinsip Kerja Sistem Udara Presisi

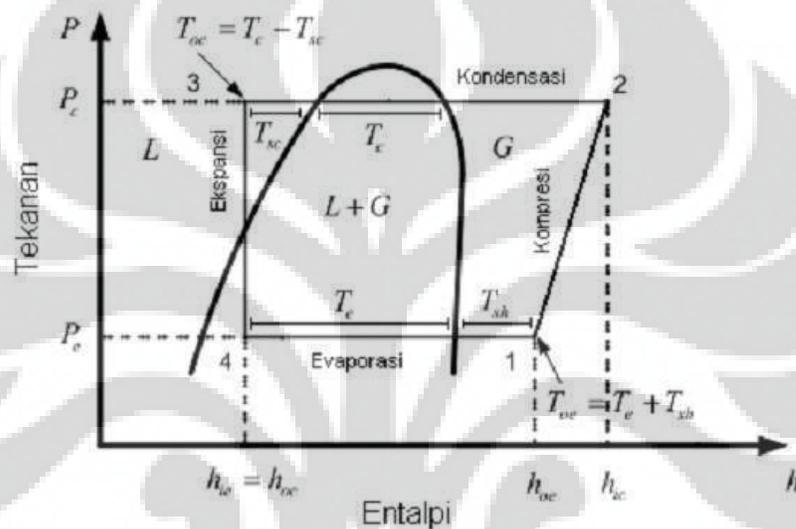
Diagram PAC dapat ditunjukkan pada gambar 2.6 berikut.



Gambar 2.5. Diagram pipa sistem tata udara presisi^[3]

Pada sistem udara presisi terdapat dua aliran, yaitu aliran udara dan aliran refrigeran, dimana fluida yang bekerja sebagai refrigeran pada PAC adalah R134a. Pertama-tama, aliran udara ditarik masuk oleh *fan* dari ruang pusat data ke dalam PAC melalui evaporator kemudian masuk ke kondensator kedua dan masuk kedalam kabinet. Dari kabinet udara kembali masuk ke dalam PAC untuk kemudian dibuang keluar oleh kondensator pertama (gambar 2.4).

Siklus aliran refrigeran pada PAC dapat dijelaskan melalui gambar 2.6 berikut.



Gambar 2.6. Diagram P-h siklus refrigerasi^[3]

Tahap yang dilewati terdiri dari 4 tahap:

1) Tahap Kompresi (1-2)

Pada tahap kompresi, kompresor mulai bekerja menaikkan tekanan refrigeran dan mengalirkannya ke kondensator pertama saat PAC dinyalakan. Refrigeran yang keluar dari kompresor adalah berupa gas dengan suhu dan tekanan yang tinggi.

2) Tahap Kondensasi (2-3)

Tahap kondensasi terjadi didalam kondensator pertama. Pada kondensator pertama temperature refrigeran lebih tinggi dibandingkan temperature udara di luar, sehingga panas dari dalam refrigeran berpindah ke luar refrigeran yang menyebabkan udara yang dibuang oleh udara kondensator pertama menjadi lebih

panas. Pembuangan panas dari refrigeran ke lingkungan luar menimbulkan terjadi proses kondensasi sehingga terjadi perubahan fase dari gas ke cair pada refrigeran.

3) Tahap Ekspansi (3-4)

Tahap kompensasi terjadi pada pipa kapiler. Refrigeran R134a cair yang keluar dari kondensor pertama mengalir ke dalam pipa kapiler. Disini tekanan refrigerant menurun drastis karena ada efek penghambatan oleh alat ekspansi.

4) Tahap Evaporasi(4-1)

Tahap evaporasi terjadi di evaporator, dimana refrigerant yang bertekanan dan bersuhu rendah masuk kedalam evaporator. Di evaporator temperatur refrigeran lebih rendah dibandingkan dengan temperatur udara dalam PAC yang mengakibatkan terjadinya penyerapan panas oleh refrigeran. Hal ini berdampak pada udara yang keluar dari evaporator sehingga menjadi lebih rendah dibandingkan dengan temperature udara ruang pusat data yang masuk ke dalam evaporator yang kelembapan relatif (RH) nya cukup tinggi ($\phi_1 = 95\%$ s/d 100%). Penyerapan panas ini menimbulkan terjadinya proses evaporasi yaitu refrigeran berubah fase dari cair ke gas (gambar 2.6).

Refrigeran dalam bentuk gas yang keluar dari evaporator kemudian masuk kembali ke kompresor untuk di kompresi dan seterusnya. Siklus inilah yang dinamakan sebagai siklus refrigerasi (*vapor-compression cycle*). PAC yang digunakan disini memiliki kondensor tambahan, sehingga refrigerant yang keluar dari kompresor tidak hanya mengalir ke kondensor pertama, tetapi juga ke kondensor kedua apabila kutub kondensor kedua dibuka (gambar 2.5).

2.4.2 Asumsi-asumsi pada Sistem Tata Udara Presisi

Asumsi-asumsi yang berlaku pada sistem tata udara presisi yang harus diperhatikan adalah:

- Campuran udara terjadi di dalam evaporator, kondenser dan lingkungan,
- Temperature 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,

- Temperature kondensasi di kondensor dianggap konstan,
- Sisi udara di kondensor hanya meliputi daerah kering (*dry region*) saja,
- Aliran refrigeran yang mengalir ke kondensor 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 volumetric (*air volumetric flow*) dalam sistem dianggap konstan, dan
- Rugi-rugi panas pada daerah aliran udara diabaikan.

Asumsi-asumsi diatas dibutuhkan dalam membahas persamaan matematis sistem tata udara presisi^[3]. Keluaran yang akan dikendalikan pada sistem PAC adalah temperatur kabinet (T_{cab}) dan kelembapan relatif kabinet (ϕ_{cab}), sehingga persamaan matematis yang digunakan adalah persamaan-persamaan dari model kompresor, model evaporator, model kondensor kedua, model udara masuk ke dalam kabinet (*supply air*) dan model kabinet.

Persamaan matematis model kondenser pertama tidak dibutuhkan karena informasi keluaran sistem berupa temperatur dan kelembapan relatif (RH) udara dalam cabinet (T_{cab} , ϕ_{cab}) sudah diperoleh tanpa memerlukan informasi mengenai udara yang dibuang keluar oleh kondensor pertama.

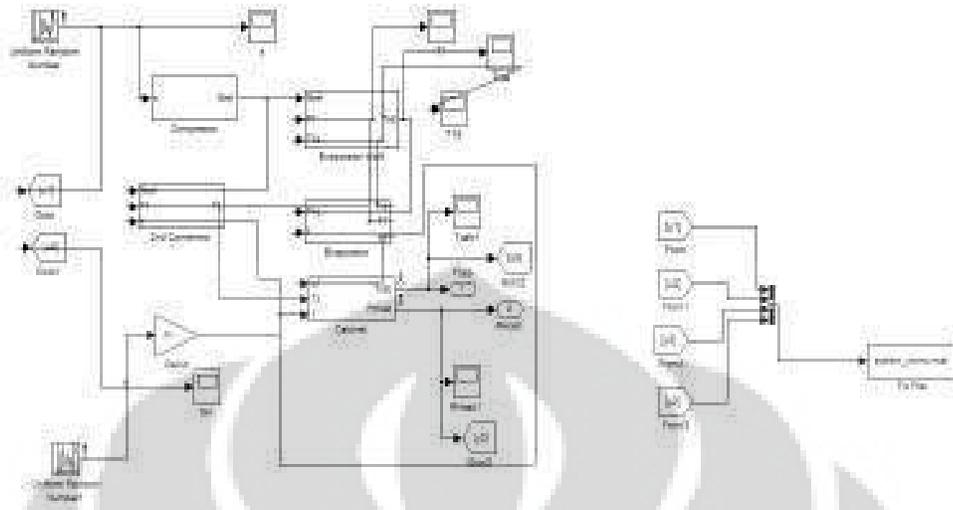
BAB 3

MERANCANG PENGENDALI

Model proses dari sistem tata udara presisi (PAC) yang dimanfaatkan pada perancangan pengendali MPC Adaptif ini merupakan model keadaan linear yang diperoleh dari hasil identifikasi sistem menggunakan metode PO-MOESP yang dilakukan oleh tim peneliti lain. Seperti yang telah dijelaskan sebelumnya, MPC Adaptif merupakan penggabungan dari model pengendali MPC dan *Recursive Identification* (Identifikasi bersifat rekursif). Identifikasi rekursif ini menghasilkan nilai parameter ruang keadaan yang diperoleh berdasarkan masukan data diskrit, sehingga keluaran sistem nya di-*update* secara rekursif. Dengan demikian, nilai matrix parameter sistemnya akan terus berubah sebanyak jumlah data N. Setiap perubahan nilai matrix parameter yang terjadi akan langsung diolah dengan pengendali MPC, sehingga keluaran MPC pun selalu di-*update* secara rekursif berdasarkan model sistem yang diperoleh dari hasil identifikasi rekursif tersebut.

Pengendali MPC Adaptif yang dirancang adalah pengendali untuk sistem SISO (*single input single output*). Oleh karena itu, model PAC yang diidentifikasi juga harus terlebih dahulu dimodifikasi menjadi sistem SISO, yaitu dengan satu masukan dan satu keluaran. Modifikasi model PAC yang dilakukan adalah dengan men-*set* masukan kipas menjadi konstan pada 23 - 24 Volt, sedangkan masukan kompresornya tetap seperti semula. Untuk keluaran, hanya keluaran temperatur yang diukur sedangkan kelembapan relatifnya diabaikan.

Model PAC yang digunakan untuk identifikasi ditunjukkan pada gambar 3.1 dibawah ini.



Gambar 3.1. Model PAC untuk diidentifikasi

Pada gambar diatas terlihat bahwa model PAC tersebut adalah sistem MIMO (*multi input multi output*) dengan masukan kipas dan kompresor. Untuk memperoleh model PAC SISO, maka inpuatan kipasnya dibuat konstan dan kompresor nya tetap pada keadaan semula. Untuk keluaran, hanya temperatur yang diukur.

3.1 Model PAC dengan Identifikasi *Offline*

Sebelum update nilai matrix parameter sistemnya dilakukan secara rekursif (*online*), proses identifikasi terlebih dahulu dilakukan secara *offline* untuk memperoleh parameter yang terbaik untuk digunakan dalam perancangan pengendalinya. Adapun model sistem yang diperoleh dengan identifikasi secara *offline* pada model PAC dengan SISO adalah sebagai berikut:

$$A = \begin{bmatrix} 0.9952 & -0.0101 & 0.0016 & 0.0028 & 0.0009 & -0.0015 & 0.0002 & 0.0039 \\ -0.0020 & 0.9460 & 0.2570 & 0.0330 & -0.0515 & -0.0208 & -0.0147 & -0.0615 \\ -0.0020 & -0.1467 & 0.7221 & -0.3110 & 0.0525 & -0.1451 & 0.0566 & -0.2897 \\ -0.0010 & -0.0342 & -0.1450 & 0.6903 & 0.5995 & -0.0574 & 0.0968 & 0.7070 \\ 0.0002 & 0.0157 & 0.0085 & -0.4113 & 0.7564 & 0.3271 & -0.0802 & 0.5522 \\ 0 & -0.0032 & 0.0316 & 0.0658 & -0.1332 & 0.5663 & 0.7949 & 0.2680 \\ -0.0002 & -0.0013 & -0.0619 & -0.1060 & -0.0592 & -0.6828 & 0.3320 & 0.6752 \\ -0.0004 & -0.0140 & -0.0370 & 0.0376 & -0.2211 & 0.3095 & -0.4442 & 0.5944 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0011 \\ -0.0019 \\ 0.0062 \\ 0.0036 \\ 0.0004 \\ -0.0017 \\ 0.0007 \\ 0.0011 \end{bmatrix}$$

$$C = [-0.6690 \quad -0.0541 \quad 0.0417 \quad 0.0170 \quad -0.0053 \quad 0.0035 \quad -0.0035 \quad 0.0068]$$

$$D = [3.6260e - 006]$$

Matrix A, B, C dan D diatas digunakan setelah dilakukan pengujian *observability* dan *controllability* pasangan model sistem tersebut. Nilai eigen juga harus didalam *unit circle* untuk menjamin kestabilan sistem. Eigen value dari model yang digunakan untuk perancangan pengendali adalah:

$$Ev = \begin{bmatrix} 0.3039 + 0.9422i \\ 0.3039 - 0.9422i \\ 0.7455 + 0.6100i \\ 0.7455 - 0.6100i \\ 0.6829 \\ 0.9950 \\ 0.8882 \\ 0.9378 \end{bmatrix}$$

Nilai eigen diatas menunjukkan bahwa sistem berada didalam *unit circle* yang menunjukkan bahwa sistem yang digunakan sudah stabil. Apabila diuji *observability* dan *controllability*, maka sistem sudah *fully observability* dan *fully controllability*. Dengan demikian, model diatas bisa digunakan dalam perancangan MPC dengan *constraint*.

3.2 Pengendali MPC SISO dengan *Constraint*

Pengendali MPC dengan *constraints* ini dirancang dengan menggunakan *m-file* pada perangkat lunak Matlab. Program *m-file* yang dirancang terdiri dari 2 *m-file*: satu *m-file* berisi program utama dan satu *m-file* berisi program perhitungan fungsi kriteria. Program *m-file* yang berisi perhitungan fungsi kriteria ini akan dipanggil oleh program utama sebagai alternatif dalam perhitungan optimasi pengendali MPC dengan *constraint*.

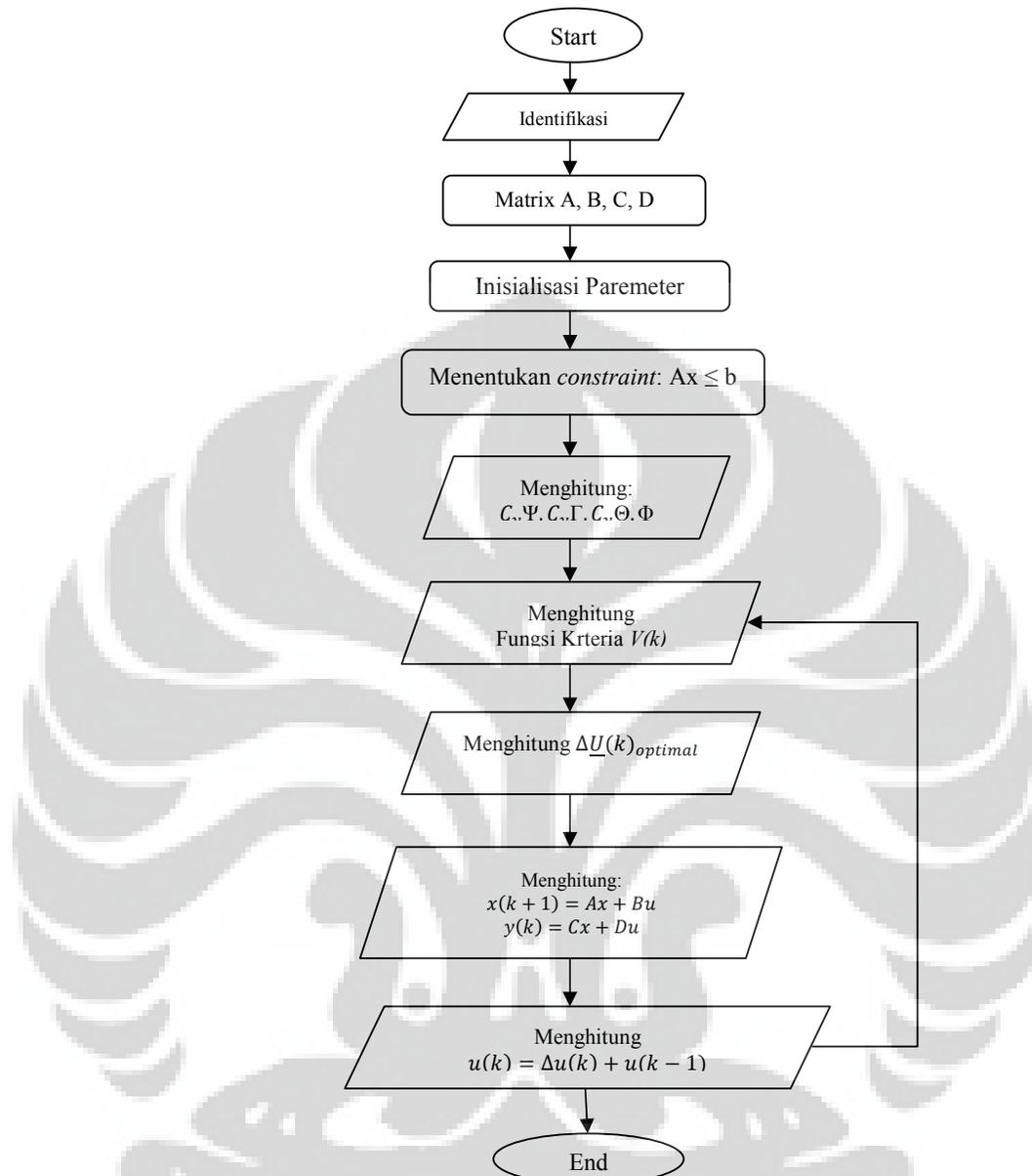
Sinyal keluaran sistem dari kedua program *m-file* yang telah dirancang ini akan ditampilkan dalam bentuk grafik yang terdiri dari dua *line*: *line* merah untuk sinyal referensi yang diberikan dan *line* biru untuk sinyal keluaran sistem. Hasil

pengendali yang telah dirancang dikatakan baik apabila pada grafik ditunjukkan grafik sinyal keluaran sistem yang mengikuti perubahan sinyal referensi yang diberikan.

3.2.1 Algoritma MPC SISO dengan *Constraint*

Sebelum pengendali MPC Adaptif dengan *constraint* dibuat, terlebih dahulu pengendali MPC dengan *constraint* dirancang. Pengendali MPC yang dirancang adalah menggunakan satu masukan dan satu keluaran atau sering disebut sebagai SISO (*single input single output*). Adapun alur algoritma sederhana pengendali MPC SISO dengan *constraint* yang dirancang menggunakan program *m-file* ditunjukkan pada gambar 3.2.





Gambar 3.2. Diagram alir algoritma pengendali MPC dengan *constraint*

Diagram alir pada gambar 3.2. diatas merupakan algoritma yang digunakan dalam merancang pengendali MPC dengan *constraint* pada program *m-file* yang telah dibuat. Program utama *m-file* yang dibuat berisikan algoritma 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 diinisialisasi terlebih dahulu.

Nilai $\underline{x}(k)$ dan $\underline{u}(k)$ mula-mula dibuat nol. Dengan demikian, inisiali parameter untuk pengendali MPC ini adalah: $t_{awal} = 0$, $T_s = 8$, $t_{akhir} = 3000$, $m = 1$, $p = 1$, $n = 8$. T_{akhir} ditentukan 3000 detik karena model untuk pengendali ini diperoleh dari PAC dengan t_{akhir} 3000 detik.

- 2) Menentukan nilai parameter-parameter pengendali MPC dengan *constraints* seperti H_p (horizon prediksi), H_u (horizon pengendali), \underline{Q} , \underline{R} , dan batasan-batasan perubahan sinyal kendali (*slew rate*) serta nilai sinyal referensinya. Nilai parameter yang digunakan pada perancangan ini adalah $H_p = 5$, $H_u = 2$ dengan $\underline{Q} = 10000 \underline{I}_{(p \times H_p)}$ dan $\underline{R} = 10000 \underline{I}_{(m \times H_u)}$. Akan tetapi, nilai H_p , H_u , \underline{Q} dan \underline{R} akan divariasikan untuk menganalisa pengaruh dari nilai parameter-parameter tersebut terhadap performansi pengendali MPC.
- 3) Menentukan batasan sinyal kendali untuk mengoptimasi pengendali MPC yang dirancang. Batasan sinyal kendali yang ditetapkan adalah:

$$-10000 \leq u(k) \leq 300$$

$$-10 \leq \Delta u(k) \leq 10$$

- 4) Membentuk sinyal referensi $T(k)$ yang diinginkan. Bentuk sinyal referensi dan sinyal pengendali akan di-*plot* untuk melihat performansi dari pengendali yang telah dirancang. Sinyal pengendali yang dibentuk adalah:
 - $T = 22$ untuk 1500 detik pertama
 - $T = 20$ untuk 1500 detik berikutnya
- 5) Menghitung nilai matrix $C_y \Psi$, $C_y \Gamma$, $C_y \Theta$ dan Φ untuk dipakai pada perhitungan kriteria di *m-file* yang lainnya.

$$C_y \Psi = \begin{bmatrix} -0.6658 & -0.0513 & 0.0127 & -0.0019 & 0.0088 & -0.0022 & 0.0040 & -0.0105 \\ -0.6625 & -0.0433 & -0.0047 & -0.0134 & 0.0106 & -0.0060 & 0.0050 & -0.0025 \\ -0.6592 & -0.0329 & -0.0139 & -0.0164 & 0.0025 & -0.0027 & -0.0036 & 0.0066 \\ -0.6559 & -0.0219 & -0.0173 & -0.0105 & -0.0085 & 0.0064 & -0.0083 & 0.0045 \\ -0.6527 & -0.0114 & -0.0172 & 0.0005 & -0.0145 & 0.0105 & -0.0005 & -0.0028 \end{bmatrix}$$

$$C_y \Gamma = \begin{bmatrix} -0.0003 \\ -0.0009 \\ -0.0016 \\ -0.0024 \\ -0.0032 \end{bmatrix}$$

$$C_y \Theta = \begin{bmatrix} -0.0003 & 0 \\ -0.0009 & -0.0003 \\ -0.0016 & -0.0016 \\ -0.0024 & -0.0016 \\ -0.0032 & -0.0024 \end{bmatrix}$$

$$\Phi = \begin{bmatrix} D & D \\ D & D \\ D & D \\ D & D \\ D & D \end{bmatrix} = \begin{bmatrix} 0.3626e-05 & 0.3626e-05 \\ 0.3626e-05 & 0.3626e-05 \\ 0.3626e-05 & 0.3626e-05 \\ 0.3626e-05 & 0.3626e-05 \\ 0.3626e-05 & 0.3626e-05 \end{bmatrix}$$

- 6) Membuat matrix persamaan dan pertidaksamaan seperti persamaan (2.22) dan (2.23) dari batasan-batasan yang telah ditentukan, sehingga diperoleh matrix:

$$\underline{F} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, F_1 = \begin{bmatrix} -10 \\ -10 \\ -10000 \\ -10000 \\ 10 \\ 10 \\ 10000 \\ 10000 \end{bmatrix}$$

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

- 7) Menghitung nilai $\Delta U(k)_{opt}$ dilakukan dengan memanfaatkan fungsi *fmincon* pada Matlab. Fungsi *fmincon* () akan memanggil fungsi kriteria, nilai-nilai parameter sistem, batasan-batasan sinyal kendali dan hasil perhitungan persamaan model prediksi.
- 8) Nilai $\Delta u(k)$ yang digunakan untuk memperbarui sinyal kendali hanya nilai pada baris pertama matriks Δu , sedangkan isi baris yang lainnya dibuang karena pada proses pencuplikan berikutnya sudah didapatkan nilai $\Delta u(k)$ yang baru. Maka nilai $u(k)$ yang harus diberikan ke *plant* adalah sebagai berikut :

$$u(k) = \Delta u(k) + u(k-1)$$

- 9) Memplot grafik sinyal pengendali dan sinyal keluaran sistem sepanjang nilai t_f untuk setiap T_s untuk ditampilkan setelah program *m-file* selesai dijalankan.

Program utama *m-file* kemudian dijalankan selama 3000 detik. Pada saat program diajalan, maka program terlebih dahulu membaca inisialisasi parameter yang diberikan, dan kemudian melakukan algoritma dibawah ini.

Pada saat program dijalankan, algoritma yang terjadi adalah:

- Melakukan perhitungan nilai $C_y\Psi$, $C_y\Gamma$, $C_y\Theta$ dan Φ
- Minimasi fungsi kriteria menggunakan fungsi *fmincon* () sehingga diperoleh nilai $\Delta U(k)_{opt}$.
- Menghitung nilai keluaran dengan persamaan $\underline{y}(k) = \underline{C} \underline{x}(k) + \underline{D} \underline{u}(k)$.
- Menghitung nilai keadaan dengan persamaan $\underline{x}(k + 1) = \underline{A} \underline{x}(k) + \underline{B} \underline{u}(k)$.
- Menghitung nilai sinyal kendali $\underline{u}(k)$.
- Memasukkan nilai $\underline{x}(k + 1)$ menjadi $\underline{x}(k)$ untuk pencuplikan selanjutnya.
- Memasukkan nilai $\underline{u}(k)$ mejadi $\underline{u}(k - 1)$ untuk pencuplikan selanjutnya.
- Menampilkan grafik sinyal referensi dan sinyal keluaran sitem,

3.2.2 Menghitung Fungsi Kriteria

Fungsi kriteria dalam bentuk *m-file* ini berisi rumus perhitungan fungsi kriteria $V(k)$ dan disimpan dengan nama 'kriteriasiso.m' untuk dipanggil pada program utama 'cobasiso.m' sebagai argument masukan dari fungsi *fmincon* ().

Fungsi ini menggunakan \underline{x}_{aug} sebagai argument masukan dan nilai $V(k)$ sebagai keluaran. Fungsi 'kriteriasiso.m' ini menggunakan algoritma sebagai berikut:

- Mengambil nilai $\Delta \underline{u}(k)$ dari \underline{x}_{aug} .

Vektor \underline{x}_{aug} yang dibuat dengan nilai $H_p=5$ dan $H_u = 2$ adalah:

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

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

2. Menghitung nilai vektor $\underline{E}(k)$ melalui persamaan:

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

Sehingga diperoleh nilai awal:

$$T(1) = \begin{bmatrix} 22 \\ 22 \\ 22 \\ 22 \end{bmatrix}$$

$$E(1) = T(1) = \begin{bmatrix} 22 \\ 22 \\ 22 \\ 22 \end{bmatrix}$$

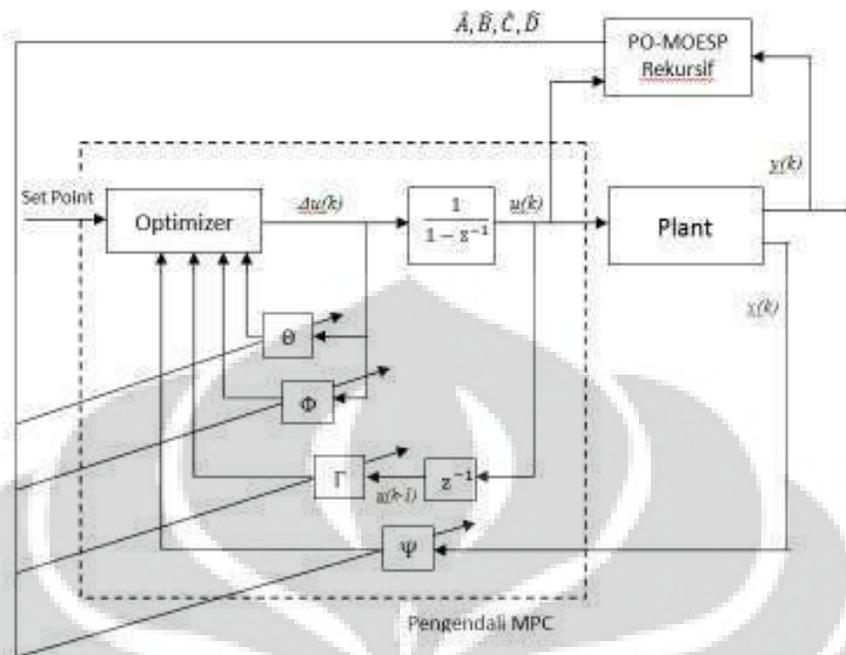
3. Menghitung nilai fungsi kriteria $V(k)$ melalui persamaan:

$$V(k) = [\underline{E}(k) - (\underline{C}_y \underline{\Theta} + \Phi) \Delta \underline{U}(k)]^T \underline{Q} (\underline{E}(k) - (\underline{C}_y \underline{\Theta} + \Phi) \Delta \underline{U}(k)) + \Delta \underline{U}^T(k) \underline{R} \Delta \underline{U}(k).$$

3.3 Pengendali MPC SISO Adaptif

Pengendali MPC SISO Adaptif merupakan penggabungan dari pengendali MPC SISO dengan *Constraint* dengan identifikasi rekursif. Pengendali MPC SISO Adaptif ini meng-*update* masukan identifikasi rekursif berupa u dan y yang diperoleh dari keluaran pengendali MPC SISO. Bagan diagramnya dapat dilihat pada gambar 3.3.

Gambar 3.3 menunjukkan bahwa pengendali MPC menggunakan model $\hat{A}, \hat{B}, \hat{C}, \hat{D}$ hasil estimasi dari identifikasi rekursif menggunakan algoritma PO-MOESP rekursif. Keluaran pengendali MPC berupa u yang kemudian menggenerasi y digunakan sebagai masukan dari algoritma PO-MOESP rekursif. Dengan demikian, keluaran dari identifikasi akan selalu di-*update* berdasarkan keluaran pengendali MPC SISO yang juga selalu di-*update*.

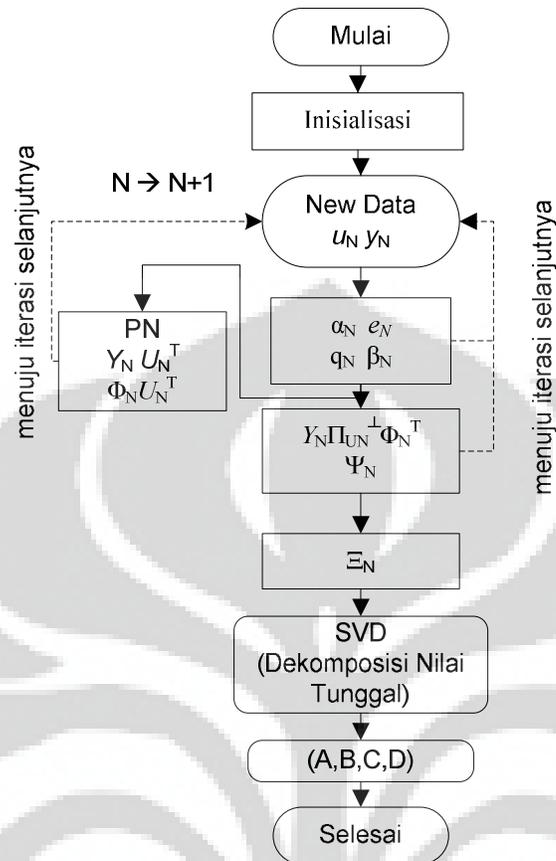


Gambar 3.3. Diagram Pengendali MPC Adaptif

Pengendali MPC SISO Adaptif dirancang dengan menggunakan pengendali MPC SISO dengan *constraint* yang dikombinasikan dengan identifikasi rekursif. Sehingga, algoritma yang digunakan pada perancangan MPC SISO adaptif ini adalah algoritma pengendali MPC SISO dengan *constraint* digabungkan dengan algoritma identifikasi rekursif yang menggunakan algoritma PO-MOESP rekursif.

Identifikasi dengan algoritma PO-MOESP rekursif ini dilakukan pada model PAC yang telah dimodifikasi menjadi model SISO. Model PAC dimodifikasi karena pengendali MPC yang telah dirancang juga untuk sistem SISO. Sehingga, identifikasi rekursif ini akan menghasilkan nilai $\hat{A}, \hat{B}, \hat{C}, \hat{D}$ untuk sistem SISO sebanyak iterasi tertentu. Dan nilai $\hat{A}, \hat{B}, \hat{C}, \hat{D}$ untuk setiap nilai iterasi akan digunakan oleh pengendali MPC SISO dengan *constraint* yang telah dirancang. Sehingga, pengendali MPC juga akan menampilkan grafik sinyal keluaran sistem dan sinyal referensi untuk setiap iterasi tersebut.

Algoritma PO-MOESP rekursif yang digunakan untuk memperoleh nilai $\hat{A}, \hat{B}, \hat{C}, \hat{D}$ yang dipakai sebagai model untuk merancang pengendali MPC SISO pada PAC. Algoritma identifikasi dengan menggunakan PO-MOESP rekursif ditunjukkan oleh diagram alir pada gambar 3.4 berikut ini.



Gambar 3.4. Diagram alir algoritma PO-MOESP rekursif

Gambar 3.4 diatas menunjukkan algoritma PO-MOESP rekursif. PO-MOESP adalah MOESP biasa dengan konstruksi instrumental variables dari pengukuran masukan lampau dan kelauran lampau. Dimensi sistem LTI yang diidentifikasi diasumsikan memiliki representasi sebagai berikut:

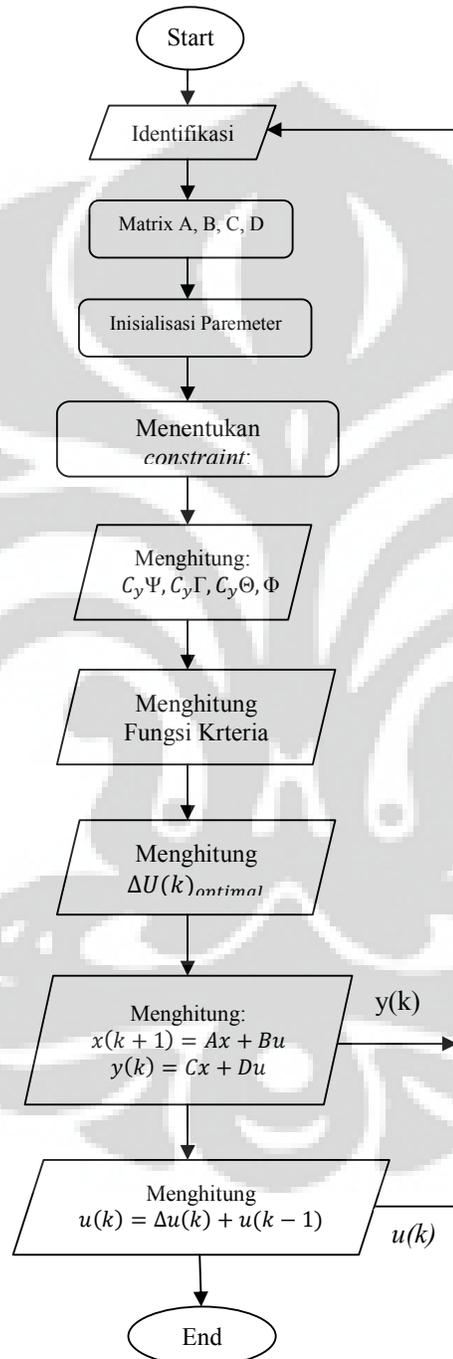
$$x_{k+1} = Ax_k + Bu_k$$

$$y_k = Cx_k + Du_k$$

dimana $x_k \in^n$, $u_k \in^r$. Matriks A harus stabil dan pasangan matriks (A,C) harus *observable*.

Pengendali MPC SISO Adaptif yang dirancang disimpan dalam program *m-file* yang diberi nama 'cobasiso2.m' yang juga akan dijalankan sama halnya seperti program *m-file* MPC yang telah dibahas sebelumnya. Karena pengendali MPC Adaptif merupakan penggabungan antara identifikasi rekursif dan pengendali MPC dengan *constraint*, maka program *m-file* yang digunakan untuk

perancangan pengendali MPC Adaptif juga merupakan penggabungan antara program *m-file* pengendali MPC SISO dengan *constraint* dan program *m-file* algoritma PO-MOESP rekursif. Adapun algoritma dari pengendali MPC Adaptif ini dapat dilihat pada gambar 3.5.



Gambar 3.5. Diagram alir algoritma pengendali MPC SISO Adaptif dengan *constraint*.

BAB 4

HASIL SIMULASI DAN ANALISIS

Pada bab 4 ini dianalisa hasil simulasi pengendali MPC SISO Adaptif dengan *constraint* yang dirancang menggunakan program *m-file* seperti yang telah dijelaskan pada bab 3. Simulasi yang dilakukan menggunakan berlangsung selama 3000 detik dengan waktu pencuplikan selama 8 detik. Simulasi dilakukan dengan memvariasikan nilai parameter H_p , H_u , Q , dan R untuk menganalisa pengaruh dari perubahan parameter tersebut terhadap performansi sistem. Variasi nilai parameter ini juga dilakukan untuk mencari nilai parameter terbaik yang dapat digunakan pada perancangan pengendali sehingga dapat menghasilkan sinyal keluaran terbaik. Model MPC yang dirancang adalah model linear SISO yang masukannya merupakan hasil dari identifikasi model PAC, yang telah dimodifikasi menjadi SISO, selama 3000 detik.

4.1 Simulasi dan Analisis Hasil Pengendali MPC SISO dengan *Constraint*

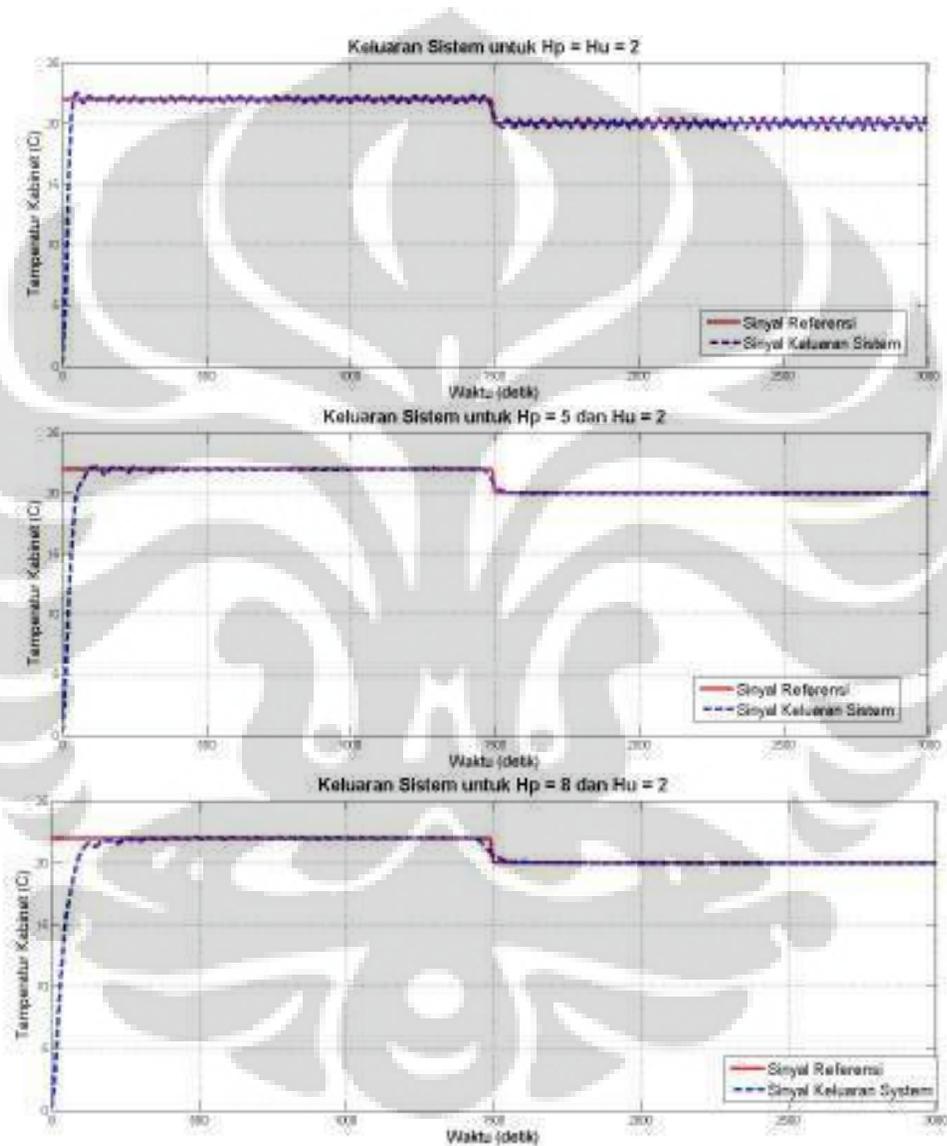
Sebelum melakukan simulasi dengan pengendali MPC SISO dengan menggunakan identifikasi rekursif, terlebih dahulu disimulasikan pengendali MPC SISO dengan indentifikasi *offline*. Pengendali dengan identifikasi secara *online* bisa dilakukan setelah pengendali menghasilkan keluaran sistem yang baik untuk identifikasi secara *offline*.

Dalam perancangan pengendali MPC SISO dengan *constraint*, diperlukan nilai parameter pengendali yang tepat karena setiap nilai parameter yang digunakan akan mempengaruhi keluaran sistem. Oleh karena itu, dilakukan simulasi dengan memvariasikan setiap parameter yang digunakan hingga diperoleh nilai terbaik dari masing-masing parameter.

4.1.1 Hasil Simulasi dan Analisis untuk Variasi Nilai *Prediction Horizon* (H_p)

Untuk melihat pengaruh *prediction horizon* (H_p) terhadap hasil pengendali MPC SISO dengan *constraint*, dilakukan beberapa percobaan dengan memvariasi nilai *prediction horizon* (H_p), sedangkan nilai *control horizon* (H_u) dijaga tetap,

yaitu pada nilai $H_u = 2$. Variasi nilai H_p yang digunakan adalah 2, 5, 8, dan 10 sehingga bisa dilihat pengaruh perubahan nilai H_p terhadap performansi pengendali MPC. Nilai faktor bobot kesalahan Q yang digunakan adalah $1000I_{H_p}$ dan faktor bobot perubahan sinyal kendali R juga bernilai sama yaitu $1000I_{H_u}$. Hasil simulasi variasi nilai H_p ini ditunjukkan pada gambar 4.1

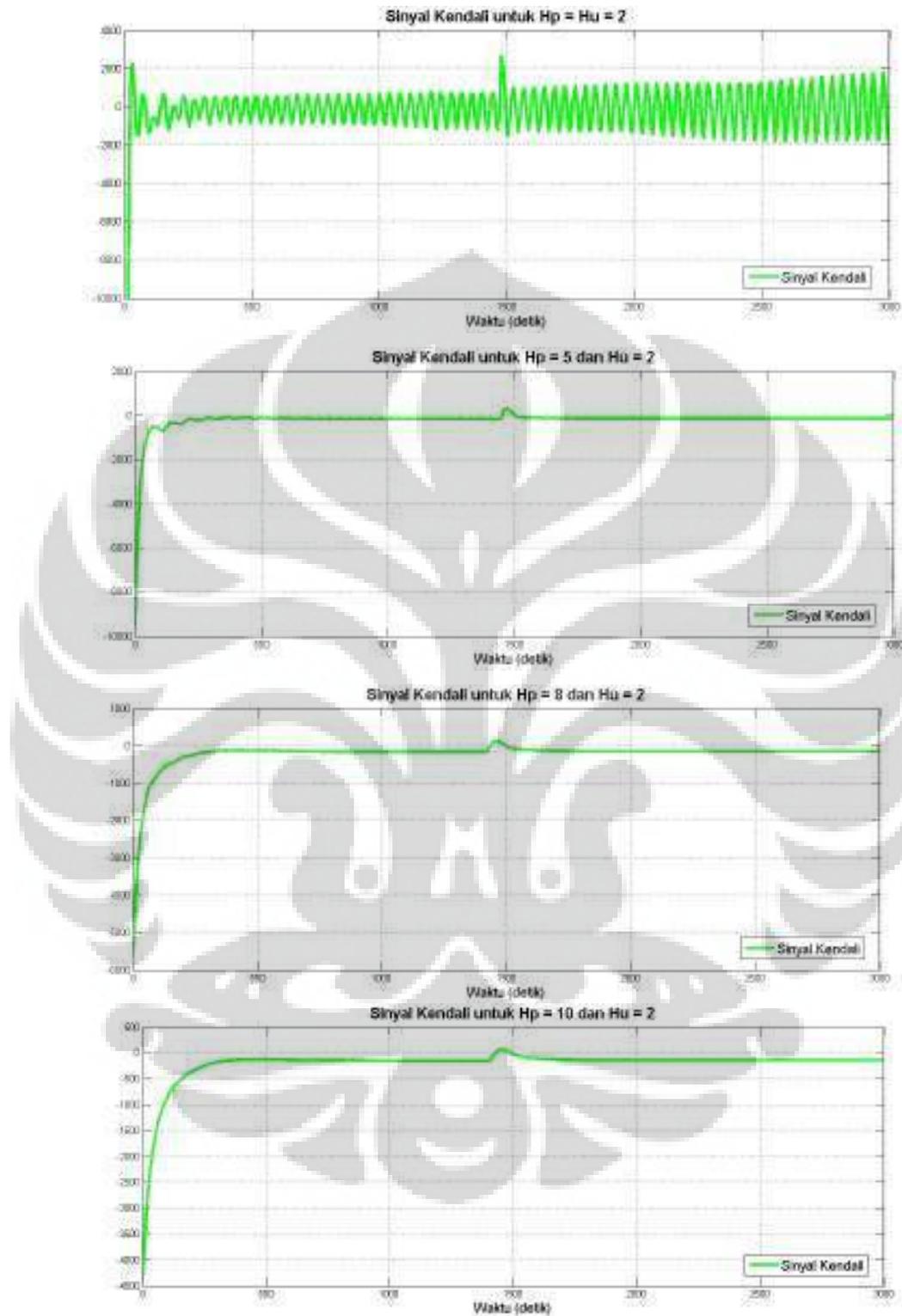




Gambar 4.1. Keluaran sistem hasil pengendali MPC SISO dengan *constraint* dengan variasi nilai H_p

Dari hasil simulasi diatas terlihat bahwa semakin besar nilai H_p yang digunakan, maka sinyal keluaran sistem menjadi lebih stabil, karena pada prinsipnya H_p berfungsi untuk memprediksi keluaran kedepannya. Oleh karena itu, semakin besar nilai H_p , maka makin jauh keluaran kedepannya yang terprediksi. Akan tetapi, nilai H_p yang semakin besar ini membuat sistem menjadi lambat dalam mengikuti perubahan sinyal referensi. Perilaku sinyal keluaran sistem dalam mengikuti perubahan sinyal referensi dapat dilihat pada nilai $H_p = 5$ dan $H_p = 10$. Pada perubahan sinyal referensi pada detik 1500, sinyal keluaran sistem untuk nilai $H_p = 10$ cenderung lebih cepat berubah mendahului sinyal referensinya bila dibandingkan dengan sinyal keluaran sistem untuk nilai $H_p = 5$. Dengan demikian, nilai $H_p = 5$ dipilih sebagai nilai H_p yang digunakan karena sinyal keluaran yang dihasilkan lebih cepat dan *overshoot* yang dihasilkan masih cenderung kecil.

Untuk perubahan sinyal kendali (masukan sistem) untuk variasi nilai H_p yang sama ditunjukkan pada gambar 4.2.

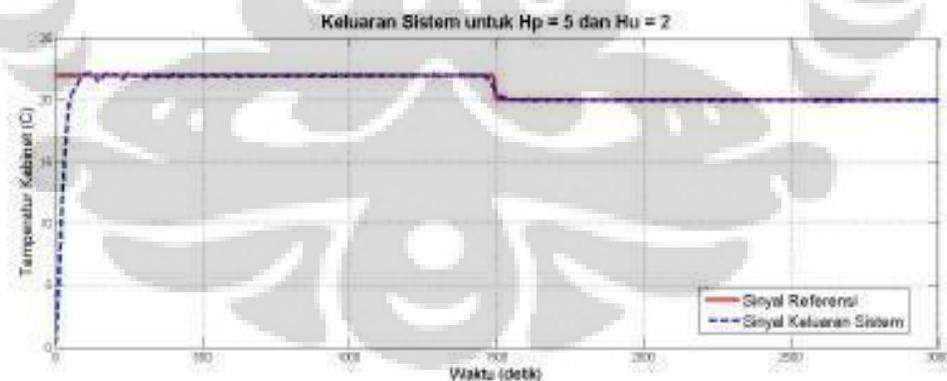


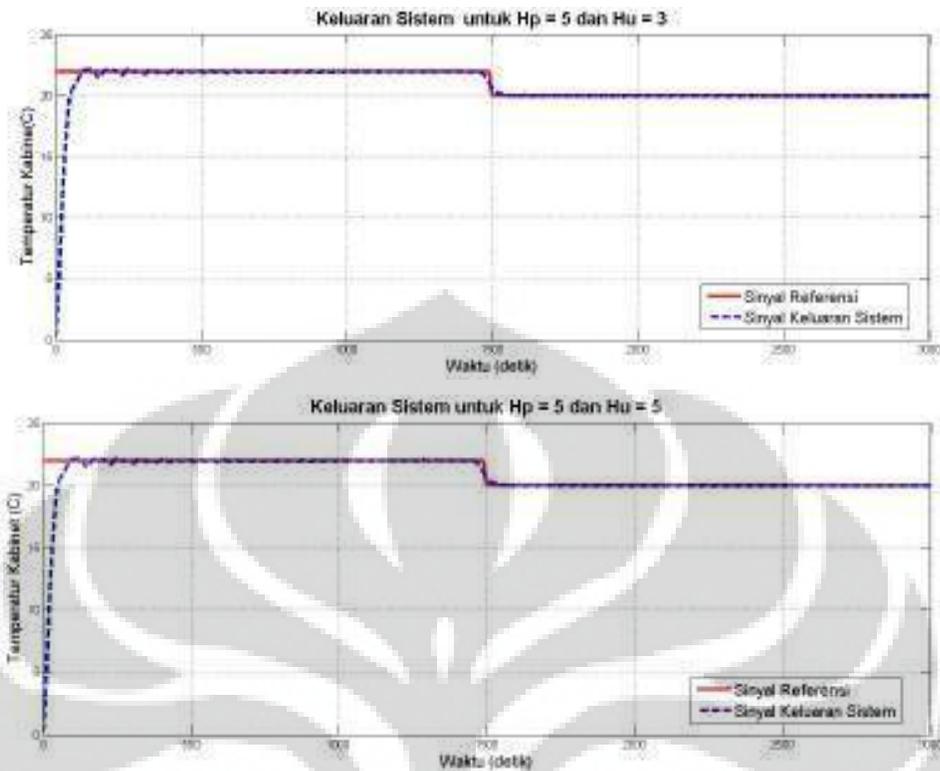
Gambar 4.2. Sinyal kendali hasil pengendali MPC dengan *constraint* dengan variasi nilai H_p

Pada gambar 4.2 ditunjukkan bahwa perubahan sinyal kendali menjadi semakin konstan apabila nilai H_p nya semakin diperbesar. Meskipun perubahan sinyal kendali pada nilai $H_p = 2$ sangat besar, namun pada saat diberikan sinyal referensi, nilai perubahannya menjadi lebih kecil karena sinyal kendali mampu mengikuti sinyal referensi. Untuk variasi nilai H_p 5, 8 dan 10, perubahan sinyal kendali sudah adalah serupa karena pada $H_p = 5$, sinyal kendali sudah menjadi stabil.

4.1.2 Hasil Simulasi dan Analisis untuk Variasi Nilai *Control Horizon* (H_u)

Setelah nilai H_p divariasikan pada bagian sebelumnya, pada bagian ini nilai H_u divariasikan untuk melihat pengaruh dari perubahan H_u terhadap keluaran sistem hasil pengendali MPC SISO yang telah dirancang. Untuk variasi nilai H_u ini, nilai H_p dijaga tetap, yaitu pada nilai $H_p = 5$ yang dipilih setelah melakukan variasi nilai H_p pada bagian 4.1.1. Variasi nilai H_u yang digunakan adalah 2, 3 dan 5. Nilai H_u tidak boleh melebihi nilai H_p seperti yang tertulis pada buku *Maciejowski*. Sedangkan untuk nilai faktor bobot kesalahan (Q) dan faktor bobot perubahan sinyal kendali (R) yang digunakan masih sama seperti pada bagian variasi H_p , yaitu $Q = 1000I_{H_p}$ dan $R = 1000I_{H_u}$. Hasil simulasi ditunjukkan pada gambar 4.3.

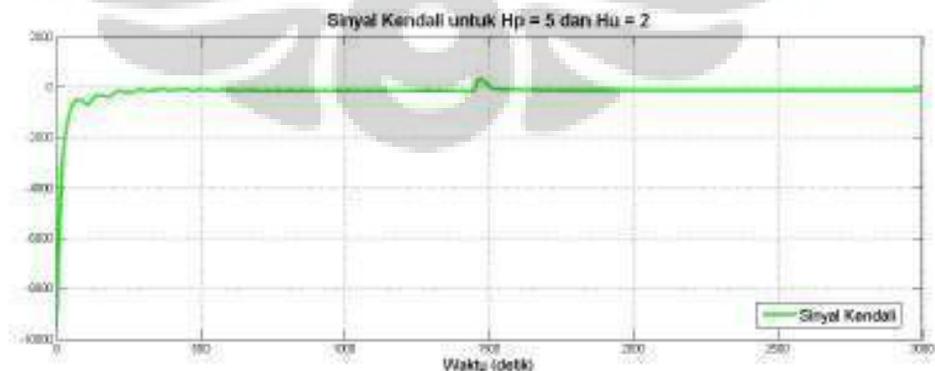


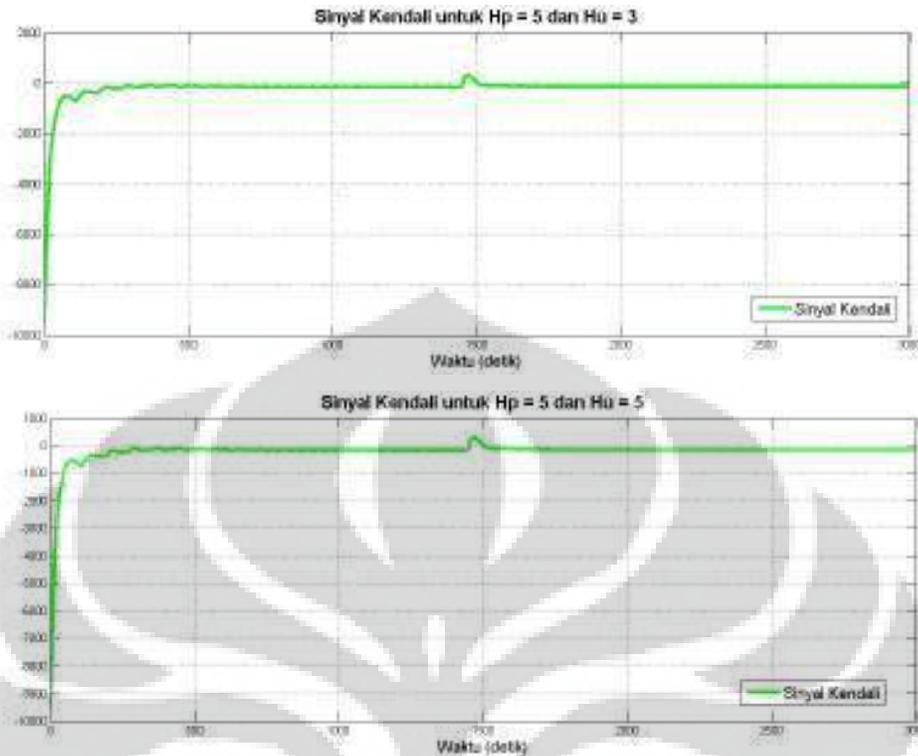


Gambar 4.3. Keluaran sistem hasil pengendali MPC SISO dengan *constraint* dengan variasi nilai H_u

Pengaruh perubahan nilai H_u pada hasil pengendali MPC SISO dengan *constraint* ini belum dapat dilihat dari grafik pada gambar 4.3. Dengan kata lain, grafik yang dihasilkan menunjukkan performansi yang sama untuk ke tiga nilai H_u yang berbeda.

Perubahan sinyal kendali dari variasi nilai H_u diatas ditunjukkan pada gambar 4.4.

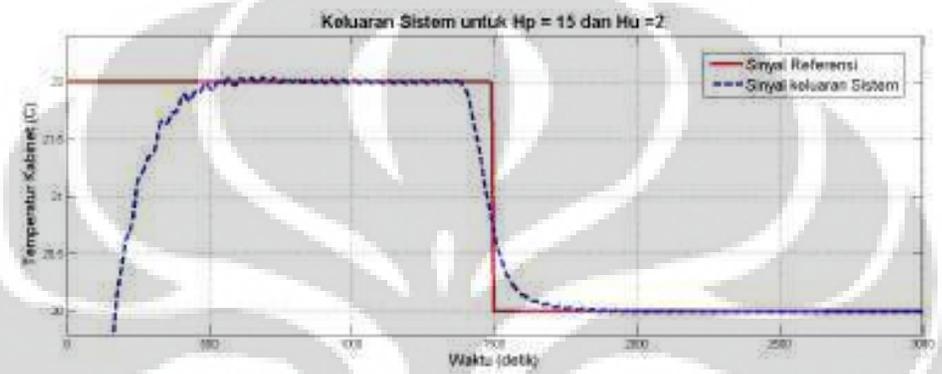




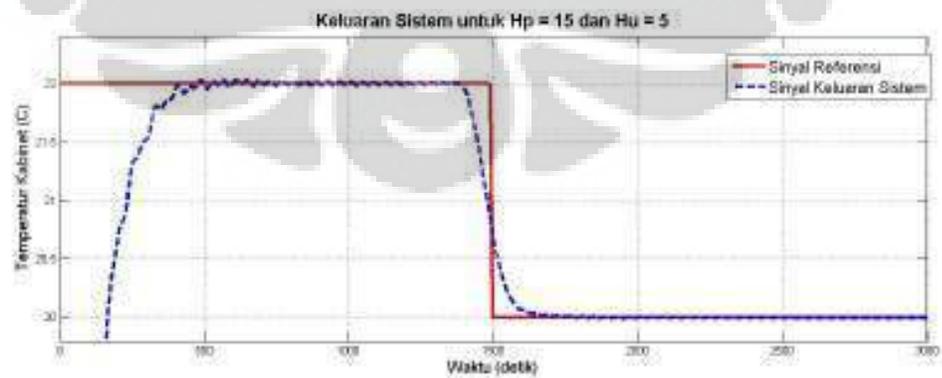
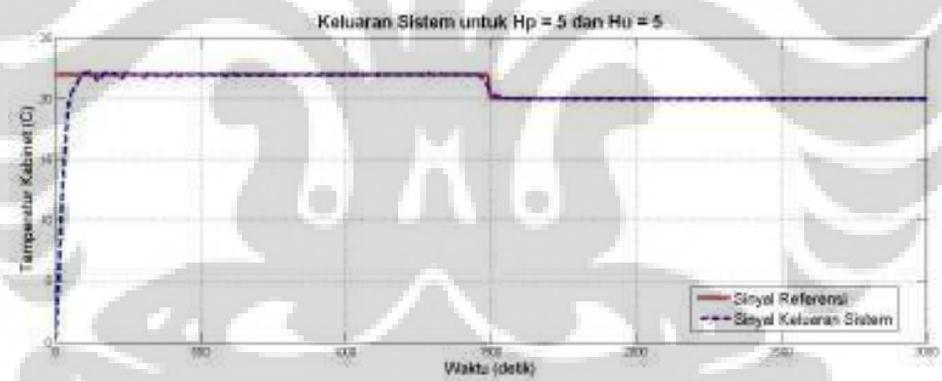
Gambar 4.4. Sinyal kendali hasil pengendali MPC dengan *constraint* dengan variasi nilai H_u

Gambar 4.4 menunjukkan perubahan sinyal kendali yang sama untuk ke tiga nilai H_u yang berbeda. Hal ini bisa disebabkan karena nilai faktor bobot perubahan sinyal kendali yang digunakan terlalu besar yaitu $\underline{R} = 1000$. Nilai \underline{R} yang besar dapat menekan perubahan sinyal kendali.

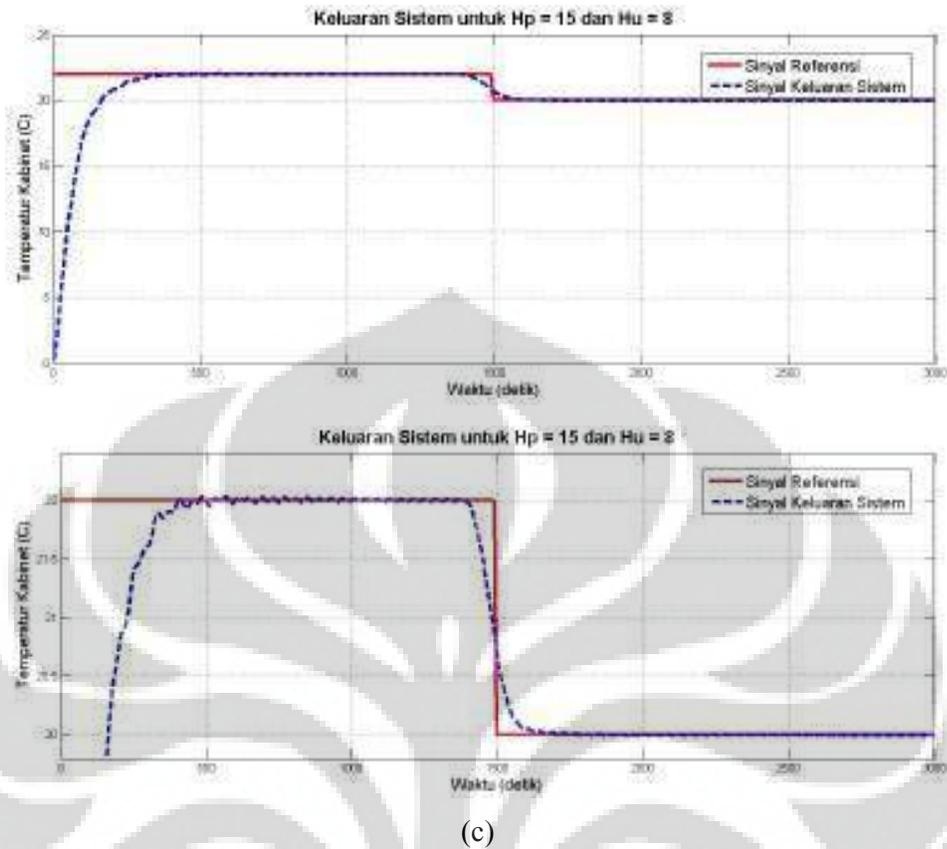
Karena pada gambar 4.3 belum bisa dilihat pengaruh variasi nilai H_u terhadap performansi sistem, maka penulis memperbesar nilai H_p menjadi $H_p = 15$ dan memperkecil matrix \underline{R} menjadi $R = 0.1$, sehingga pengaruh perubahan nilai H_u terhadap performansi sistem dapat terlihat. Nilai matrix bobot R dibuat sekecil mungkin agar perubahan sinyal kendali tidak terlalu ditekan, sehingga perubahan sinyal keluaran pun bisa tampak. Variasi nilai H_u adalah 2, 5 dan 8. Hasil simulasi ditunjukkan pada gambar 4.5.



(a)



(b)



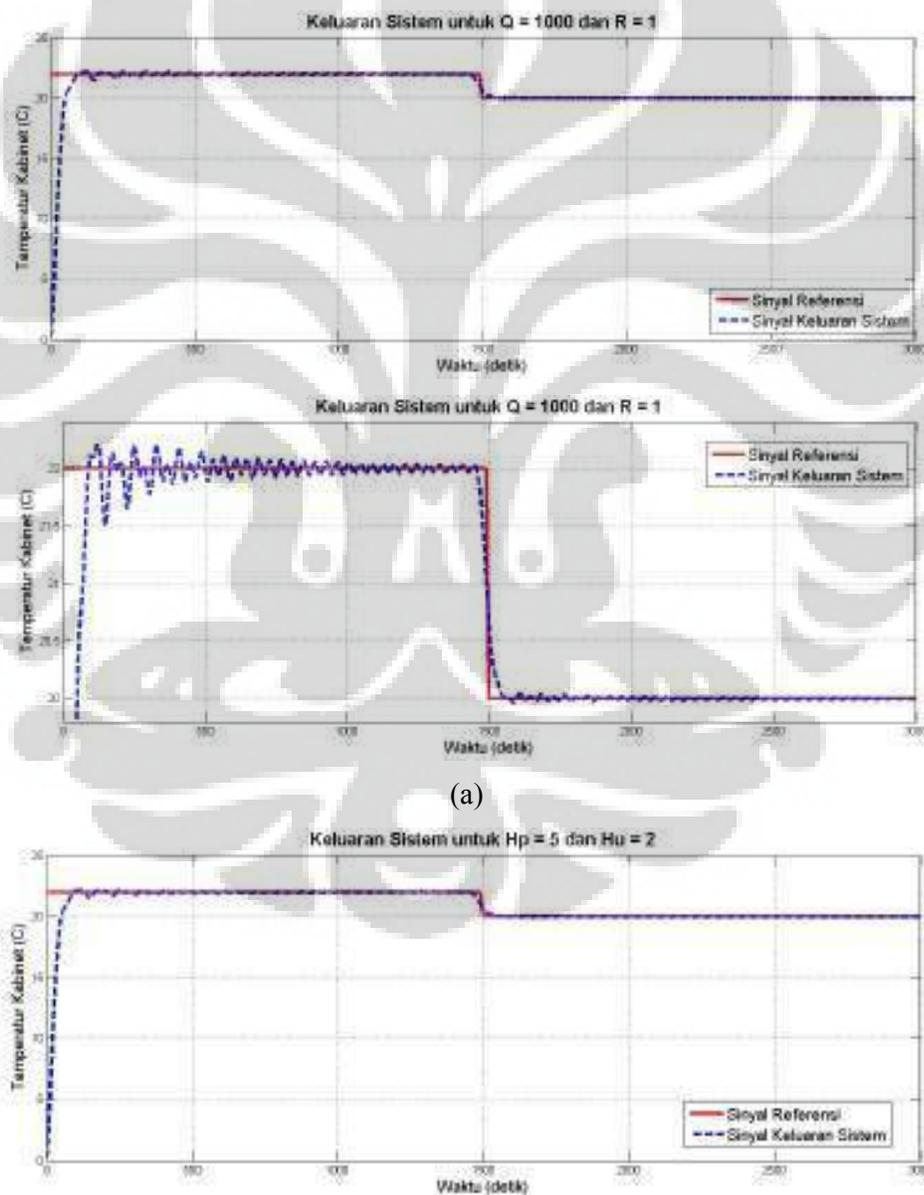
(c)
Gambar 4.5(a), (b), (c). Keluaran sistem hasil pengendali MPC SISO dengan *constraint* dengan variasi nilai Hu

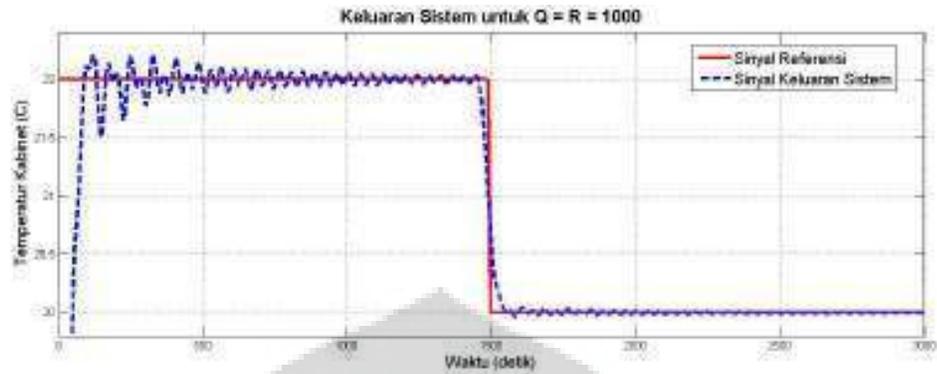
Setelah nilai H_p diperbesar menjadi $H_p = 15$, maka perubahan nilai H_u terhadap performansi sistem menjadi dapat dilihat. Dari grafik pada gambar 4.5, ditunjukkan bahwa semakin besar nilai H_u , maka sinyal keluaran sistem semakin cepat mengikuti perubahan sinyal referensi, meskipun perbedaan yang ditunjukkan tergolong kecil. Hal ini terjadi karena H_u berfungsi sebagai pengendali sinyal kendali, dimana semakin banyak sinyal kendali kedepannya yang terkendali, maka keluarannya semakin baik.

Dengan demikian, nilai H_u yang dipilih oleh penulis dalam memvariasikan nilai R dan Q pada bagian berikutnya adalah $H_u = 2$, karena semakin kecil nilai H_u yang digunakan maka semakin cepat pula respon sistem.

4.1.3 Hasil Simulasi dan Analisis untuk Variasi Faktor Bobot Perubahan Sinyal Kendali \underline{R}

Dari variasi H_p dan H_u yang telah dilakukan, diperoleh nilai terbaik yaitu $H_p = 5$ dan $H_u = 2$. Nilai ini digunakan untuk memvariasikan nilai faktor bobot perubahan sinyal kendali \underline{R} . Variasi nilai matrix \underline{R} dilakukan untuk melihat pengaruh perubahan besar matrix \underline{R} terhadap performansi hasil pengendali MPC SISO dengan *constraint*. Variasi nilai \underline{R} yang digunakan adalah $0.1I_{H_u}$, $1000I_{H_u}$ dan $10000I_{H_u}$. Sedangkan, nilai faktor bobot kesalahan \underline{Q} yang digunakan adalah $1000I_{H_p}$. Hasil simulasi ditunjukkan pada gambar 4.6.





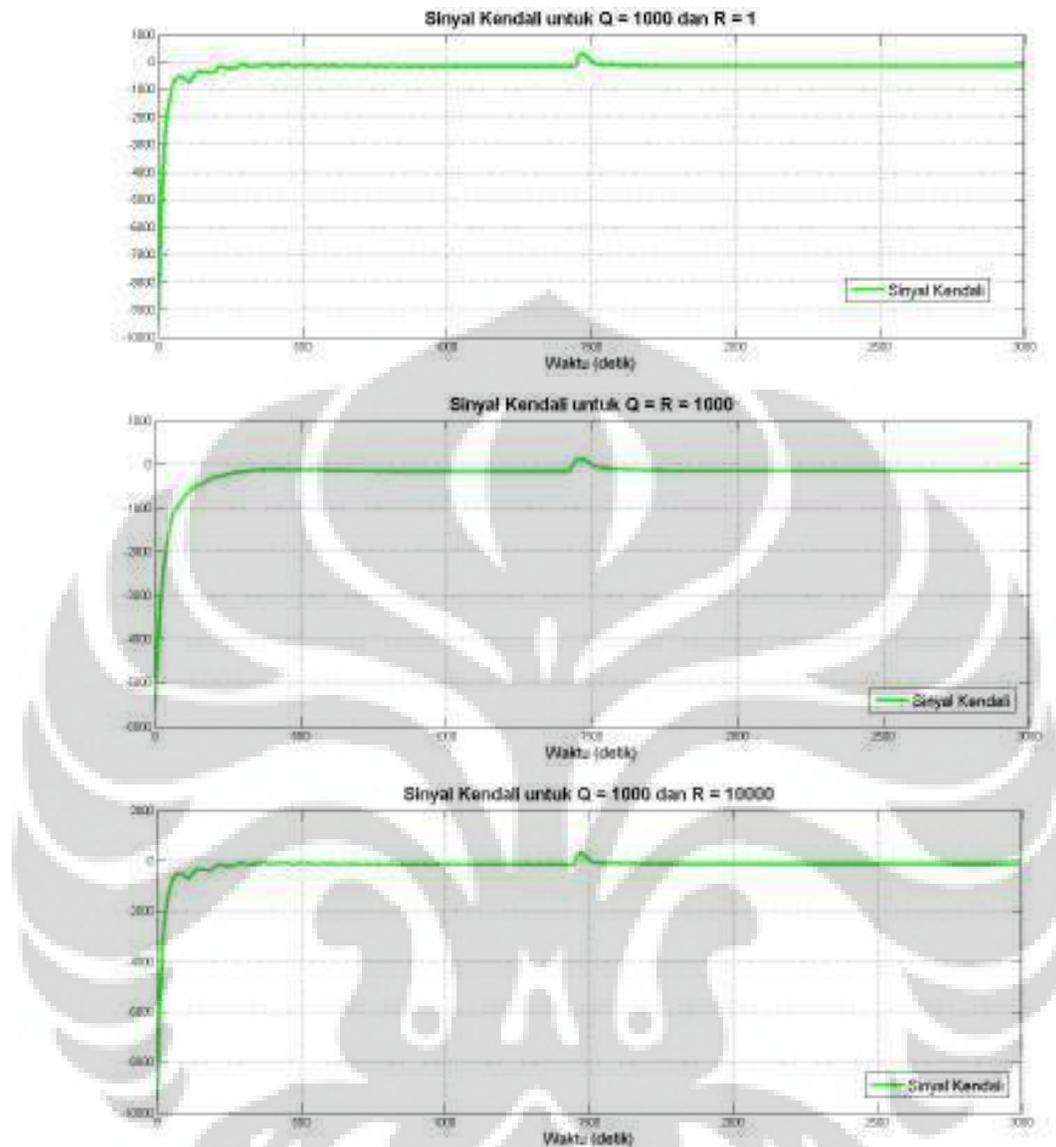
(b)



(c)

Gambar 4.6(a), (b), (c). Keluaran sistem hasil pengendali MPC SISO dengan *constraint* dengan variasi nilai faktor bobot perubahan sinyal kendali \underline{R} .

Gambar 4.6 belum bisa menunjukkan pengaruh dari perubahan nilai matrix \underline{R} terhadap performansi sistem, sehingga nilai parameter lainnya perlu dimodifikasi terlebih dahulu agar pengaruhnya bisa terlihat. Sementara itu, untuk perubahan sinyal kendalinya dapat dilihat pada gambar 4.7.

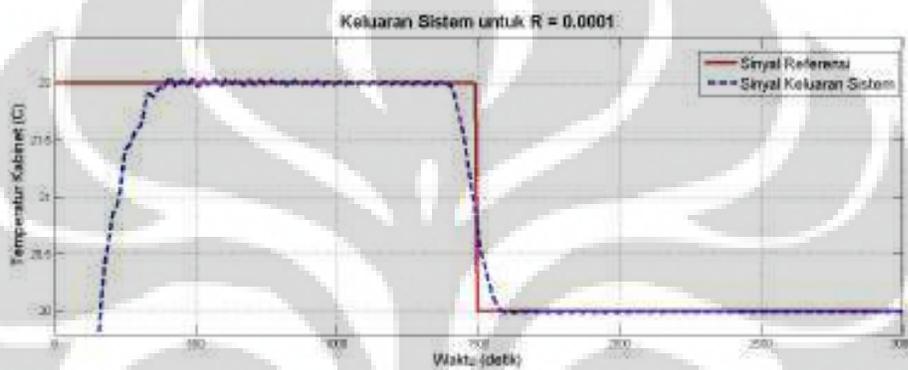
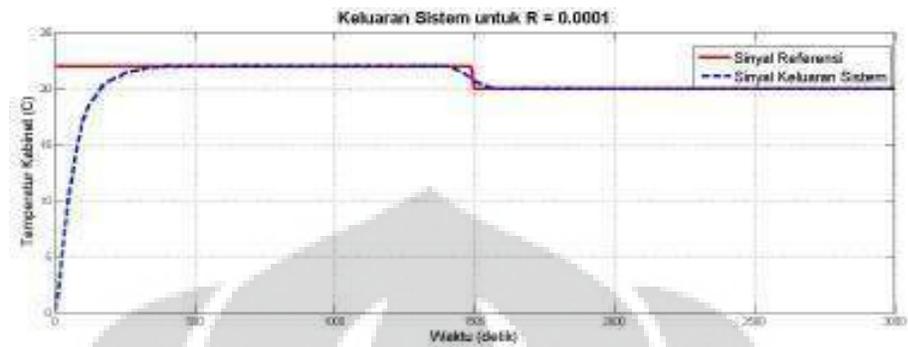


Gambar 4.7. Sinyal kendali hasil pengendali MPC SISO dengan *constraint* dengan variasi nilai faktor bobot perubahan sinyal kendali \underline{R} .

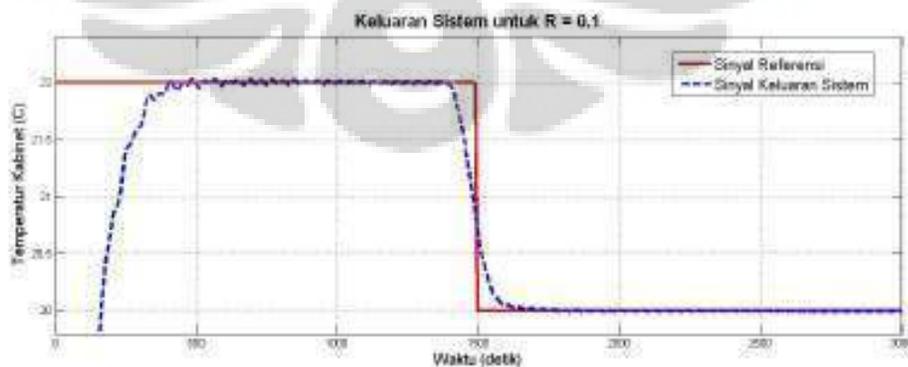
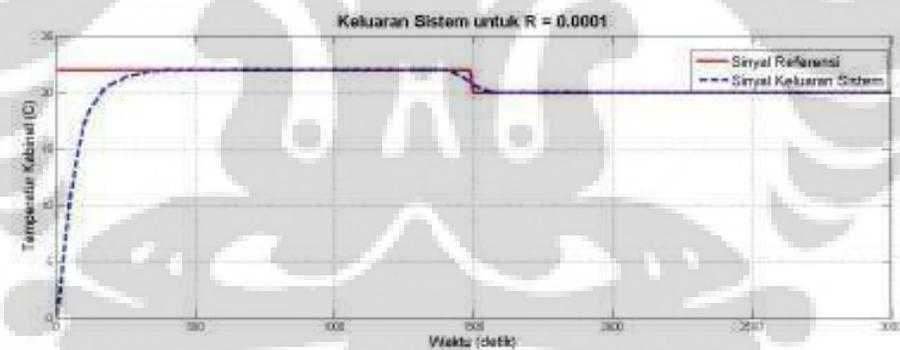
Gambar 4.7 menunjukkan bahwa variasi nilai \underline{R} yang diberikan tidak terlalu mempengaruhi perubahan sinyal kendali secara signifikan. Perbedaan kecil terjadi hanya pada bentuk sinyal keluaran sistem sebelum menjadi stabil, dimana pada saat $\underline{R} = 1$ dan $\underline{R} = 10000$ terdapat *ripple* pada sinyal kendali sebelum menjadi stabil, sedangkan pada $\underline{R} = 1000$, *ripple* tersebut tidak ditemukan.

Seperti halnya pada variasi nilai H_u , untuk melihat pengaruh perubahan nilai \underline{R} terhadap performansi sistem, penulis mengubah inputan nilai H_u dan H_p yang digunakan menjadi $H_u=15$ dan $H_u = 8$, dengan variasi nilai matrix \underline{R} adalah

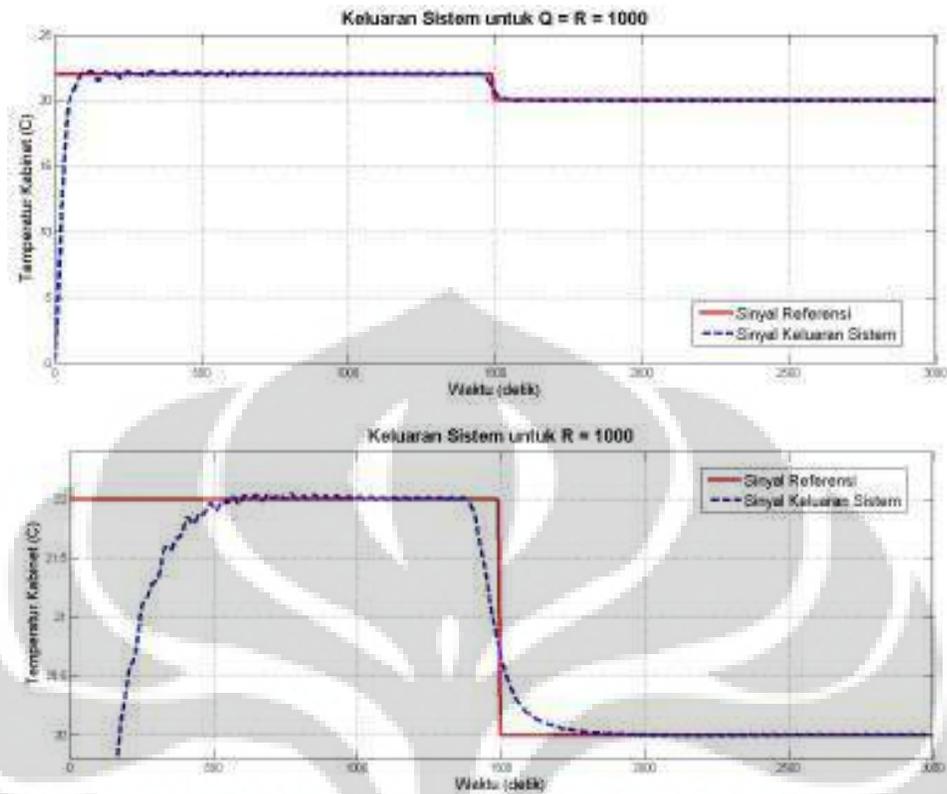
$0.001I_{Hu}$, $0.1I_{Hu}$ dan $100I_{Hu}$. Hasil simulasi untuk nilai parameter ini ditunjukkan pada gambar 4.8.



(a)



(b)



(c)

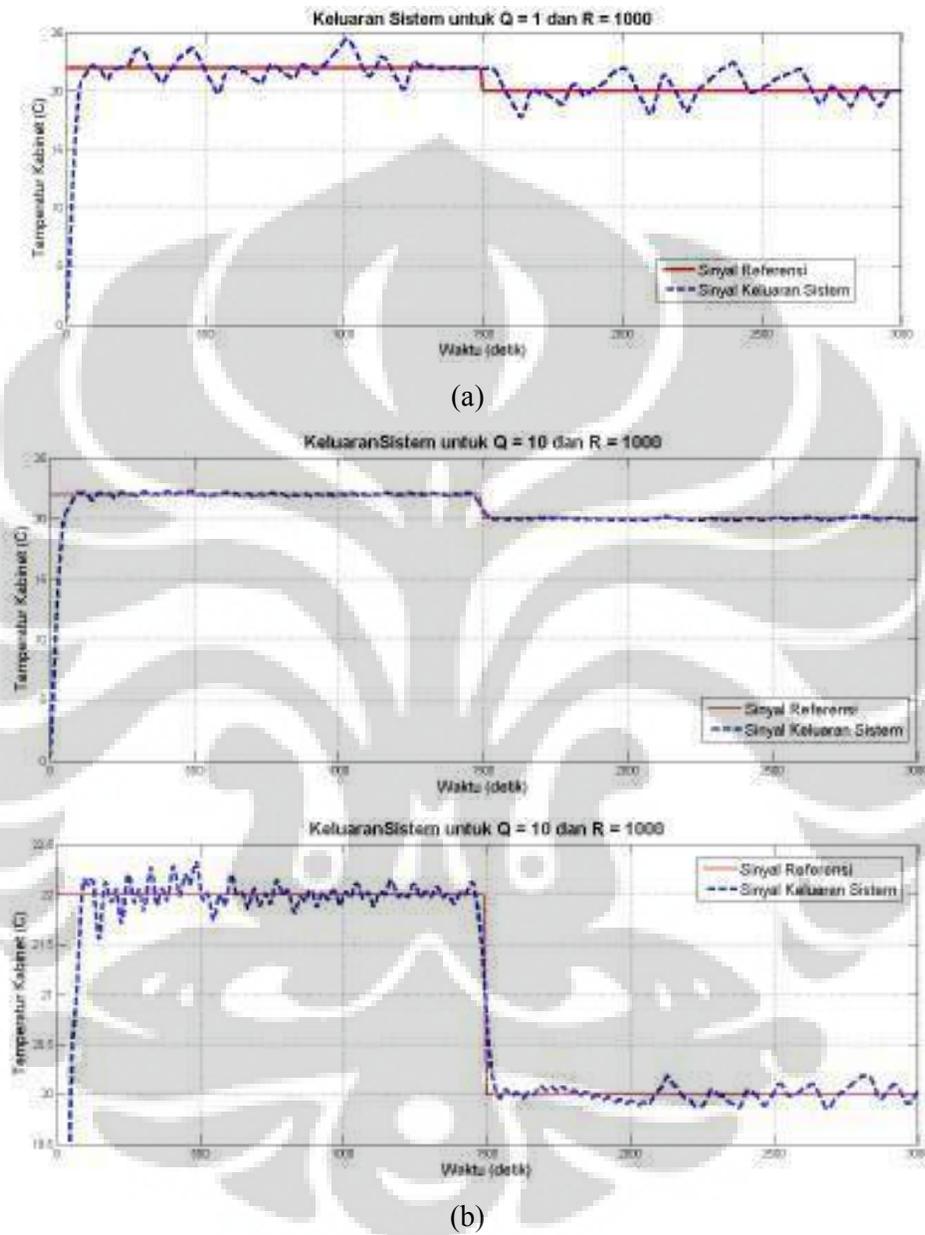
Gambar 4.8(a), (b), (c). Keluaran sistem hasil pengendali MPC SISO dengan *constraint* dengan variasi nilai faktor bobot perubahan sinyal kendali \underline{R} .

Gambar 4.8 menunjukkan bahwa semakin besar nilai matrix \underline{R} nya, maka semakin lambat respon sistemnya, dan sebaliknya dimana semakin kecil nilai matrix \underline{R} nya, maka semakin cepat pula respon sistemnya. Akan tetapi, perbedaan waktu respon sistemnya tidaklah terlalu besar. Dengan demikian, nilai matrix \underline{R} yang digunakan oleh penulis ada $\underline{R} = 1000I_{Hu}$ karena perubahan sinyal kendalinya lebih mulus.

4.1.4 Hasil Simulasi dan Analisis untuk Variasi Faktor Bobot Kesalahan \underline{Q}

Pada bagian ini akan dilakukan variasi nilai faktor bobot kesalahan \underline{Q} untuk melihat pengaruh dari perubahan nilai matrix \underline{Q} terhadap performansi keluaran sistem. Karena pada bagian sebelumnya telah dilakukan variasi nilai H_p , H_u dan \underline{R} , maka penulis menggunakan nilai H_p , H_u dan \underline{R} yang telah dipilih dari setiap variasinya, yaitu $H_p = 5$, $H_u = 2$ dan $\underline{R} = 1000I_{Hu}$. Nilai variabel-variable

ini digunakan untuk memvariasikan nilai bobot matrix Q . Simulasi variasi nilai Q ditunjukkan pada gambar 4.9.





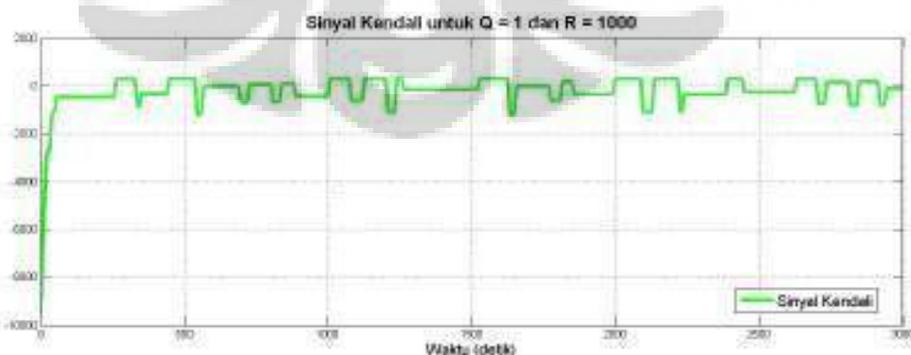
(c)

Gambar 4.9(a), (b), (c). Keluaran sistem hasil pengendali MPC SISO dengan *constraint* dengan variasi nilai faktor bobot kesalahan Q .

Dari hasil pengendali MPC SISO dengan *constraint* diatas, dapat dilihat bahwa penambahan nilai faktor bobot kesalahan Q membuat sinyal keluaran sistem menjadi lebih stabil. Dengan kata lain, Matriks bobot Q memiliki pengaruh pada error antara sinyal referensi dengan keluaran dan respon keluaran sistem.

Pada gambar 4.9 (a) dan (b) ditunjukkan bahwa penambahan nilai Q dari $Q = 1$ menjadi $Q = 10$ sudah mampu memperbaharui performansi sistem menjadi jauh lebih baik. Meskipun pada $Q = 10$ sinyal keluaran sistem sudah mengikuti sinyal referensi, *overshoot* setelah perubahan sinyal referensi masih cukup besar, untuk itu nilai Q ditambahkan menjadi $Q = 1000$. Pada $Q = 1000$, sinyal keluaran sistem sudah cenderung stabil sebelum sinyal referensi berubah dan hanya menghasilkan *overshoot* kecil setelah perubahan sinyal referensi, sehingga nilai $Q = 1000I_{Hp}$ dipilih untuk digunakan pada variasi nilai parameter selanjutnya.

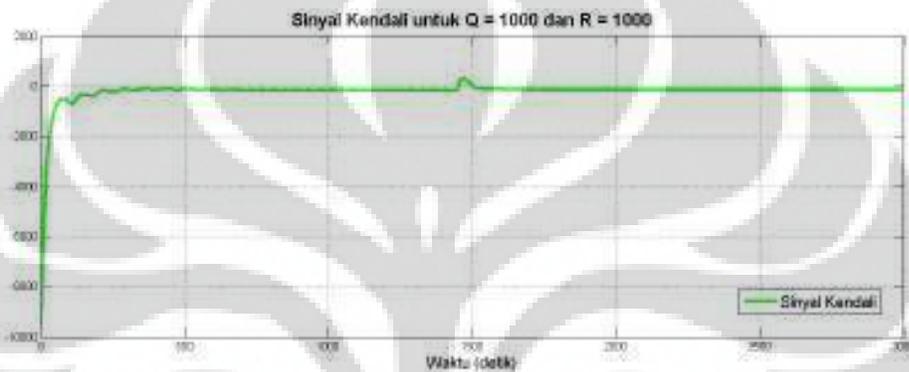
Perubahan sinyal kendali dari variasi Q yang telah dilakukan, ditunjukkan pada gambar 4.10 dibawah.



(a)



(b)



(c)

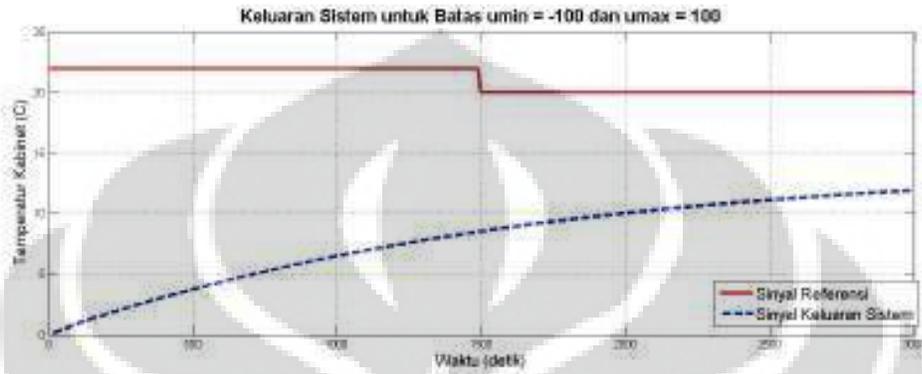
Gambar 4.10. Sinyal Kendali hasil pengendali MPC SISO dengan *constraint* dengan variasi nilai faktor bobot kesalahan Q .

Seperti yang dapat dilihat pada gambar 4.10(a), bahwa perubahan sinyal kendali belum stabil apabila kita memberikan nilai Q yang kecil. Perubahan sinyal kendali mulai membaik seiring dengan penambahan nilai Q dan menjadi stabil pada saat $Q = 1000$.

4.1.5 Pengaruh Variasi Batas Sinyal Kendali pada hasil pengendali MPC

Pada bagian sebelumnya telah dilakukan variasi nilai H_p , H_u , R dan Q untuk mengetahui pengaruh dari perubahan masing-masing nilai parameter tersebut terhadap performansi sinyal keluaran sistem. Dari variasi nilai-nilai ini juga diperoleh nilai parameter terbaik yang digunakan untuk menunjukkan performansi terbaik dari hasil perancangan pengendali MPC SISO dengan *constraint* yang telah dilakukan pada bab 3.

Oleh karena itu, dengan memanfaatkan nilai-nilai parameter yang telah diperoleh dari variasi sebelumnya, akan dianalisa pengaruh dari besar *constraint* yang digunakan terhadap performansi sinyal keluaran sistem. Nilai parameter yang digunakan pada variasi besar *constraint* ini antara lain $H_p = 5$, $H_u = 2$, $\underline{R} = 1000I_{Hu}$ dan $\underline{Q} = 1000I_{Hp}$. Hasil simulasi ditunjukkan pada gambar 4.11.



(a)



(b)

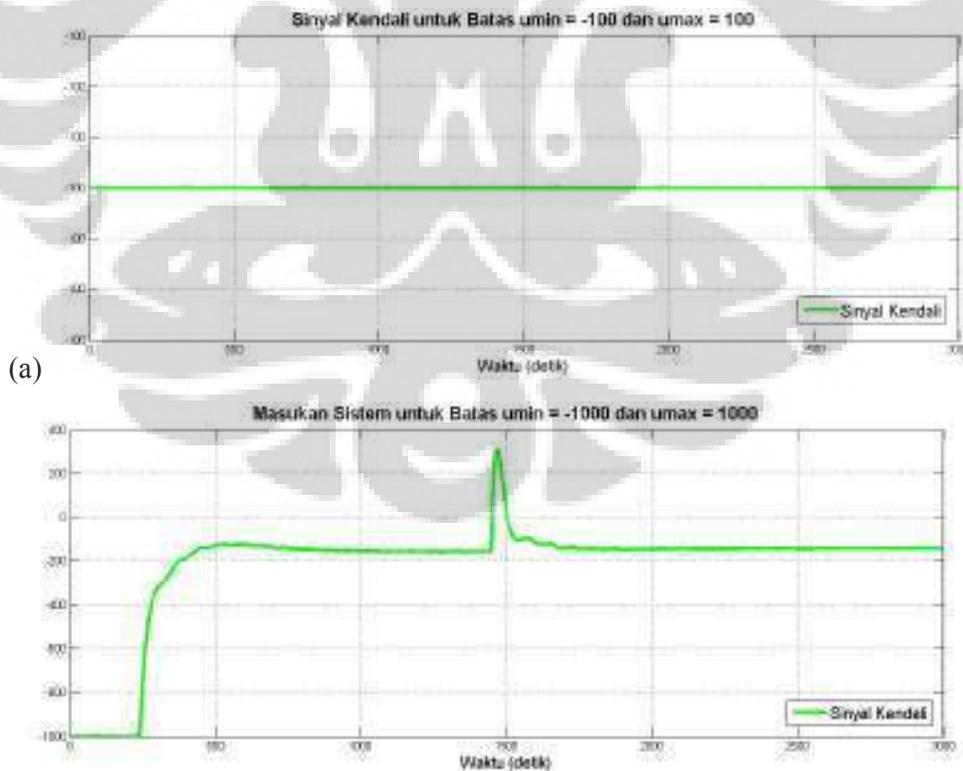


(c)

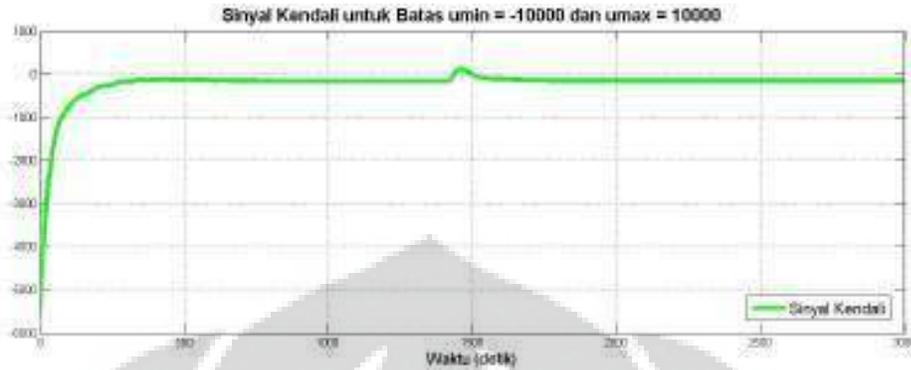
Gambar 4.11(a), (b), (c). Keluaran sistem hasil pengendali MPC SISO dengan *constraint* dengan variasi besar *constraint*

Model MPC SISO dengan *constraint* yang dirancang menghasilkan nilai akhir sinyal kendali, $u(k) = -141.0614$. Nilai $u(k)$ yang besar ini menyebabkan sinyal keluaran sistem dengan $u_{min} = -100$ (gambar 4.11(a)) menjadi tidak bagus dan sama sekali tidak mengikuti sinyal referensi yang diberikan. Hal ini disebabkan karena batasan tersebut tidak mencakup nilai-nilai perubahan $u(k)$ yang terjadi. Gambar 4.11(b) menunjukkan performansi sinyal keluaran sistem yang cenderung stabil karena batasan tersebut sudah mencakup nilai $u(k)$ yang dihasilkan oleh pengendali. Akan tetapi, respon sistem yang dihasilkan masih relatif lambat karena belum semua nilai sinyal kendali yang dihasilkan per percuplikannya bisa dicakup oleh batasan tersebut. Oleh karena itu, besar batasan sinyal pengendalnya dinaikkan menjadi ± 10.000 untuk memastikan agar setiap nilai sinyal kendali yang dihasilkan oleh pengendali MPC yang dirancang mampu dicakup untuk setiap perubahan nilainya. Grafik dengan batasan ini ditunjukkan pada gambar 4.11(c) dimana respon sistemnya menjadi lebih cepat dalam mengikuti perubahan sinyal referensi.

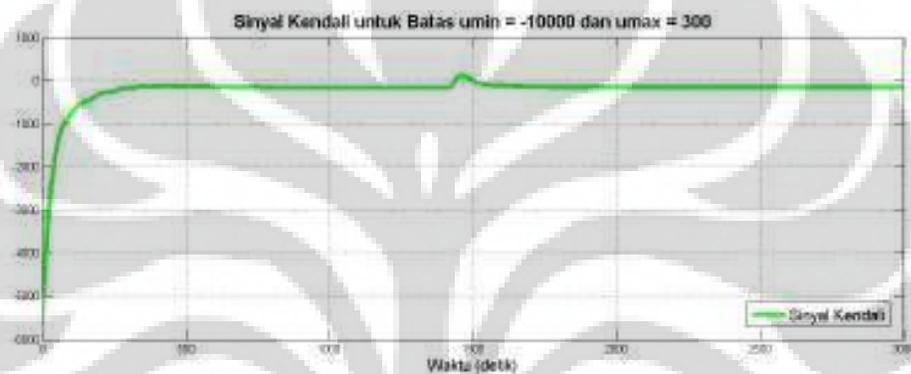
Sedangkan untuk perubahan sinyal kendali yang dihasilkan oleh pengendali MPC yang dirancang ditunjukkan pada gambar 4.12 dibawah.



(b)



(c)



(d)

Gambar 4.12(a), (b), (c), (d). Sinyal kendali hasil pengendali MPC SISO dengan *constraint* dengan variasi besar *constraint*

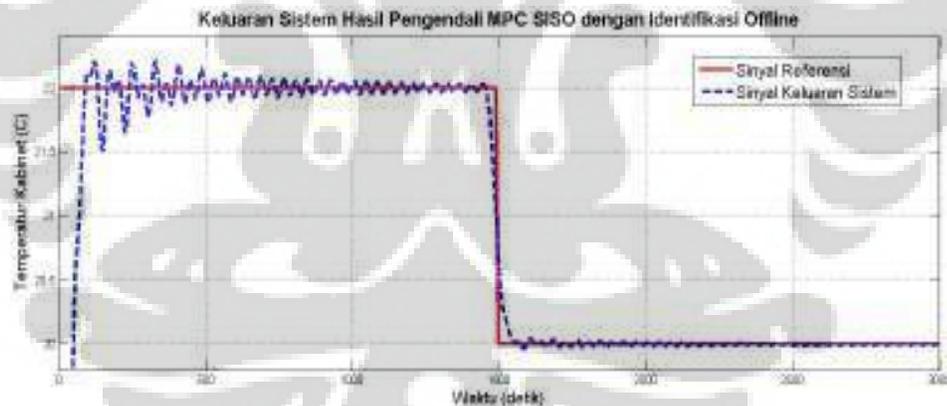
Pada gambar 4.12(a) diatas dapat dilihat bahwa untuk batasan sinyal kendali ± 100 hanya menunjukkan sinyal keluaran sistem yang nilai kendalinya berada pada rentang tersebut, sehingga bentuk sinyal kendalinya mejadi garis lurus saja. Gambar 4.12(b) menunjukkan batasan ± 1.000 sudah mencakup seluruh nilai sinyal kendali positif yang digenerasi oleh pengendali yang dirancang, namun untuk nilai sinyal kendali negatif yang dihasilkan masih belum tercakup semuanya. Hal ini yang mengakibatkan grafik yang dihasilkan tidak dimulai dari detik ke nol. Pada gambar 4.12(c) dengan batasan sinyal kendali ± 10.000 sudah mencakup seluruh sinyal kendali yang dihasilkan oleh pengendali MPC yang dirancang, baik yang bagian negatif maupun bagian yang positifnya. Oleh karena itu, grafiknya yang dihasilkan sudah menunjukkan sinyal kendali dari detik ke nol hingga detik ke 3000 dan sinyal kendalinya sudah stabil. Akan tetapi, setelah

grafik sinyal kendali diperbesar dengan menggunakan perangkat lunak Matlab, maka batasan $u_{min} = -6000$ dan $u_{max} = 300$ ternyata sudah mencakup semua nilai sinyal kendali yang dihasilkan oleh pengendali, seperti ditunjukkan pada gambar 4.12(c). Untuk itu, penulis memilih batas $u_{min} = -10.000$ dan $u_{max} = 300$ sebagai batas sinyal kendali terbaik.

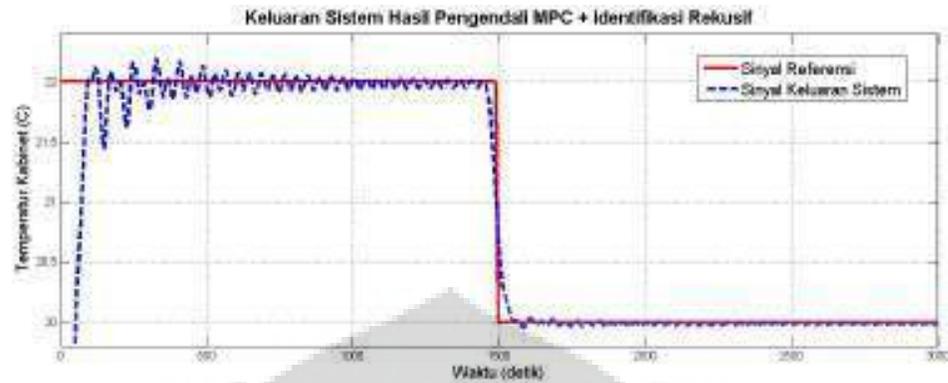
4.2 Simulasi Pengendali MPC SISO dengan Identifikasi Rekursif Secara Terpisah

Simulasi Pengendali MPC SISO dengan Identifikasi Rekursif secara terpisah ini belum tergolong pengendali MPC SISO adaptif karena program pengendali MPC SISO dengan *constraint* dibuat terpisah dengan program identifikasi rekursifnya. Dalam hal ini, pengendali MPC SISO nya menggunakan model yang merupakan hasil akhir dari identifikasi rekursif secara terpisah.

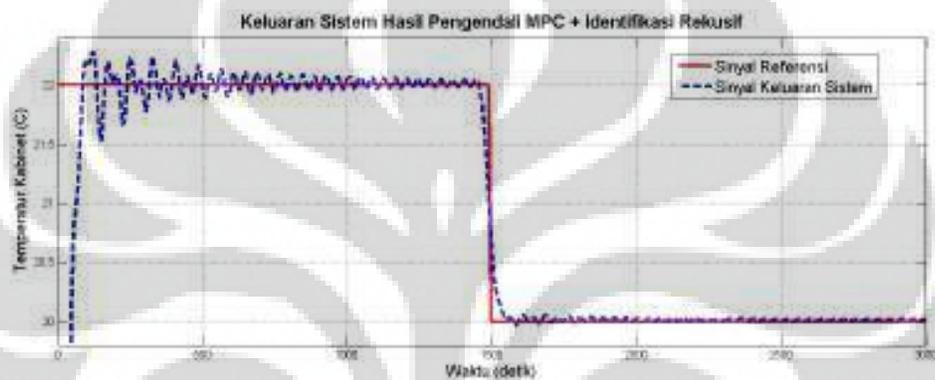
Simulasi ini dilakukan untuk melihat pengaruh identifikasi rekursif itu sendiri terhadap model keluarannya untuk setiap iterasi. Hasil dari iterasi *online* yang berbeda dapat dilihat dari gambar 4.13.



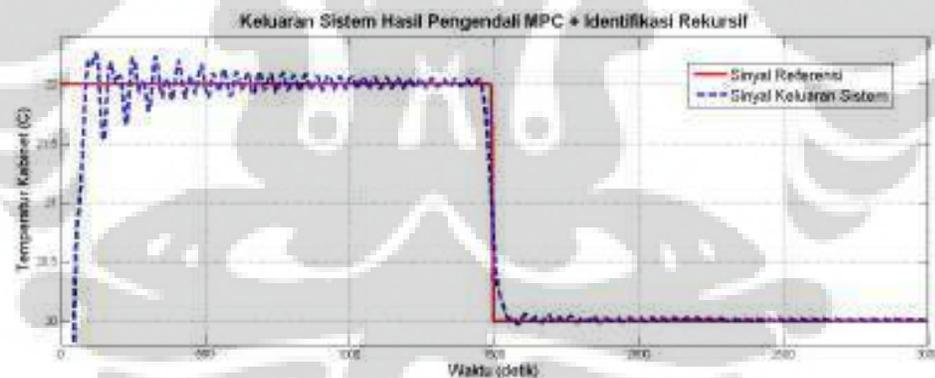
(a)



(b)



(c)



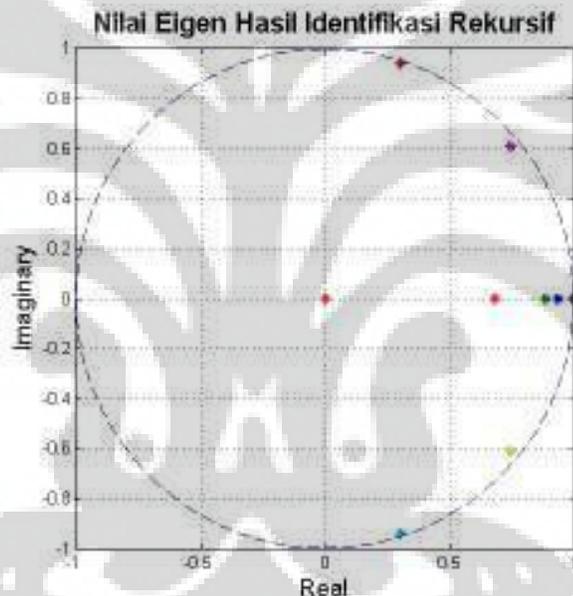
(d)

Gambar 4.13:(a). Keluaran sistem hasil pengendali MPC SISO dengan identifikasi *offline* secara terpisah; (b). Keluaran Sistem hasil pengendali MPC SISO untuk 200 iterasi rekursif identifikasi; (c) 150 iterasi rekursif identifikasi; (d) 100 iterasi rekursif identifikasi.

Gambar 4.13 (a) diatas menunjukkan hasil pengendali MPC SISO sebelum dengan identifikasi rekursif yang dapat dibandingkan dengan gambar 4.13(b), (c)

dan (d), yang merupakan hasil pengendali MPC SISO setelah dengan identifikasi rekursif secara terpisah. Apabila grafik sebelum dan setelah dengan identifikasi rekursif dibandingkan, maka dapat terlihat bahwa *overshoot* dihasilkan oleh pengendali MPC SISO sebelum dengan identifikasi rekursif, sedangkan setelahnya melu berkurang sejalan dengan bertambahnya jumlah iterasi *online* yang dilakukan, yaitu pada 200 iterasi *online*. Sehingga, dapat disimpulkan bahwa pengendali MPC dengan identifikasi rekursif terpisah menghasilkan performansi sistem yang lebih baik karena *overshoot* nya lebih kecil.

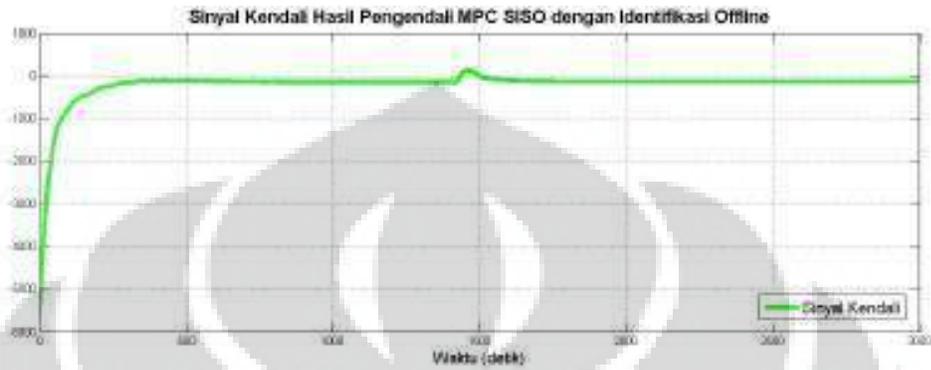
Nilai eigen dari identifikasi rekursif yang dilakukan dapat dilihat pada gambar 4.14, dimana nilai eigennya cenderung tidak berubah. Hal ini dikarenakan model yang digunakan adalah model linier.



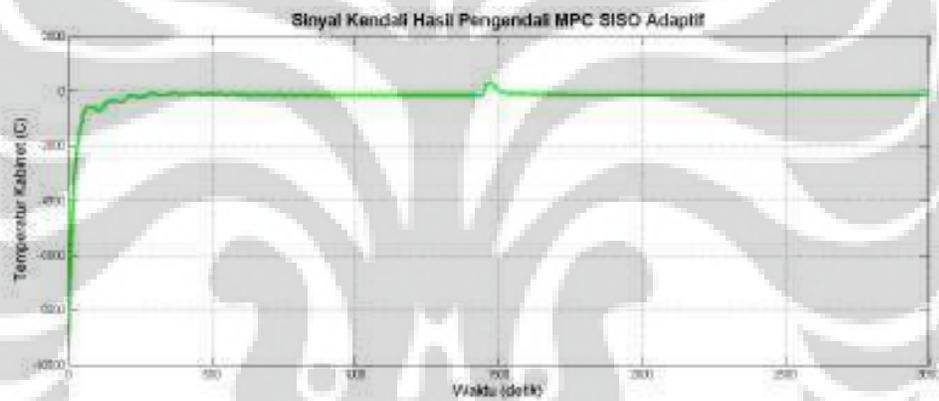
Gambar 4.14. Nilai eigen hasil identifikasi rekursif

Perubahan sinyal kendali hasil pengendali MPC SISO Adaptif dapat dilihat pada gambar 4.15. Gambar 4.15(a) menunjukkan perubahan sinyal kendali sebelum dengan identifikasi rekursif dan gambar 4.15(b), (c) dan (d) memperlihatkan perubahan sinyal kendali setelah dengan identifikasi rekursif. Sinyal kendali hasil pengendali setelah dengan identifikasi rekursif menghasilkan *ripple* sebelum menjadi stabil, sementara sinyal kendali sebelum identifikasi rekursif tidak. Dan perubahan sinyal kendali untuk variasi yang berbeda tidak menunjukkan perbedaan. Oleh karena itu, penulis menyimpulkan bahwa dengan

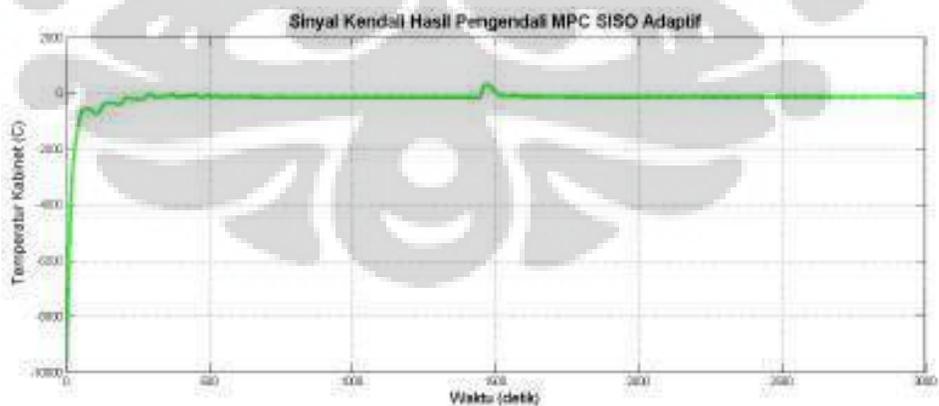
identifikasi *online*, performansi sistem hasil pengendali MPC SISO dengan *constraint* tidak terlalu berpengaruh besar, karena *ripple* kecil yang dihasilkan bisa diabaikan.



(a)



(b)



(c)



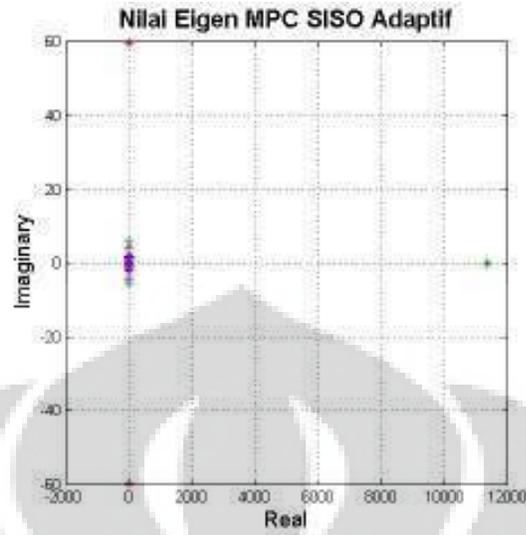
(d)

Gambar 4.15:(a). Sinyal kendali hasil pengendali MPC SISO dengan identifikasi *offline* secara terpisah; (b). Sinyal kendali hasil pengendali MPC SISO Adaptif untuk 200 iterasi online; (c). untuk 150 iterasi *online*; (d). untuk 100 iterasi *online*

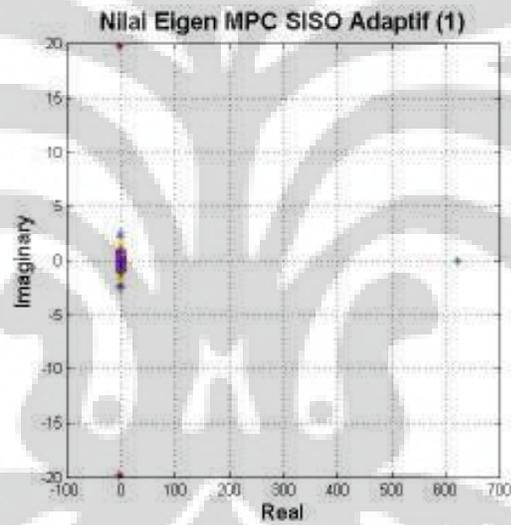
4.3 Simulasi Hasil Pengendali MPC SISO Adaptif

Pengendali MPC SISO Adaptif merupakan penggabungan dari pengendali MPC SISO dengan *Constraint* dengan identifikasi rekursif. Hal ini berbeda dengan sub bab sebelumnya yang pengendali MPC SISO-nya dibuat terpisah dengan identifikasi rekursifnya. Pengendali MPC SISO Adaptif disini meng-*update* masukan identifikasi rekursif berupa u dan y yang diperoleh dari keluaran pengendali MPC SISO. Bagan diagramnya dapat dilihat pada gambar 3.3.

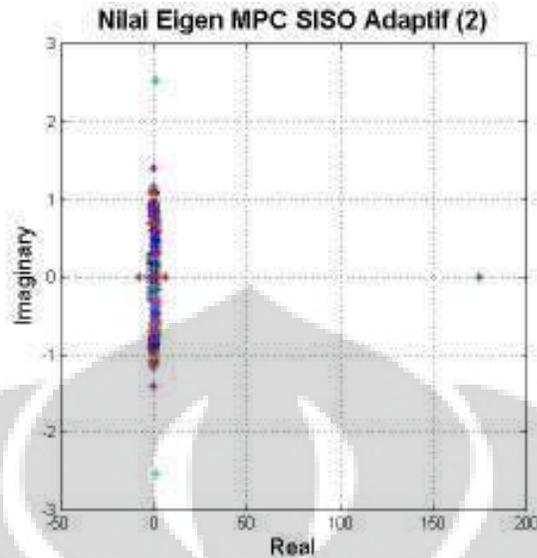
Hasil simulasi dari pengendali MPC SISO Adaptif yang telah dirancang dapat dilihat dari grafik nilai eigennya, grafik keluaran sistem dan grafik sinyal kendalinya. Gambar 4.16 menunjukkan grafik nilai egen hasil pengendali MPC SISO Adaptif untuk batasan sinyal kendali yang berbeda.



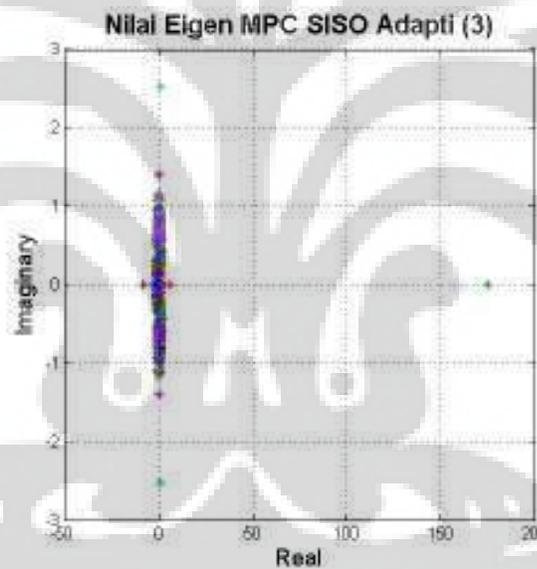
(a)



(b)



©



(d)

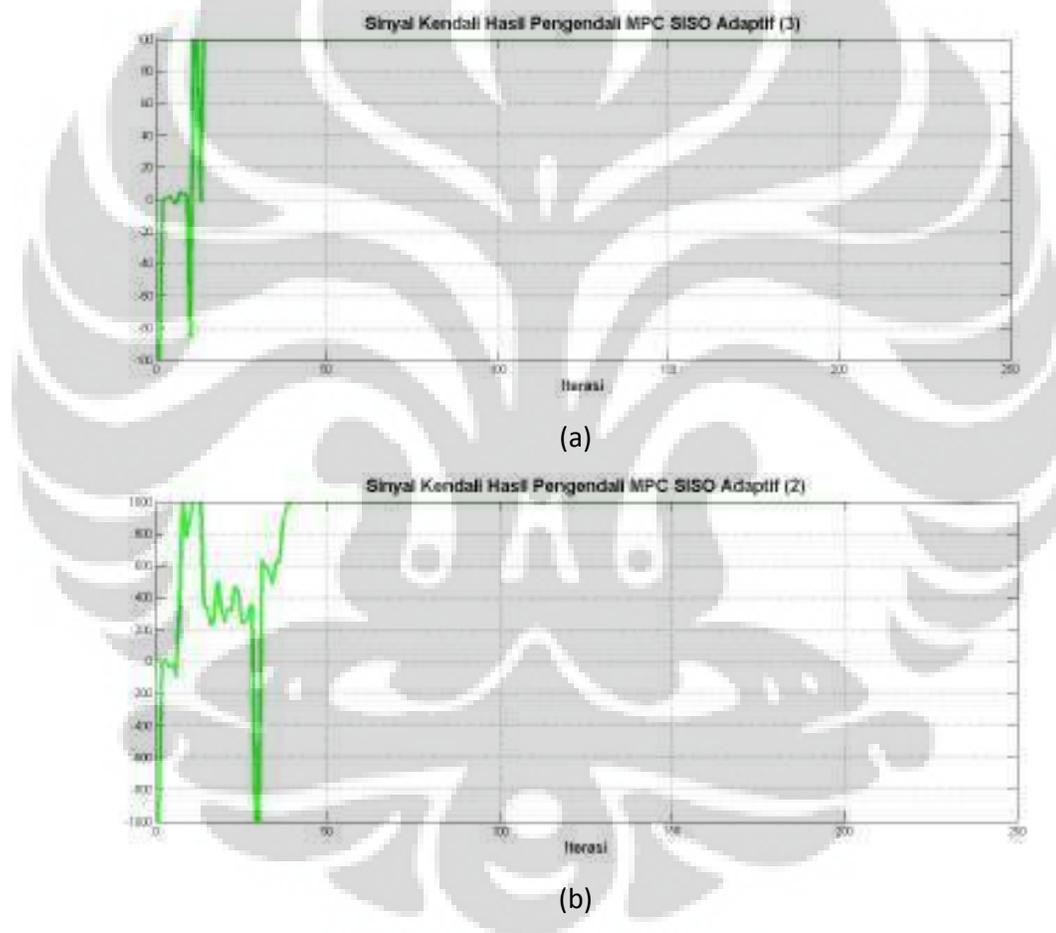
Gambar 4.16(a). Nilai eigen hasil pengendali MPC SISO Adaptif untuk batasan *constraint* $u_{min} = -10$ dan $u_{max} = 10$; (b). $u_{min} = -1000$ dan $u_{max} = 1000$; (c). $u_{min} = -10000$ dan $u_{max} = 10000$; (d). $u_{min} = -100000$ dan $u_{max} = 100000$

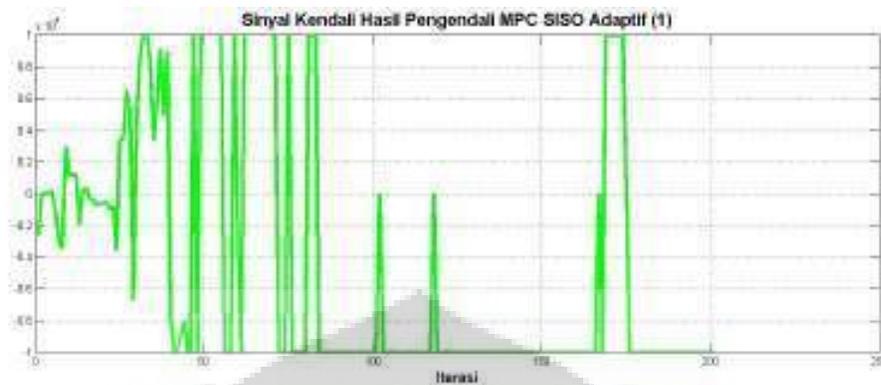
Dari graik nilai eigen pada gambar 4.17 dapat disimpulkan bahwa perubahan sinyal kendali yang generasi oleh model pengendali menentukan nilai eigen untuk iterasi berikutnya. Perubahan nilai eigen menandakan perubahan nilai model yang digunakan untuk iterasi berikutnya. Sehingga, batasan *constraint* yang

diberikan mempengaruhi model yang dikeluarkan oleh pengendali MPC SISO dengan *constraint*.

Sepertinya halnya telah diuji pada pengendali MPC SISO dengan identifikasi *offline* dimana sinyal keluaran sistem mampu mengikuti sinyal referensi yang diberikan pada saat batasan sinyal kendalinya dibuat sangat besar. Hal yang sama juga berlaku pada pengendali MPC SISO Adaptif ini, dimana batas sinyal kendali maximum adalah $u_{min} = -1000$ dan $u_{max} = 1000$.

Grafik sinyal kendali hasil pengendali MPC SISO Adaptif ini dapat dilihat pada gambar 4.17.

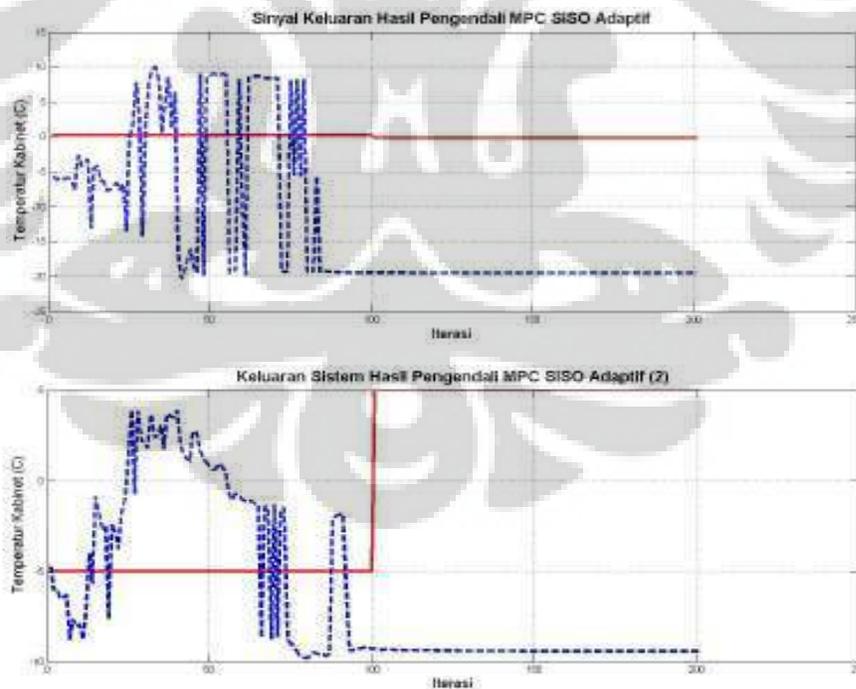




(c)

Gambar 4. 17(a). Sinyal kendali hasil pengendali MPC SISO adaptif untuk batas $constraint$ $u_{min} = -10$ dan $u_{max} = 10$; (b). $u_{min} = -1000$ dan $u_{max} = 1000$; (c). $u_{min} = -10000$ dan $u_{max} = 10000$; (d). $u_{min} = -100000$ dan $u_{max} = 100000$

Gambar 4.18 merupakan penyebab berubahnya nilai eigen yang dihasilkan untuk iterasi berikutnya. Karena perubahan sinyal kendlai yang sangat besar, maka model yang dihasilkan menjadi tidak bagus. Hal ini tentunya juga mempengaruhi sinyal keluaran sistem hasil pengendali MPC SISO Adaptif yang dirancang, yang ditunjukkan pada gambar 4.19 dibawah.



Gambar 4.18. Sinyal keluaran sistem hasil pengendali MPC SISO adaptif untuk sinyal referensi yang berbeda

Gambar 4.18 menunjukkan bahwa teori yang ada telah terpenuhi dimana perubahan sinyal referensi akan mempengaruhi sinyal keluaran sistem yang akan dihasilkan oleh pengendali MPC.



BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari skripsi ini, yaitu:

1. Hasil pengendali MPC sistem SISO dengan *constraint* pada sistem tata udara presisi sudah terlihat bagus karena sinyal keluaran sistem sudah stabil dan bisa mengikuti sinyal referensi yang diberikan.
2. Keluaran temperatur kabinet sudah berada pada interval suhu yang diinginkan yaitu pada batas $20^{\circ}\text{C} - 22^{\circ}\text{C}$.
3. Nilai parameter pengendali terbaik untuk MPC SISO dengan *constraint* untuk identifikasi *offline* pada sistem tata udara presisi adalah $H_p = 5$, $H_u = 2$, $Q = 1000I_{Hp}$ dan $R = 1000I_{Hu}$.
4. Hasil simulasi pengendali MPC SISO Adaptif yang dirancang masih belum menghasilkan keluaran seperti yang diinginkan.

5.2 Saran

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

1. Mendesain model pengendali MPC untuk sistem MIMO (*multi input multi output*) agar bisa merepresentasikan konsep PAC secara keseluruhan.
2. Identifikasi sistem harus lebih baik agar nilai A, B, C dan D yang dihasilkan bisa cocok dengan model pengendali MPC yang dirancang.
3. Identifikasi rekursif harusnya membuat keluaran sistem menjadi lebih baik untuk setiap iterasinya.

DAFTAR REFERENSI

- [1] Subiantoro, Aries. 2007. *Diktat kuliah sistem kendali adaptif dan nonlinier*. Depok: Departemen Teknik Elektro, Fakultas Teknik, Universitas Indonesia
- [2] J. M. Maciejowski. 2002. *Predictive Control with Constraints*. Prentice Hall
- [3] Surayuda, Rise. 2010. *Perancangan Pengendali Model Predictive Control Dengan Constraint Pada Sistem Tata Udara Presisi*. Depok: Departemen Elektro, Fakultas Teknik, Universitas Indonesia
- [4] Kristiaawan, Antonius Yuda. 2008. *Perancangan dan Implementasi Pengendali Model Predictive Control dengan Constraints pada Coupled-tank Control Apparatuspp-100*. Depok: Departemen Elektro, Fakultas Teknik, Universitas Indonesia
- [5] Victor. 2011. *Identifikasi Model Ruang Keadaan Dengan Vektor Pengkoreksi Pada Sistem Tata Udara Presisi Menggunakan Algoritma N4SID*. Depok: Departemen Elektro, Fakultas Teknik, Universitas Indonesia
- [6] J. -S. Kim. "Recent Advances in Adaptive MPC." In International Conference on Control, Automation and Systems 2010, Oct. 27-30, 2010 in KINTEX, Gyeonggi-do, Korea, pp. 218 - 222
- [7] Y. Zhu. "A New Adaptive MPC System." In Proceedings of the 2011 4th International Symposium on Advanced Control of Industrial Processes, Thousand Islands Lake, Hangzhou, P.R. China, May 23-26, 2011, pp. 447 - 449