

**PERANCANGAN DAN IMPLEMENTASI  
PENGENDALI  
*MODEL PREDICTIVE CONTROL* DENGAN  
*CONSTRAINT* UNTUK PENGATURAN LEVEL PADA  
*COUPLED-TANK BASIC PROCESS RIG 38-100***

**SKRIPSI**

Oleh :

**JESSE MELVIN**

**04 03 03 05 47**



**DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNIK UNIVERSITAS INDONESIA  
GENAP 2007/2008**

**PERANCANGAN DAN IMPLEMENTASI  
PENGENDALI  
*MODEL PREDICTIVE CONTROL DENGAN CONSTRAINT*  
UNTUK PENGATURAN LEVEL PADA *COUPLED-TANK*  
*BASIC PROCESS RIG 38-100***

**SKRIPSI**

Oleh :

**JESSE MELVIN**

**04 03 03 05 47**



**SKRIPSI INI DIAJUKAN UNTUK MELENGKAPI SEBAGIAN  
PERSYARATAN MENJADI SARJANA TEKNIK**

**DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNIK UNIVERSITAS INDONESIA  
GENAP 2007/2008**

# **PERNYATAAN KEASLIAN SKRIPSI**

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul :

**PERANCANGAN DAN IMPLEMENTASI PENGENDALI  
MODEL PREDICTIVE CONTROL DENGAN CONSTRAINT UNTUK  
PENGATURAN LEVEL PADA COUPLED-TANK  
BASIC PROCESS RIG 38-100**

yang dibuat untuk melengkapi sebagian prasyarat menjadi Sarjana Teknik pada Program Studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, sejauh yang saya ketahui bukan merupakan tiruan atau duplikasi dari skripsi yang sudah dipublikasikan dan atau pernah dipakai untuk mendapatkan gelar kesarjanaan di lingkungan Universitas Indonesia maupun di Perguruan Tinggi atau Instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Depok, Juni 2008

**Jesse Melvin**

**NPM 04 04 03 05 47**

# PENGESAHAN

Skripsi dengan judul :

**PERANCANGAN DAN IMPLEMENTASI PENGENDALI  
MODEL PREDICTIVE CONTROL DENGAN CONSTRAINT UNTUK  
PENGATURAN LEVEL PADA COUPLED-TANK  
BASIC PROCESS RIG 38-100**

dibuat untuk melengkapi sebagian prasyarat menjadi Sarjana Teknik pada Program Studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia. Skripsi ini telah diujikan pada sidang ujian skripsi pada Juli 2008 dan dinyatakan memenuhi syarat/sah sebagai skripsi pada Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia.

Depok, Juni 2008

Dosen Pembimbing

**Ir. Aries Subiantoro, M.Sc.**

**NIP 132 137 887**

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada :

**Ir. Aries Subiantoro, M.Sc.**

selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberi pengarahan, diskusi dan bimbingan serta persetujuan sehingga skripsi ini dapat selesai dengan baik.



Jesse Melvin  
NPM 04 04 03 05 47  
Departemen Teknik Elektro

Dosen Pembimbing  
Ir. Aries Subiantoro, M.Sc,

**PERANCANGAN DAN IMPLEMENTASI PENGENDALI  
MODEL PREDICTIVE CONTROL DENGAN CONSTRAINT UNTUK  
PENGATURAN LEVEL PADA COUPLED-TANK  
BASIC PROCESS RIG 38-100**

**ABSTRAK**

Pada sistem kendali konvensional, batasan-batasan seperti amplitudo dan *slew rate* sinyal kendali tidak diperhitungkan pada proses pengendalian. Hal ini tentu dapat menyebabkan hasil kendali menjadi kurang baik, terutama jika terjadi pemotongan paksa terhadap sinyal kendali sebelum masuk ke *plant*. Untuk mengatasi hal tersebut dirancanglah suatu pengendali MPC. Dengan MPC, keluaran proses yang akan datang dapat diprediksi dan batasan-batasan yang ada tidak diabaikan sehingga keluaran sistem menjadi bagus. Selain keluaran sistem menjadi bagus, adanya batasan juga dapat membuat kinerja alat menjadi optimal.

Pada skripsi ini, sistem yang akan dikendalikan dengan metode MPC dengan *constraints* adalah *Coupled-Tank Basic Process Rig 38-100*. Model yang digunakan pada perancangan pengendali berbentuk ruang keadaan yang didapat dengan menggunakan metode Kuadrat Terkecil berdasarkan pada data masukan dan data variabel keadaan alat. Masukan sistem adalah tegangan pompa pada tangki pertama dan keluaran yang akan dikendalikan adalah ketinggian air pada tangki kedua.

Dari uji eksperimen terbukti bahwa metode pengendali MPC dengan *constraints* memberikan hasil yang lebih baik dibandingkan dengan metode Aturan Kendali Ruang Keadaan. Hal tersebut dapat terlihat dari tanggapan sistem, dimana tanggapan sistem dengan menggunakan metode MPC lebih cepat serta tidak adanya *overshoot* maupun *undershoot* pada keluaran sistem saat terjadi perubahan nilai trayektori acuan.

**Kata Kunci : batasan, pengendali, masukan, keluaran, model.**

Jesse Melvin  
NPM 04 04 03 05 47  
Electrical Engineering Department

Counsellor  
Ir. Aries Subiantoro, M.Sc.

**THE DESIGNING AND IMPLEMENTING OF MODEL PREDICTIVE  
CONTROLLER WITH CONSTRAINT FOR COUPLED TANK  
BASIC PROCESS RIG 38-100 LEVEL CONTROL**

**ABSTRACT**

In conventional control system, constraints, such as amplitude and slew rate of input signal are not computed in control process. This matter of course can make the control result become worst, especially when force cutting occur to input signal before it enters to the plant. To solve those problems, a MPC controller is designed. With MPC, process output can be predicted and the existence of constraints will not be ignored and, as the result, it makes output system become well. Besides improve output system quality, the existence of the constraints can also make the device works at optimum condition everytime.

In this following final thesis, system that will be controlled by MPC with constraints method is Coupled-Tank Basic Process Rig 38-100. Model that is used in controller design has state space form. This model is formed by using Least Squares method based on input and state variable data. Input system is pump in first tank and output that will be controlled is water level in second tank.

Experiments prove that MPC with constraints give better result than State Controller method. With MPC, system response become faster and there are no overshoot nor undershoot when the set point change.

**Key Words : constraints, controller, input, output, model.**

# DAFTAR ISI

PERNYATAAN KEASLIAN SKRIPSI.....	ii
PENGESAHAN .....	iii
UCAPAN TERIMA KASIH.....	iv
ABSTRAK .....	v
ABSTRACT .....	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR .....	x
BAB 1 PENDAHULUAN .....	1
1.1 LATAR BELAKANG .....	1
1.2 TUJUAN .....	2
1.3 PEMBATAAN MASALAH.....	2
1.4 SISTEMATIKA PENULISAN .....	2
BAB 2 LANDASAN TEORI.....	3
2.1 IDENTIFIKASI SISTEM .....	3
2.2 <i>MODEL PREDICTIVE CONTROL (MPC)</i> .....	5
2.2.1 Konsep Dasar <i>Model Predictive Control</i> .....	5
2.2.2 Fungsi Kriteria pada <i>Model Predictive Control</i> .....	7
2.2.3 Model Proses .....	8
2.2.4 Prediksi .....	9
2.2.5 Strategi Pengendali <i>Model Predictive Control</i> tanpa <i>Constraints</i> .....	12
2.2.6 Strategi Pengendali <i>Model Predictive Control</i> dengan <i>Constraints</i> .....	14
2.2.6.1 Pembentukan <i>Constraints</i> .....	14
2.2.6.2 Metode <i>Quadratic Programming</i> .....	16
2.3 <i>REDUCED-ORDER STATE OBSERVER</i> .....	18
2.3.1 Pembentukan Persamaan <i>State</i> dan Persamaan Keluaran.....	19
2.3.2 Pembentukan Persamaan Dinamik <i>Reduced-Order</i> <i>Observer</i> .....	20
BAB 3 PERANCANGAN SISTEM .....	23

3.1	DESKRIPSI PROSES .....	23
3.1.1	Sistem Dua Tangki Berhubungan .....	23
3.1.2	Kalibrasi Komponen .....	25
3.1.2.1	Kalibrasi <i>Servo Valve</i> .....	25
3.1.2.2	Kalibrasi <i>Sensor tipe Potensiometer</i> .....	25
3.1.3	Interkoneksi Alat .....	25
3.2	PEMBUATAN MODEL <i>COUPLED-TANK BASIC PROCESS RIG 38-100</i> .....	26
3.2.1	Penentuan Daerah Kerja .....	26
3.2.2	Pencarian Model Parametrik .....	28
3.2.3	Perancangan <i>Reduced-Order State Observer</i> .....	29
3.2.3.1	Pengetesan <i>Observability</i> Sistem .....	29
3.2.3.2	Pembentukan Persamaan Karakteristik <i>Observer</i> .....	30
3.2.4	Identifikasi Model Proses .....	31
3.3	ALGORITMA <i>MODEL PREDICTIVE CONTROL DENGAN CONSTRAINTS</i> .....	37
3.4	PERHITUNGAN SINYAL KENDALI .....	42
BAB 4	UJI EKSPERIMEN DAN ANALISA .....	53
4.1	PENGARUH NILAI <i>CONTROL HORIZON</i> PADA HASIL PENGENDALIAN <i>MPC WITH CONSTRAINT</i> .....	53
4.2	PENGARUH NILAI <i>PREDICTION HORIZON</i> PADA HASIL PENGENDALIAN <i>MPC WITH CONSTRAINT</i> .....	62
4.3	PENGARUH NILAI FAKTOR BOBOT PERUBAHAN SINYAL KENDALI ( <i>R</i> ) PADA HASIL PENGENDALIAN <i>MPC WITH CONSTRAINT</i> .....	71
4.4	PENGARUH NILAI FAKTOR BOBOT KESALAHAN ( <i>Q</i> ) PADA HASIL PENGENDALIAN <i>MPC WITH CONSTRAINT</i> .....	79
4.5	UJI EKSPERIMEN PENGENDALI <i>MPC WITH CONSTRAINT</i> TANPA NILAI TRAYEKTORI ACUAN YANG AKAN DATANG .....	88

4.6 PERBANDINGAN KINERJA PENGENDALI METODE MPC <i>WITH CONSTRAINT</i> DAN METODE ATURAN KENDALI RUANG KEADAAN.....	92
4.6.1 Landasan Teori Aturan Kendali Ruang Keadaan .....	92
4.6.2 Uji Eksperimen dan Analisa .....	94
4.6.2.1 Pengetesan <i>Controllability</i> dan Perancangan Pengendali.....	94
4.6.2.2 Uji Eksperimen Pengendali Ruang Keadaan.....	95
BAB 5 KESIMPULAN.....	98
DAFTAR ACUAN .....	99
DAFTAR PUSTAKA .....	100
LAMPIRAN.....	101



## DAFTAR GAMBAR

<b>Gambar 2.1.</b>	Struktur pengendali MPC.....	7
<b>Gambar 2.2.</b>	Kalkulasi keluaran proses dan pengendali terprediksi.....	7
<b>Gambar 2.3.</b>	Skematik <i>Observed-State Feedback Control System</i> .....	18
<b>Gambar 2.4.</b>	Skematik <i>reduced-order observer</i> dengan <i>state feedback control system</i> .....	22
<b>Gambar 3.1.</b>	(a) Bentuk fisik <i>Coupled-Tank Basic Process Rig 38-100</i> . (b) Sketsa <i>Coupled-Tank Basic Process Rig 38-100</i> . ....	23
<b>Gambar 3.2.</b>	Sketsa <i>Process Interface 38-200</i> .....	24
<b>Gambar 3.3.</b>	Sketsa gabungan <i>Basic Process Rig 38-100</i> dan <i>Process Interface</i> .....	25
<b>Gambar 3.4.</b>	Skema interkoneksi antara <i>Coupled-Tank Basic Process Rig 38-100</i> dengan sebuah PC.....	26
<b>Gambar 3.5.</b>	Grafik hubungan keluaran tunak terhadap masukan <i>Unit Step</i> dengan amplitudo yang berbeda-beda.....	27
<b>Gambar 3.6.</b>	Blok SIMULINK yang digunakan untuk mengambil data masukan dan keluaran dari <i>Coupled-Tank Basic Process Rig 38-100</i> . ....	31
<b>Gambar 3.7.</b>	(a) Grafik sinyal masukan saat identifikasi. (b) Grafik dari sensor ketinggian pada tangki pertama. (c) Grafik dari sensor ketinggian pada tangki kedua.....	34
<b>Gambar 3.8.</b>	Grafik keluaran proses dan keluaran model pada tangki kedua..	36
<b>Gambar 3.9.</b>	Selisih antara keluaran proses dengan keluaran model pada tangki kedua .....	36
<b>Gambar 3.10.</b>	Blok diagram pengendali MPC <i>with constraints</i> . ....	37
<b>Gambar 3.11.</b>	Diagram alir algoritma MPC <i>with constraints</i> . ....	39
<b>Gambar 3.12.</b>	Diagram alir metode <i>Active Set</i> untuk menyelesaikan <i>Quadratic Programming</i> . ....	42
<b>Gambar 4.1.</b>	Keluaran sistem hasil uji simulasi dengan nilai <i>control horizon</i> yang berbeda. ....	55

<b>Gambar 4.2.</b>	Sinyal kendali hasil uji simulasi dengan nilai <i>control horizon</i> yang berbeda .....	56
<b>Gambar 4.3.</b>	Estimasi masukan hasil uji simulasi dengan nilai <i>control horizon</i> yang berbeda.....	58
<b>Gambar 4.4.</b>	Keluaran sistem hasil uji eksperimen dengan nilai <i>control horizon</i> yang berbeda. ....	60
<b>Gambar 4.5.</b>	Sinyal masukan hasil uji eksperimen dengan nilai <i>control horizon</i> yang berbeda .....	61
<b>Gambar 4.6.</b>	Estimasi masukan hasil uji eksperimen dengan nilai <i>control horizon</i> yang berbeda.....	62
<b>Gambar 4.7.</b>	Keluaran sistem hasil uji simulasi dengan nilai <i>prediction horizon</i> yang berbeda. ....	64
<b>Gambar 4.8.</b>	Sinyal masukan hasil uji simulasi dengan nilai <i>prediction horizon</i> yang berbeda. ....	65
<b>Gambar 4.9.</b>	Estimasi masukan hasil uji simulasi dengan nilai <i>prediction horizon</i> yang berbeda.....	67
<b>Gambar 4.10.</b>	Keluaran sistem hasil uji eksperimen dengan nilai <i>prediction horizon</i> yang berbeda.....	68
<b>Gambar 4.11.</b>	Sinyal kendali hasil uji eksperimen dengan nilai <i>prediction horizon</i> yang berbeda.....	69
<b>Gambar 4.12.</b>	Estimasi masukan hasil uji eksperimen dengan nilai <i>prediction horizon</i> yang berbeda.....	70
<b>Gambar 4.13.</b>	Keluaran sistem hasil uji simulasi untuk nilai matriks $R$ yang berbeda-beda.....	72
<b>Gambar 4.14.</b>	Sinyal masukan hasil uji simulasi untuk nilai matriks $R$ yang berbeda-beda. ....	74
<b>Gambar 4.15.</b>	Estimasi masukan hasil uji simulasi untuk nilai matriks $\underline{R}$ yang berbeda-beda.....	75
<b>Gambar 4.16.</b>	Keluaran sistem hasil uji eksperimen untuk nilai matriks $R$ yang berbeda-beda. ....	77
<b>Gambar 4.17.</b>	Sinyal kendali hasil uji eksperimen untuk nilai matriks $R$ yang berbeda-beda. ....	78

<b>Gambar 4.18.</b> Estimasi masukan hasil uji eksperimen untuk nilai matriks $\underline{R}$ yang berbeda-beda.....	79
<b>Gambar 4.19.</b> Keluaran sistem hasil uji simulasi untuk nilai matriks $\underline{Q}$ yang berbeda-beda.....	81
<b>Gambar 4.20.</b> Sinyal masukan hasil uji simulasi untuk nilai matriks $\underline{Q}$ yang berbeda-beda. ....	82
<b>Gambar 4.21.</b> Estimasi masukan hasil uji simulasi untuk nilai matriks $\underline{Q}$ yang berbeda-beda.....	84
<b>Gambar 4.22.</b> Keluaran sistem hasil uji eksperimen untuk nilai matriks $\underline{Q}$ yang berbeda-beda. ....	85
<b>Gambar 4.23.</b> Sinyal kendali hasil uji eksperimen untuk nilai matriks $\underline{Q}$ yang berbeda-beda. ....	86
<b>Gambar 4.24.</b> Estimasi masukan hasil uji eksperimen untuk nilai matriks $\underline{R}$ yang berbeda-beda.....	87
<b>Gambar 4.25.</b> Keluaran sistem hasil uji eksperimen dimana sistem tidak mengetahui nilai trayektori acuan untuk masa yang akan datang. ....	89
<b>Gambar 4.26.</b> Sinyal kendali hasil uji eksperimen dimana sistem tidak mengetahui nilai trayektori acuan untuk masa yang akan datang. ....	90
<b>Gambar 4.27.</b> Estimasi masukan hasil uji eksperimen dimana sistem tidak mengetahui nilai trayektori acuan untuk masa yang akan datang.....	91
<b>Gambar 4.28.</b> Pengendali ruang keadaan lingkaran tertutup.....	92
<b>Gambar 4.29.</b> Pengendali ruang keadaan lingkaran tertutup dengan penguat <i>precompensator</i> .....	93
<b>Gambar 4.30.</b> Hasil keluaran uji eksperimen pengendali ruang keadaan dengan dua perubahan trayektori acuan.....	95
<b>Gambar 4.31.</b> Sinyal kendali uji eksperimen pengendali ruang keadaan dengan dua perubahan trayektori acuan.....	96

# PENDAHULUAN

## 1.1 LATAR BELAKANG

*Model Predictive Control* (MPC) merupakan suatu metodologi pengendalian yang saat ini memiliki pengaruh yang sangat penting dalam bidang industri dibandingkan dengan pengendali konvensional seperti *Two-Degree of Freedom* ataupun Aturan Kendali Ruang Keadaan.

Pada sistem kendali konvensional, batasan-batasan (*constraints*) seperti amplitudo dan *slew rate* sinyal kendali tidak diperhitungkan pada proses pengendalian. Hal ini tentu dapat menyebabkan hasil kendali menjadi kurang baik, terutama jika terjadi pemotongan paksa terhadap sinyal kendali sebelum masuk ke *plant*. Pemotongan sinyal kendali biasanya terjadi ketika nilai trayektori acuan berubah secara mendadak. Hal tersebut tentu tidak akan terjadi pada MPC karena pengendali dapat memprediksi keluaran proses yang akan datang serta tidak mengabaikan batasan-batasan yang ada. Selain agar keluaran sistem menjadi bagus, adanya batasan pada proses pengendali dapat membuat kinerja alat menjadi optimal sehingga alat tidak cepat rusak dan dapat beroperasi dalam jangka waktu yang lama.

Banyaknya faktor yang harus diperhitungkan pada pengendali MPC membuat algoritma MPC menjadi sangat panjang dan rumit. Akan tetapi dengan kecepatan komputasi perangkat keras saat ini, tidak lagi menjadi masalah utama. Masalah utama metode MPC adalah keperluan akan model proses. Model proses pada MPC berguna untuk memprediksi keluaran sistem sehingga pengendali MPC dapat memberikan sinyal masukan yang sesuai. Oleh sebab itu, algoritma MPC membutuhkan model proses yang baik.

Skripsi ini mencoba menerapkan algoritma MPC dengan batasan (*constraint*) pada *Coupled-Tank Basic Process Rig 38-100* dan hasilnya akan dibandingkan dengan hasil metode Aturan Kendali Ruang Keadaan.

## 1.2 TUJUAN

Skripsi ini bertujuan untuk merancang dan implementasi suatu pengendali *Model Predictive Control with constraints* pada sistem tangki terhubung dengan menggunakan model ruang keadaan linier serta melihat pengaruh parameter-parameter pengendali pada hasil pengendalian. Selain itu, skripsi ini juga bertujuan untuk membandingkan kinerja antara metode *Model Predictive Control with constraints* dan metode Aturan Kendali Ruang Keadaan yang dirancang dengan menggunakan bantuan fasilitas *s-function program MatLab 7.0*.

## 1.3 PEMBATASAN MASALAH

Skripsi ini membahas perancangan MPC dengan *constraints* menggunakan metode *Quadratic Programming* dalam menghitung besar perubahan sinyal kendali. *Constraints* yang digunakan adalah amplitudo dan *slew rate* sinyal kendali.

Model yang digunakan pada skripsi ini adalah model ruang keadaan linier dengan pompa pada tangki pertama sebagai masukan dan ketinggian air pada tangki kedua sebagai keluaran sistem dengan cara identifikasi.

## 1.4 SISTEMATIKA PENULISAN

Skripsi ini terbagi dalam lima bab, yang masing-masing memiliki pokok bahasan tertentu sebagai bagian dari tujuan pembahasan skripsi.

Bab satu merupakan pendahuluan yang berisi latar belakang, tujuan, pembatasan masalah dan sistematika penulisan skripsi. Bab dua membahas mengenai landasan teori yang menyangkut tentang identifikasi model dan konsep dasar metode MPC *with constraints*. Bab tiga merupakan pembahasan tentang penjelasan proses dan perancangan pengendali dengan metode MPC *with constraints*. Bab empat berisi simulasi, uji coba, dan analisa hasil pengendali dengan metode MPC *with constraints*. Selain itu, pada bab ini juga sedikit dibahas mengenai metode Aturan Kendali Ruang Keadaan dan hasil pengendaliannya jika dibandingkan dengan metode MPC *with constraints*. Bab lima merupakan kesimpulan dari keseluruhan pembahasan dalam skripsi ini.

## BAB 2

### LANDASAN TEORI

#### 2.1 IDENTIFIKASI SISTEM

Pada skripsi ini, model proses ditentukan berdasarkan data masukan dan keluaran dengan menggunakan metode Kuadrat Terkecil. Inti dari metode Kuadrat Terkecil adalah bahwa kecocokan antara model dengan sistem yang akan diidentifikasi diperoleh dengan meminimumkan selisih kuadrat antara keluaran model dengan keluaran sistem yang diidentifikasi untuk semua  $N$  data pengamatan [1]. Selisih kuadrat antara keluaran model dan keluaran sistem dapat dinyatakan dalam fungsi kriteria berikut

$$J_{LS} = \sum_{i=1}^N \varepsilon_i^2 = \sum_{i=1}^N (y(i) - \hat{y}(i))^2 \quad (2.1)$$

dengan :

$J_{LS}$  = fungsi kriteria

$\varepsilon_i$  = kesalahan prediksi data ke- $i$

$y(i)$  = data keluaran ke- $i$

$\hat{y}(i)$  = prediksi keluaran ke- $i$

Fungsi kriteria pada persamaan (2.1) disebut juga sebagai *loss function*.

Keluaran model untuk satu langkah prediksi kedepan dari model dinamik orde- $n$  adalah sebagai berikut

$$\hat{y}(k) = -a_1 y(k-1) - \dots - a_n y(k-n) + b_1 u(k-1) + \dots + b_n u(k-n) \quad (2.2)$$

Persamaan (2.2) kemudian dapat ditulis ke dalam bentuk vektor matriks sebagai berikut

$$\hat{y}(k) = \underbrace{\begin{bmatrix} -y(k-1) & \cdots & -y(k-n) & u(k-1) & \cdots & u(k-n) \end{bmatrix}}_{\underline{\rho}^T} \underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_1 \\ \vdots \\ b_n \end{bmatrix}}_{\underline{\hat{\theta}}} \quad (2.3)$$

Dengan mensubstitusikan persamaan (2.3) ke persamaan (2.1), maka persamaan *loss function*  $J_{LS}$  menjadi

$$J_{LS} = \sum_{i=1}^N \left( y(i) - \underline{\rho}^T(i) \underline{\hat{\theta}} \right)^2 \quad (2.4)$$

Untuk sejumlah  $N$  data, persamaan (2.3) dapat ditulis kembali dalam bentuk matriks menjadi

$$\underbrace{\begin{bmatrix} \hat{y}(1) \\ \hat{y}(2) \\ \vdots \\ \hat{y}(N) \end{bmatrix}}_{\underline{\hat{y}}} = \underbrace{\begin{bmatrix} -y(0) & \cdots & -y(-n) & u(0) & \cdots & u(-n) \\ -y(1) & \cdots & -y(1-n) & u(1) & \cdots & u(1-n) \\ \vdots & & \vdots & \vdots & & \vdots \\ -y(N-1) & \cdots & -y(N-n) & u(N-1) & \cdots & u(N-n) \end{bmatrix}}_{\underline{P}} \underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_1 \\ \vdots \\ b_n \end{bmatrix}}_{\underline{\hat{\theta}}} \quad (2.5)$$

atau

$$\underline{\hat{y}} = \underline{P} \underline{\hat{\theta}} \quad (2.6)$$

Supaya persamaan (2.4) dapat diminimasi, maka persamaan (2.4) harus dinyatakan dalam bentuk

$$\begin{aligned} J_{LS} &= (\underline{y} - \underline{P} \underline{\hat{\theta}})^T (\underline{y} - \underline{P} \underline{\hat{\theta}}) \\ &= \underline{y}^T \underline{y} - \underline{\hat{\theta}}^T \underline{P}^T \underline{y} - \underline{y}^T \underline{P} \underline{\hat{\theta}} + \underline{\hat{\theta}}^T \underline{P}^T \underline{P} \underline{\hat{\theta}} \end{aligned} \quad (2.7)$$

Selanjutnya dengan membuat turunan pertama  $J_{LS}(\underline{\theta})$  terhadap  $\underline{\hat{\theta}}$  menjadi nol :

$$\left. \frac{\partial J_{LS}(\theta)}{\partial \theta} \right|_{\theta=\hat{\theta}} = -2\underline{P}^T \underline{y} + 2\underline{P}^T \underline{P} \hat{\theta} = 0$$

maka didapatkan rumus untuk menghitung parameter estimasi  $\hat{\theta}$  sebagai berikut

$$\hat{\theta} = (\underline{P}^T \underline{P})^{-1} \underline{P}^T \underline{y} \quad (2.8)$$

## 2.2 MODEL PREDICTIVE CONTROL (MPC)

### 2.2.1 Konsep Dasar Model Predictive Control

*Model Predictive Control* (MPC) atau sistem kendali prediktif termasuk dalam konsep perancangan pengendali berbasis model proses, dimana model proses digunakan secara eksplisit untuk merancang pengendali dengan cara meminimumkan suatu fungsi kriteria. Ide yang mendasari pada setiap jenis MPC adalah [2] :

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

Metode MPC memiliki beberapa keunggulan dibandingkan dengan metode pengendali lainnya, di antaranya adalah :

1. Konsepnya sangat intuitif serta penalaannya mudah.
2. Dapat digunakan untuk mengendalikan proses yang beragam, mulai dari proses yang sederhana, hingga proses yang kompleks, memiliki waktu tunda yang besar, *non-minimum phase* atau proses yang tidak stabil.
3. Dapat menangani sistem multivariabel.
4. Mempunyai kompensasi terhadap waktu tunda.

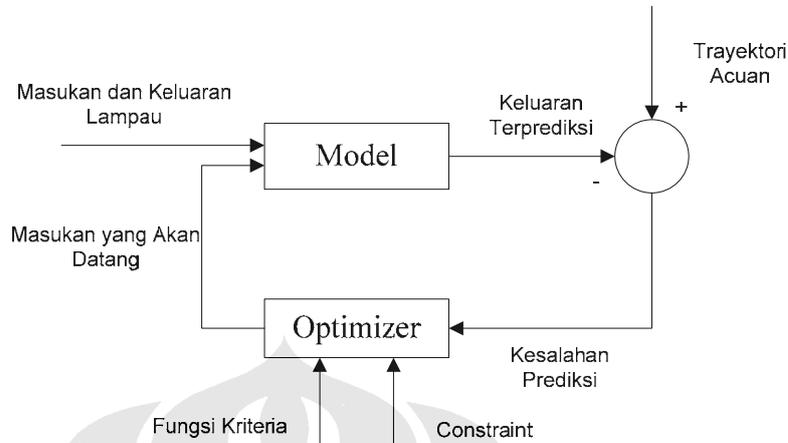
5. Mempunyai kemampuan dari pengendali *feed forward* untuk mengkompensasi gangguan yang terukur.
6. Mudah untuk mengimplementasikan pengendali yang diperoleh.
7. Dapat memperhitungkan batasan atau *constraint* dalam merancang pengendali.
8. Sangat berguna jika sinyal acuan untuk masa yang akan datang diketahui.

Selain beragam keuntungan yang dimiliki, metode MPC juga mempunyai kelemahan, yaitu masalah penurunan aturan sinyal kendali yang cukup kompleks dan keperluan akan model proses yang baik.

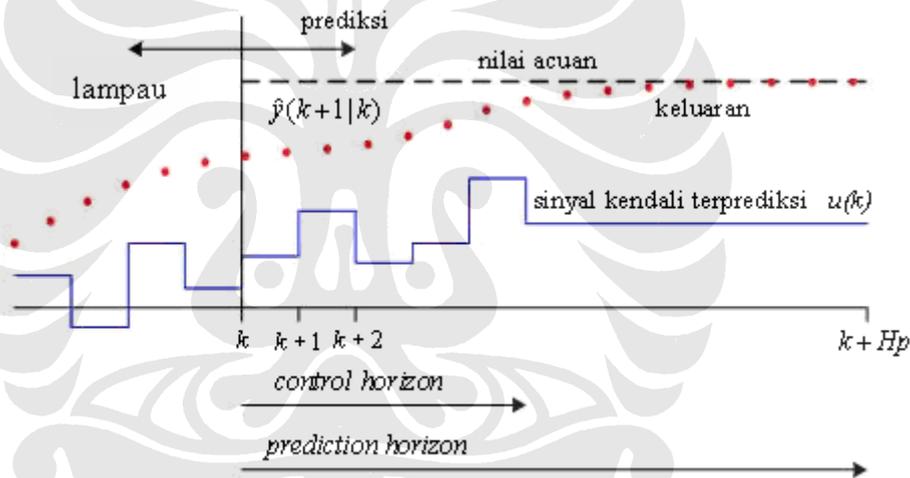
Struktur dasar dari pengendali MPC dapat dilihat pada gambar 2.1. Metodologi semua jenis pengendali yang termasuk kedalam kategori MPC dapat dikenali oleh strategi berikut [1] :

1. Keluaran proses yang akan datang untuk rentang *horizon Hp* yang ditentukan yang dinamakan sebagai *prediction horizon*, diprediksi pada setiap waktu pencuplikan dengan menggunakan model proses. Keluaran proses terprediksi ini  $y(k+i/k)$  untuk  $i = 1 \dots Hp$  bergantung pada nilai masukan dan keluaran lampau dan kepada sinyal kendali yang akan datang  $u(k+i/k)$ ,  $i = 0 \dots Hp-1$ , yang akan digunakan sistem dan harus dihitung.
2. Serangkaian sinyal kendali dihitung dengan mengoptimasi suatu fungsi kriteria yang ditetapkan sebelumnya, dengan tujuan untuk menjaga proses sedekat mungkin terhadap trayektori acuan  $r(k+i)$ . Fungsi kriteria tersebut umumnya berupa suatu fungsi kuadratik dari kesalahan antara sinyal keluaran terprediksi dengan trayektori acuan. 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 dapat diilustrasikan pada gambar 2.2.
3. Sinyal kendali  $u(k/k)$  dikirim ke proses, sedangkan sinyal kendali terprediksi berikutnya dibuang, karena pada pencuplikan berikutnya  $y(k+1)$  sudah diketahui nilainya. Maka langkah pertama diulang dengan nilai keluaran proses yang baru dan semua prosedur perhitungan yang diperlukan

diperbaiki. Sinyal kendali yang baru  $u(k+1/k+1)$  (nilainya berbeda dengan  $u(k+1/k)$ ) dihitung dengan menggunakan konsep *receding horizon*.



Gambar 2.1. Struktur pengendali MPC



Gambar 2.2. Kalkulasi keluaran proses dan pengendali terprediksi

### 2.2.2 Fungsi Kriteria pada *Model Predictive Control*

Seperti yang telah dinyatakan sebelumnya bahwa perhitungan sinyal kendali pada MPC dilakukan dengan meminimumkan suatu fungsi kriteria. Fungsi kriteria yang digunakan dalam algoritma MPC berbentuk kuadratik seperti berikut

$$V(k) = \sum_{i=1}^{Hp} \|\hat{y}(k+i|k) - r(k+i|k)\|_{\underline{Q}(i)}^2 + \sum_{i=0}^{Hu-1} \|\Delta \hat{u}(k+i|k)\|_{\underline{R}(i)}^2 \quad (2.9)$$

dengan :

$\hat{y}(k+i|k)$  = keluaran terprediksi untuk  $i$ -langkah kedepan saat waktu  $k$

$\underline{r}(k+i|k)$  = nilai trayektori acuan (*reference trajectory*)

$\underline{\Delta\hat{u}}(k+i|k)$  = perubahan nilai sinyal kendali terprediksi untuk  $i$ -langkah  
kedepan saat waktu  $k$

$\underline{Q}(i)$  dan  $\underline{R}(i)$  = faktor bobot

$H_p$  = *prediction horizon*

$H_u$  = *control horizon*

Dari persamaan fungsi kriteria tersebut, selalu dibuat asumsi bahwa nilai  $H_u < H_p$  dan  $\underline{\Delta\hat{u}}(k+i|k) = 0$  untuk  $i \geq H_u$ , sehingga nilai masukan terprediksi  $\hat{u}(k+i|k) = \hat{u}(k+H_u-i|k)$  untuk semua  $i \geq H_u$  seperti yang terlihat pada gambar 2.2.

Bentuk dari fungsi kriteria pada persamaan (2.9) menyatakan bahwa vektor kesalahan  $\hat{y}(k+i|k) - \underline{r}(k+i|k)$  dibebankan pada setiap rentang *prediction horizon*. Walaupun demikian tetap ada kemungkinan untuk menghitung vektor kesalahan pada titik-titik tertentu saja dengan cara mengatur matiks faktor bobot  $\underline{Q}(i)$  bernilai nol pada langkah yang diinginkan. Selain vektor kesalahan, fungsi kriteria pada persamaan (2.9) juga memperhitungkan perubahan vektor masukan dalam rentang *control horizon*. Pemilihan penggunaan  $\underline{\Delta\hat{u}}(k+i|k)$  yang pada fungsi kriteria bertujuan untuk meminimumkan perubahan sinyal kendali yang masuk ke *plant*.

### 2.2.3 Model Proses

Pada pembahasan skripsi ini, model proses yang digunakan berupa model ruang keadaan diskrit linier seperti berikut :

$$\underline{x}(k+1) = \underline{A}\underline{x}(k) + \underline{B}\underline{u}(k) \quad (2.10)$$

$$\underline{y}(k) = \underline{C}\underline{x}(k) \quad (2.11)$$

dengan :

$\underline{u}(k)$  = vektor masukan berdimensi- $l$

$\underline{x}(k)$  = vektor keadaan berdimensi- $n$

$\underline{y}(k)$  = vektor keluaran berdimensi- $m$

$\underline{A}$  = matriks keadaan berdimensi  $n \times n$

$\underline{B}$  = matriks masukan berdimensi  $n \times l$

$\underline{C}$  = matriks keluaran berdimensi  $m \times n$

Model ruang keadaan pada persamaan (2.10) dan (2.11) adalah model ruang keadaan untuk proses yang bersifat linier. Pada skripsi ini, vektor masukan  $\underline{u}(k)$  dan keluaran  $\underline{y}(k)$  masing-masing berdimensi satu.

#### 2.2.4 Prediksi

Dalam menyelesaikan masalah pengendali prediktif, nilai keluaran terprediksi  $\hat{y}(k+i|k)$  harus dapat dihitung dengan menggunakan estimasi terbaik dari variabel keadaan saat ini  $\underline{x}(k)$ , nilai masukan yang lampau  $\underline{u}(k-1)$ , dan nilai perkiraan dari perubahan masukan yang akan datang  $\Delta\hat{u}(k+i|k)$ . Sebelum melangkah lebih jauh, hal pertama yang harus dilakukan adalah memprediksi nilai variabel keadaan dengan melakukan iterasi model ruang keadaan pada persamaan (2.10) dan (2.11). Perhitungan prediksi variabel keadaan adalah sebagai berikut

$$\hat{\underline{x}}(k+1|k) = \underline{A}\underline{x}(k) + \underline{B}\hat{\underline{u}}(k|k) \quad (2.12)$$

$$\begin{aligned} \hat{\underline{x}}(k+2|k) &= \underline{A}\hat{\underline{x}}(k+1|k) + \underline{B}\hat{\underline{u}}(k+1|k) \\ &= \underline{A}^2\underline{x}(k) + \underline{A}\underline{B}\hat{\underline{u}}(k|k) + \underline{B}\hat{\underline{u}}(k+1|k) \end{aligned} \quad (2.13)$$

⋮

$$\begin{aligned} \hat{\underline{x}}(k+Hp|k) &= \underline{A}\hat{\underline{x}}(k+Hp-1|k) + \underline{B}\hat{\underline{u}}(k+Hp-1|k) \\ &= \underline{A}^{Hp}\underline{x}(k) + \underline{A}^{Hp-1}\underline{B}\hat{\underline{u}}(k|k) + \dots + \underline{B}\hat{\underline{u}}(k+Hp-1|k) \end{aligned} \quad (2.14)$$

Pada setiap langkah prediksi digunakan  $\hat{\underline{u}}(k|k)$  bukan  $\underline{u}(k)$ , karena besarnya nilai  $\underline{u}(k)$  belum diketahui ketika menghitung prediksi.

Sekarang, diasumsikan bahwa nilai masukan hanya berubah pada waktu  $k$ ,  $k+1$ , ...,  $k+Hu-1$ , dan setelah itu menjadi konstan, sehingga didapatkan bahwa  $\hat{u}(k+i|k) = \hat{u}(k+Hu-1|k)$  untuk  $Hu \leq i \leq Hp-1$ . Selanjutnya, perhitungan prediksi diubah sehingga mengandung  $\underline{\Delta\hat{u}}(k+i|k)$  daripada  $\hat{u}(k+i|k)$ , dengan

$$\underline{\Delta\hat{u}}(k+i|k) = \hat{u}(k+i|k) - \hat{u}(k+i-1|k) \quad (2.15)$$

dan pada setiap waktu pencuplikan  $k$  nilai yang sudah diketahui hanya  $\underline{u}(k-1)$ , maka

$$\hat{u}(k|k) = \underline{\Delta\hat{u}}(k|k) + \underline{u}(k-1|k) \quad (2.16)$$

$$\hat{u}(k+1|k) = \underline{\Delta\hat{u}}(k+1|k) + \underline{\Delta\hat{u}}(k|k) + \underline{u}(k-1|k) \quad (2.17)$$

⋮

$$\hat{u}(k+Hu-1|k) = \underline{\Delta\hat{u}}(k+Hu-1|k) + \dots + \underline{\Delta\hat{u}}(k|k) + \underline{u}(k-1|k) \quad (2.18)$$

Dengan mensubstitusikan persamaan (2.16) – (2.18) ke persamaan (2.12) – (2.14), diperoleh persamaan

$$\hat{x}(k+1|k) = \underline{A}x(k) + \underline{B}[\underline{\Delta\hat{u}}(k|k) + \underline{u}(k-1)] \quad (2.19)$$

$$\begin{aligned} \hat{x}(k+2|k) &= \underline{A}^2x(k) + \underline{A}\underline{B}[\underline{\Delta\hat{u}}(k|k) + \underline{u}(k-1)] \\ &\quad + \underline{B}[\underline{\Delta\hat{u}}(k+1|k) + \underline{\Delta\hat{u}}(k|k) + \underline{u}(k-1)] \\ &= \underline{A}^2x(k) + (\underline{A} + \underline{I})\underline{B}\underline{\Delta\hat{u}}(k|k) + \underline{B}\underline{\Delta\hat{u}}(k+1|k) + (\underline{A} + \underline{I})\underline{B}\underline{u}(k-1) \end{aligned} \quad (2.20)$$

⋮

$$\begin{aligned} \hat{x}(k+Hu|k) &= \underline{A}^{Hu}x(k) + (\underline{A}^{Hu-1} + \dots + \underline{A} + \underline{I})\underline{B}\underline{\Delta\hat{u}}(k|k) + \dots \\ &\quad + \underline{B}\underline{\Delta\hat{u}}(k+Hu-1|k) + (\underline{A}^{Hu-1} + \dots + \underline{A} + \underline{I})\underline{B}\underline{u}(k-1) \end{aligned} \quad (2.21)$$

Dengan mengacu pada persamaan  $\hat{u}(k+i|k) = \hat{u}(k+Hu-i|k)$  untuk  $i > Hu$ , maka perhitungan prediksi untuk  $i > Hu$  adalah

$$\begin{aligned} \hat{x}(k+Hu+1|k) &= \underline{A}^{Hu+1}x(k) + (\underline{A}^{Hu} + \dots + \underline{A} + \underline{I})\underline{B}\underline{\Delta\hat{u}}(k|k) + \dots \\ &\quad + (\underline{A} + \underline{I})\underline{B}\underline{\Delta\hat{u}}(k+Hu-1|k) + (\underline{A}^{Hu} + \dots + \underline{A} + \underline{I})\underline{B}\underline{u}(k-1) \end{aligned} \quad (2.22)$$

$$\begin{aligned}
& \vdots \\
\hat{\underline{x}}(k + Hp | k) &= \underline{A}^{Hp} \underline{x}(k) + (\underline{A}^{Hp-1} + \dots + \underline{A} + \underline{I}) \underline{B} \Delta \hat{\underline{u}}(k | k) + \dots \\
& \quad + (\underline{A}^{Hp-Hu} + \dots + \underline{A} + \underline{I}) \underline{B} \Delta \hat{\underline{u}}(k + Hu - 1 | k) \\
& \quad + (\underline{A}^{Hp-1} + \dots + \underline{A} + \underline{I}) \underline{B} \underline{u}(k - 1)
\end{aligned} \tag{2.23}$$

Akhirnya, persamaan (2.19) – (2.23) dapat disusun ke dalam bentuk vektor matriks sebagai berikut

$$\begin{aligned}
\begin{bmatrix} \hat{\underline{x}}(k+1 | k) \\ \vdots \\ \hat{\underline{x}}(k + Hu | k) \\ \hat{\underline{x}}(k + Hu + 1 | k) \\ \vdots \\ \hat{\underline{x}}(k + Hp | k) \end{bmatrix} &= \underbrace{\begin{bmatrix} \underline{A} \\ \vdots \\ \underline{A}^{Hu} \\ \underline{A}^{Hu+1} \\ \vdots \\ \underline{A}^{Hp} \end{bmatrix}}_{\Psi} \underline{x}(k) + \underbrace{\begin{bmatrix} \underline{B} \\ \vdots \\ \sum_{i=0}^{Hu-1} \underline{A}^i \underline{B} \\ \sum_{i=0}^{Hu} \underline{A}^i \underline{B} \\ \vdots \\ \sum_{i=0}^{Hp-1} \underline{A}^i \underline{B} \end{bmatrix}}_{\Gamma} \underline{u}(k-1) \\
& \quad + \underbrace{\begin{bmatrix} \underline{B} & \dots & \underline{0}_{n \times l} \\ \underline{AB} + \underline{B} & \dots & \underline{0}_{n \times l} \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{Hu-1} \underline{A}^i \underline{B} & \dots & \underline{B} \\ \sum_{i=0}^{Hu} \underline{A}^i \underline{B} & \dots & \underline{AB} + \underline{B} \\ \vdots & \vdots & \vdots \\ \sum_{i=0}^{Hp-1} \underline{A}^i \underline{B} & \dots & \sum_{i=0}^{Hp-Hu} \underline{A}^i \underline{B} \end{bmatrix}}_{\Theta} \begin{bmatrix} \Delta \hat{\underline{u}}(k) \\ \vdots \\ \Delta \hat{\underline{u}}(k + Hu - 1) \end{bmatrix}
\end{aligned} \tag{2.24}$$

Selain itu, persamaan prediksi keluaran  $\hat{\underline{y}}(k + i | k)$  dapat ditulis seperti berikut ini

$$\hat{\underline{y}}(k + 1 | k) = \underline{C} \hat{\underline{x}}(k + 1 | k) \tag{2.25}$$

$$\hat{\underline{y}}(k + 2 | k) = \underline{C} \hat{\underline{x}}(k + 2 | k) \tag{2.26}$$

$\vdots$

$$\hat{\underline{y}}(k + Hp | k) = \underline{C} \hat{\underline{x}}(k + Hp | k) \tag{2.27}$$

Persamaan (2.25) – (2.27) kemudian dapat ditulis kedalam vektor matriks sebagai berikut

$$\begin{bmatrix} \hat{y}(k+1|k) \\ \vdots \\ \hat{y}(k+Hp|k) \end{bmatrix} = \underbrace{\begin{bmatrix} \underline{C} & \underline{0}_{m \times n} & \cdots & \underline{0}_{m \times n} \\ \underline{0}_{m \times n} & \underline{C} & \cdots & \underline{0}_{m \times n} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{0}_{m \times n} & \underline{0}_{m \times n} & \cdots & \underline{C} \end{bmatrix}}_{\underline{C}_y} \begin{bmatrix} \hat{x}(k+1|k) \\ \vdots \\ \hat{x}(k+Hp|k) \end{bmatrix} \quad (2.28)$$

### 2.2.5 Strategi Pengendali *Model Predictive Control* tanpa *Constraints*

Fungsi kriteria yang akan diminimumkan sama seperti pada persamaan (2.9) dan dapat ditulis sebagai berikut

$$V(k) = \|\underline{Y}(k) - \underline{T}(k)\|_{\underline{Q}}^2 + \|\underline{\Delta U}(k)\|_{\underline{R}}^2 \quad (2.29)$$

dimana

$$\underline{Y}(k) = \begin{bmatrix} \hat{y}(k+1|k) \\ \vdots \\ \hat{y}(k+Hp|k) \end{bmatrix}, \quad \underline{T}(k) = \begin{bmatrix} r(k+1|k) \\ \vdots \\ r(k+Hp|k) \end{bmatrix},$$

$$\underline{\Delta U}(k) = \begin{bmatrix} \hat{u}(k|k) \\ \vdots \\ \hat{u}(k+Hu-1|k) \end{bmatrix}$$

dan matriks faktor bobot  $\underline{Q}$  dan  $\underline{R}$  adalah sebagai berikut

$$\underline{Q} = \begin{bmatrix} Q(1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Q(Hp) \end{bmatrix} \quad (2.30)$$

$$\underline{R} = \begin{bmatrix} R(0) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R(Hu-1) \end{bmatrix} \quad (2.31)$$

Berdasarkan pada persamaan ruang keadaan (2.24) dan (2.28), maka matriks  $\underline{Y}(k)$  dapat ditulis dalam bentuk

$$\underline{Y}(k) = \underline{C}_y \underline{\Psi} x(k) + \underline{C}_y \underline{\Gamma} u(k-1) + \underline{C}_y \underline{\Theta} \underline{\Delta U}(k) \quad (2.32)$$

Selain matriks-matriks di atas, didefinisikan juga suatu matriks penjejukan kesalahan  $\underline{E}(k)$ , yaitu selisih antara nilai trayektori acuan yang akan datang dengan tanggapan bebas dari sistem. Tanggapan bebas adalah tanggapan yang akan terjadi pada rentang *prediction horizon* jika tidak ada perubahan nilai masukan ( $\underline{\Delta U}(k) = 0$ ) [3]. Persamaan matematis dari matriks  $\underline{E}(k)$  adalah sebagai berikut

$$\underline{E}(k) = \underline{T}(k) - \underline{C}_y \underline{\Psi} x(k) - \underline{C}_y \underline{\Gamma} u(k-1) \quad (2.33)$$

Persamaan (2.29) kemudian dapat ditulis kembali dalam bentuk yang mengandung matriks  $\underline{E}(k)$  dan  $\underline{\Delta U}(k)$  sebagai berikut

$$V(k) = \|\underline{C}_y \underline{\Theta} \underline{\Delta U}(k) - \underline{E}(k)\|_{\underline{Q}}^2 + \|\underline{\Delta U}(k)\|_{\underline{R}}^2 \quad (2.34)$$

$$= [\underline{\Delta U}^T(k) \underline{\Theta}^T \underline{C}_y^T - \underline{E}^T(k)] \underline{Q} [\underline{C}_y \underline{\Theta} \underline{\Delta U}(k) - \underline{E}(k)] + \underline{\Delta U}^T(k) \underline{R} \underline{\Delta U}(k) \quad (2.35)$$

$$= \underbrace{\underline{E}^T(k) \underline{Q} \underline{E}(k)}_{\underline{c}_1} - \underbrace{\underline{\Delta U}^T(k) 2 \underline{\Theta}^T \underline{C}_y^T \underline{Q} \underline{E}(k)}_{\underline{G}} + \underbrace{\underline{\Delta U}^T(k) [\underline{\Theta}^T \underline{C}_y^T \underline{Q} \underline{C}_y \underline{\Theta} + \underline{R}] \underline{\Delta U}(k)}_{\underline{H}} \quad (2.36)$$

Pada persamaan (2.36), bagian  $\underline{E}^T(k) \underline{Q} \underline{E}(k)$  tidak mengandung unsur  $\underline{\Delta U}(k)$  sehingga bagian tersebut bisa dianggap konstan sehingga bagian tersebut tidak diikutsertakan dalam proses optimasi untuk menghitung nilai  $\underline{\Delta U}(k)$ . Persamaan (2.36) kemudian dapat ditulis kembali menjadi

$$V(k) = \underline{c}_1 - \underline{\Delta U}^T(k) \underline{G} + \underline{\Delta U}^T(k) \underline{H} \underline{\Delta U}(k) \quad (2.37)$$

dimana

$$\underline{G} = 2 \underline{\Theta}^T \underline{C}_y^T \underline{Q} \underline{E}(k) \quad (2.38)$$

dan

$$\underline{\mathcal{H}} = \underline{\Theta}^T \underline{C}_y^T \underline{Q} \underline{C}_y \underline{\Theta} + \underline{R} \quad (2.39)$$

Nilai optimal  $\underline{\Delta U}(k)$  dapat dihitung dengan membuat gradien dari  $V(k)$  bernilai nol [3]. Gradien  $V(k)$  dari persamaan (2.37) adalah

$$\nabla_{\underline{\Delta U}(k)} V(k) = -\underline{G} + 2\underline{\mathcal{H}} \underline{\Delta U}(k) \quad (2.40)$$

Dengan membuat nol nilai  $\nabla_{\underline{\Delta U}(k)} V(k)$  pada persamaan (2.40), maka didapatkan nilai optimal dari perubahan sinyal kendali sebagai berikut

$$\underline{\Delta U}(k)_{opt} = \frac{1}{2} \underline{\mathcal{H}}^{-1} \underline{G} \quad (2.41)$$

Setelah nilai matriks  $\underline{\Delta U}(k)$  didapatkan, maka nilai yang digunakan untuk mengubah sinyal kendali hanya nilai dari baris pertama matriks  $\underline{\Delta U}(k)$  sedangkan nilai dari baris yang lain dari matriks  $\underline{\Delta U}(k)$  dibuang [3].

## 2.2.6 Strategi Pengendali Model Predictive Control dengan Constraints

### 2.2.6.1 Pembentukan Constraints

Pada setiap kendali proses, pasti terdapat batasan atau *constraints* pada amplitudo sinyal kendali. Selain itu, besarnya *slew rate* sinyal kendali juga dapat menjadi batasan. Persamaan *constraints* untuk amplitudo dan *slew rate* sinyal kendali secara berturut-turut adalah sebagai berikut

$$\underline{F} \underline{U}(k) \leq \underline{f} \quad (2.42)$$

$$\underline{E} \underline{\Delta U}(k) \leq \underline{e} \quad (2.43)$$

Pada algoritma MPC, yang akan dihitung adalah nilai optimal perubahan sinyal kendali  $\underline{\Delta U}(k)$  sehingga sangat perlu untuk mengubah bentuk *constraints* yang belum mengandung  $\underline{\Delta U}(k)$  menjadi bentuk *constraints* yang mengandung  $\underline{\Delta U}(k)$ . Sebagai contoh adalah pertidaksamaan (2.42), karena pada pertidaksamaan (2.42) belum mengandung  $\underline{\Delta U}(k)$  maka bentuk pertidaksamaan (2.42) harus diubah terlebih dahulu menjadi bentuk yang mengandung  $\underline{\Delta U}(k)$ .

Untuk *constraints* yang berupa batasan nilai maksimum dan minimum sinyal kendali, maka pertidaksamaannya dapat ditulis sebagai berikut

$$\underline{u}_{\min} \leq \underline{u}(k) \leq \underline{u}_{\max} \quad (2.44)$$

Pertidaksamaan (2.44) dapat ditulis menjadi dua bentuk yang terpisah seperti berikut ini

$$-\underline{u}(k) \leq -\underline{u}_{\min} \quad (2.45)$$

$$\underline{u}(k) \leq \underline{u}_{\max} \quad (2.46)$$

Pertidaksamaan (2.45) dan (2.46) masing-masing dapat ditulis dalam bentuk yang mengandung  $\Delta U(k)$  menjadi

$$-\underline{F}'\Delta U(k) \leq -\underline{u}_{\min} + \underline{F}_1 u(k-1) \quad (2.47)$$

$$\underline{F}'\Delta U(k) \leq \underline{u}_{\max} - \underline{F}_1 u(k-1) \quad (2.48)$$

dimana

$$\underline{F}' = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix}_{HuxHu} \quad (2.49)$$

dan

$$\underline{F}_1 = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{Hux1} \quad (2.50)$$

Untuk pertidaksamaan (2.43), bentuknya tidak perlu diubah lagi karena pada pertidaksamaan tersebut sudah mengandung unsur  $\Delta U(k)$ .

Pertidaksamaan (2.43), (2.47), dan (2.48) kemudian dapat disusun menjadi sebuah vektor matriks sebagai berikut

$$\underbrace{\begin{bmatrix} -\underline{F}' \\ \underline{F}' \\ \underline{E} \end{bmatrix}}_{\underline{\Omega}} \underbrace{\Delta \underline{U}(k)}_{\underline{\delta}} \leq \underbrace{\begin{bmatrix} -\underline{u}_{\min} + \underline{F}_1 u(k-1) \\ \underline{u}_{\max} - \underline{F}_1 u(k-1) \\ \underline{e} \end{bmatrix}}_{\underline{\omega}} \quad (2.51)$$

Vektor matriks pada pertidaksamaan (2.51) digunakan pada perhitungan nilai optimal perubahan sinyal kendali  $\underline{\Delta U}(k)_{opt}$ .

### 2.2.6.2 Metode *Quadratic Programming*

Fungsi kriteria pada pengendali MPC dengan *constraints* sama dengan fungsi kriteria pada pengendali MPC tanpa *constraints* (persamaan (2.37)). Permasalahan utama proses optimasi ini adalah meminimalkan fungsi kriteria

$$\underline{\Delta U}^T(k) \underline{H} \underline{\Delta U}(k) - \underline{\Delta U}^T(k) \underline{G} \quad (2.52)$$

berdasarkan pada pertidaksamaan *constraint* (2.51) atau

$$\min_{\underline{\delta}} \frac{1}{2} \underline{\delta}^T \underline{\Phi} \underline{\delta} + \underline{\phi} \underline{\delta} \quad (2.53)$$

berdasarkan pada *constraints*

$$\underline{\Omega} \underline{\delta} \leq \underline{\omega} \quad (2.54)$$

Bentuk (2.53) dan (2.54) adalah masalah optimasi standar yang disebut sebagai permasalahan *Quadratic Programming* (QP). Bila ada bagian yang aktif di dalam himpunan *constraints* pada persamaan (2.54), maka bagian aktif tersebut akan membuat pertidaksamaan (2.54) menjadi suatu persamaan

$$\underline{\Omega}_a \underline{\delta} = \underline{\omega}_a \quad (2.55)$$

dengan matriks  $\underline{\Omega}_a$  adalah bagian yang aktif dari matriks pertidaksamaan (2.54). Persamaan (2.55) kemudian dijadikan sebagai *constraints* dari fungsi kriteria pada persamaan (2.53).

Permasalahan optimasi persamaan (2.53) dengan subyek terhadap persamaan (2.55) dapat diselesaikan dengan teori pengali *Lagrange*

$$\min_{\underline{\delta}, \underline{\lambda}} L(\underline{\delta}, \underline{\lambda}) \quad (2.56)$$

dengan

$$L(\underline{\delta}, \underline{\lambda}) = \frac{1}{2} \underline{\delta}^T \underline{\Phi} \underline{\delta} + \underline{\phi} \underline{\delta} + \underline{\lambda} (\underline{\Omega}_a \underline{\delta} - \underline{\omega}_a) \quad (2.57)$$

Selanjutnya dengan melakukan diferensiasi parsial terhadap  $\underline{\delta}$  dan  $\underline{\lambda}$  dari persamaan (2.57), maka didapatkan kondisi *Karush-Kuhn-Tucker* sebagai berikut

$$\nabla_{\underline{\delta}} L(\underline{\delta}, \underline{\lambda}) = \underline{\Phi} \underline{\delta} + \underline{\phi} + \underline{\Omega}_a^T \underline{\lambda} \quad (2.58)$$

$$\nabla_{\underline{\lambda}} L(\underline{\delta}, \underline{\lambda}) = \underline{\Omega}_a \underline{\delta} - \underline{\omega}_a \quad (2.59)$$

atau

$$\nabla L(\underline{\delta}, \underline{\lambda}) = \begin{bmatrix} \underline{\Phi} & \underline{\Omega}_a^T \\ \underline{\Omega}_a & \underline{0} \end{bmatrix} \begin{bmatrix} \underline{\delta} \\ \underline{\lambda} \end{bmatrix} - \begin{bmatrix} -\underline{\phi} \\ \underline{\omega}_a \end{bmatrix} \quad (2.60)$$

Selanjutnya dengan membuat  $\nabla L(\underline{\delta}, \underline{\lambda}) = 0$ , maka didapatkan solusi optimal untuk  $\underline{\delta}$  dan  $\underline{\lambda}$  sebagai berikut

$$\begin{bmatrix} \underline{\delta} \\ \underline{\lambda} \end{bmatrix}_{opt} = \begin{bmatrix} \underline{\Phi} & \underline{\Omega}_a^T \\ \underline{\Omega}_a & \underline{0} \end{bmatrix}^{-1} \begin{bmatrix} -\underline{\phi} \\ \underline{\omega}_a \end{bmatrix} \quad (2.61)$$

Solusi pada *Quadratic Programming* pada kondisi normal menghasilkan nilai yang *feasible*, yaitu nilai yang memenuhi pertidaksamaan *constraints* yang ada dan dapat menghasilkan nilai fungsi kriteria minimum. Masalah yang paling sering muncul pada optimasi dengan *constraints* adalah solusi yang *infeasible*, dimana nilai yang dihasilkan tidak memenuhi pertidaksamaan *constraints* yang ada. *QP solver* akan menghentikan proses perhitungan jika terjadi solusi yang *infeasible*. Hal ini tentu tidak dapat diterima karena sinyal kendali hasil komputasi harus selalu ada untuk digunakan sebagai masukan bagi *plant*, sehingga sangat penting untuk membuat metode cadangan dalam menghitung sinyal masukan

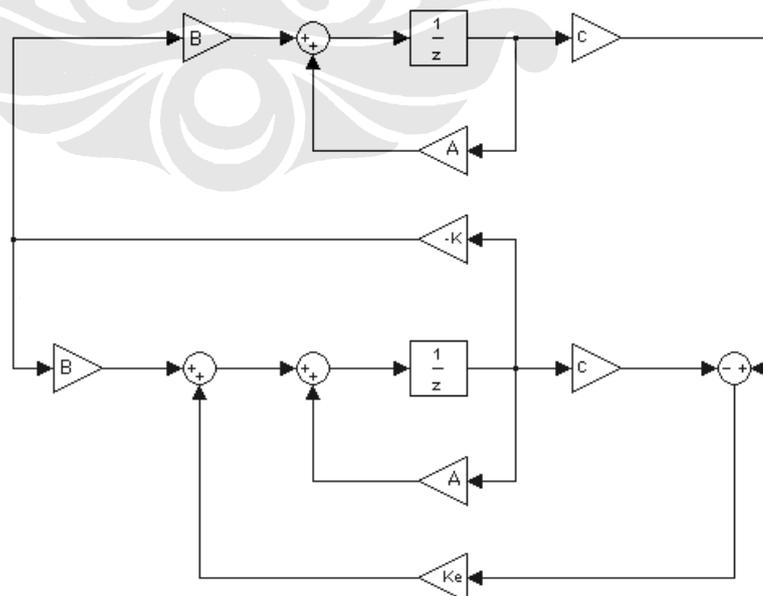
ketika algoritma MPC diterapkan. Beberapa pendekatan yang dapat dilakukan untuk menghindari terjadinya solusi yang *infeasible* pada MPC antara lain [3] :

- Menghindari *constraints* pada keluaran
- Mengatur *constraints* untuk setiap langkah pencuplikan  $k$
- Mengatur *horizon* untuk setiap langkah pencuplikan  $k$

### 2.3 REDUCED-ORDER STATE OBSERVER

Dalam suatu proses kontrol industri, sensor merupakan hal yang memegang peranan yang sangat penting. Semua data atau yang sering disebut dengan *state* yang akan di kalkulasi oleh pengendali adalah data yang direkam oleh sensor-sensor yang ada dalam sistem tersebut. Dalam banyak kasus praktikal, hanya sedikit variabel *state* yang terukur dan sisa nya adalah *state* yang tidak dapat terukur oleh sensor. Oleh karena itu, *state variable* yang tidak terukur dapat di estimasi dan hal ini sering disebut dengan proses observasi. Sistem nyata membutuhkan observasi atau estimasi *state variable* yang tidak terukur dari data-data keluaran dan variabel kendali. Untuk melakukan observasi sebagian *state variable* yang tidak terukur maka dapat dipakai algoritma observasi yang disebut *Reduced-Order State Observation*.

Berikut ini adalah skematik sederhana dari sebuah *Observed-State Feedback Control System* :



Gambar 2.3. Skematik *Observed-State Feedback Control System*

Untuk mendisain sebuah *reduced-order state observer*, asumsikan bahwa *state vector*  $x(k)$  adalah sebuah n-vektor dan *output vector*  $y(k)$  adalah m-vektor yang dapat diukur. Sehingga kita harus melakukan estimasi untuk sejumlah n-m vektor.

### 2.3.1 Pembentukan Persamaan *State* dan Persamaan Keluaran

*Reduced-order observer* dapat didisain dengan melakukan proses partisi *state vector*  $x(k)$  kedalam dua bagian [4] yaitu :

$$x(k) = \begin{bmatrix} x_a(k) \\ x_b(k) \end{bmatrix}$$

dimana  $x_a(k)$  adalah bagian dari *state vector* yang tidak dapat terukur (sehingga  $x_a(k)$  adalah sebuah (n-m) vektor) sedangkan  $x_b(k)$  adalah bagian dari *state vector* yang dapat terukur (sehingga  $x_b(k)$  adalah sebuah m-vektor). Partisi persamaan keadaan sistem menjadi seperti berikut :

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(k) \quad (2.62)$$

$$y(k) = \begin{bmatrix} I & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \quad (2.63)$$

dengan menuliskan ulang persamaan (2.62), maka persamaan untuk bagian *state vector* yang dapat diukur menjadi :

$$x_2(k+1) = A_{21}x_1(k) + A_{22}x_2(k) + B_2u(k)$$

atau

$$x_2(k+1) - A_{22}x_2(k) - B_2u(k) = A_{21}x_1(k) \quad (2.64)$$

dimana persamaan pada ruas kiri adalah *state* yang dapat diukur.

Persamaan (2.64) ini sering disebut juga dengan persamaan keluaran.

Dari persamaan (2.62) juga dapat dibentuk sebuah persamaan *state* yang tidak dapat diukur yaitu :

$$x_1(k+1) = A_{11}x_1(k) + A_{12}x_2(k) + B_1u(k) \quad (2.65)$$

Persamaan (2.65) ini sering disebut dengan persamaan *state*.

Persamaan-persamaan diatas dapat dianalogikan dengan persamaan keluaran dan persamaan *state* dari sebuah *full-order observer* yaitu :

$$y(k) = Cx(k)$$

dengan

$$x_2(k+1) - A_{22}x_2(k) - B_2u(k) = A_{21}x_1(k)$$

dan

$$x(k+1) = Ax(k) + Bu(k)$$

dengan

$$x_1(k+1) = A_{11}x_1(k) + [A_{12}x_2(k) + B_1u(k)]$$

Untuk mendisain *reduced-order observer*, kita dapat membuat persamaan *observer* sebagai berikut :

$$\tilde{x}_1(k+1) = (A_{11} - K_e A_{21})\tilde{x}_1(k) + A_{12}x_2(k) + B_1u(k) + K_e [x_2(k+1) - A_{22}x_2(k) - B_2u(k)] \quad (2.66)$$

Persamaan (2.66) dapat dianalogikan dengan persamaan *observer* pada *full-order observer* yaitu sebagai berikut :

$$\tilde{x}(k+1) = (A - K_e C)\tilde{x}(k) + Bu(k) + K_e y(k) \quad (2.67)$$

### 2.3.2 Pembentukan Persamaan Dinamik *Reduced-Order Observer*

Berdasarkan persamaan (2.63), terdapat sebuah hubungan bahwa  $y(k) = x_2(k)$ , jika hubungan ini dimasukkan ke dalam persamaan (2.66) maka akan didapat :

$$\tilde{x}_1(k+1) = (A_{11} - K_e A_{21})\tilde{x}_1(k) + K_e y(k+1) + (A_{12} - K_e A_{22})y(k) + (B_1 - K_e B_2)u(k) \quad (2.68)$$

Persamaan (2.68) diatas masih memiliki nilai  $y(k+1)$  sehingga kita harus mengukur nilai ini dan hal ini merupakan sesuatu yang menyulitkan sehingga persamaan (2.68) diatas dapat dimodifikasi menjadi berikut :

$$\begin{aligned}
& \tilde{x}_1(k+1) - K_e y(k+1) \\
&= (A_{11} - K_e A_{21}) \tilde{x}_1(k) + (A_{12} - K_e A_{22}) y(k) + (B_1 - K_e B_2) u(k) \\
&= (A_{11} - K_e A_{21}) [\tilde{x}_1(k) - K_e y(k)] + [(A_{11} - K_e A_{21}) K_e + A_{12} - K_e A_{22}] y(k) + (B_1 - K_e B_2) u(k) \quad (2.69)
\end{aligned}$$

definisikan bahwa

$$x_1(k) - K_e y(k) = x_1 - K_e x_2(k) = \eta(k) \quad (2.70)$$

dan

$$\tilde{x}_1(k) - K_e y(k) = \tilde{x}_1 - K_e x_2(k) = \tilde{\eta}(k) \quad (2.71)$$

Sehingga persamaan (2.69) dapat ditulis sebagai berikut :

$$\tilde{\eta}(k+1) = (A_{11} - K_e A_{21}) \tilde{\eta}(k) + [(A_{11} - K_e A_{21}) K_e + A_{12} - K_e A_{22}] y(k) + (B_1 - K_e B_2) u(k) \quad (2.72)$$

Persamaan (2.71) dan (2.72) menunjukkan sebuah persamaan dinamik *reduced-order observer* sehingga kita tidak perlu lagi harus mengukur  $y(k+1)$ .

Persamaan kesalahan *observer* adalah sebagai berikut :

$$e(k) = \eta(k) - \tilde{\eta}(k) = x_1(k) - \tilde{x}_1(k) \quad (2.73)$$

atau

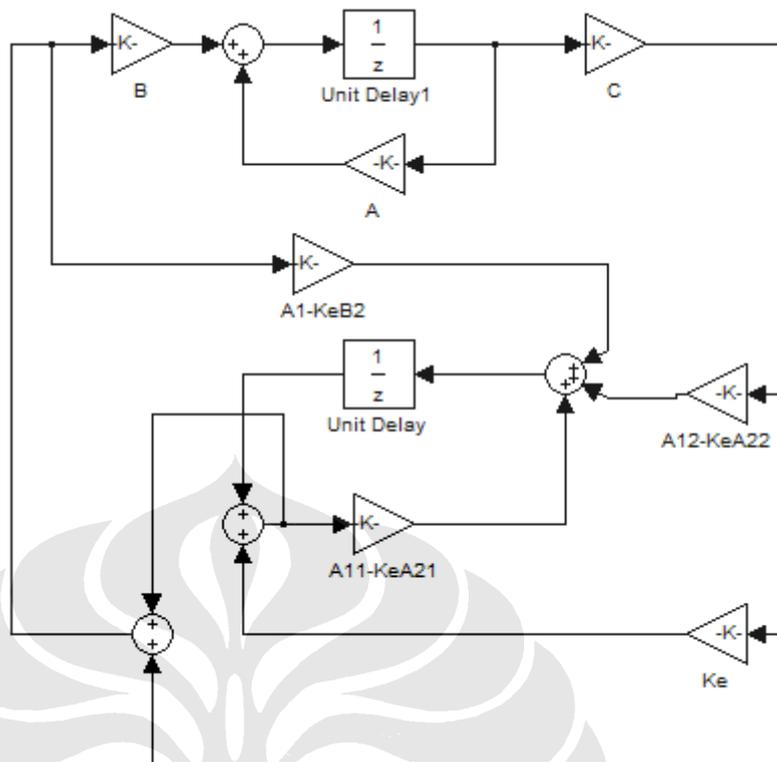
$$e(k+1) = (A_{11} - K_e A_{21}) e(k) \quad (2.74)$$

Dengan menggunakan persamaan (2.74), maka didapat persamaan karakteristik dari *reduced-order observer* adalah sebagai berikut :

$$|zI - A_{11} + K_e A_{21}| = 0 \quad (2.75)$$

dengan  $K_e$  adalah matriks penguat umpan balik *observer* yang dapat dihitung dengan memilih lokasi kutub-kutub *observer* lingkaran tertutup.

Berikut ini skematik *reduced-order observer* dengan *state feedback control system* :



Gambar 2.4. Skematik *reduced-order observer* dengan *state feedback control system*

## BAB 3

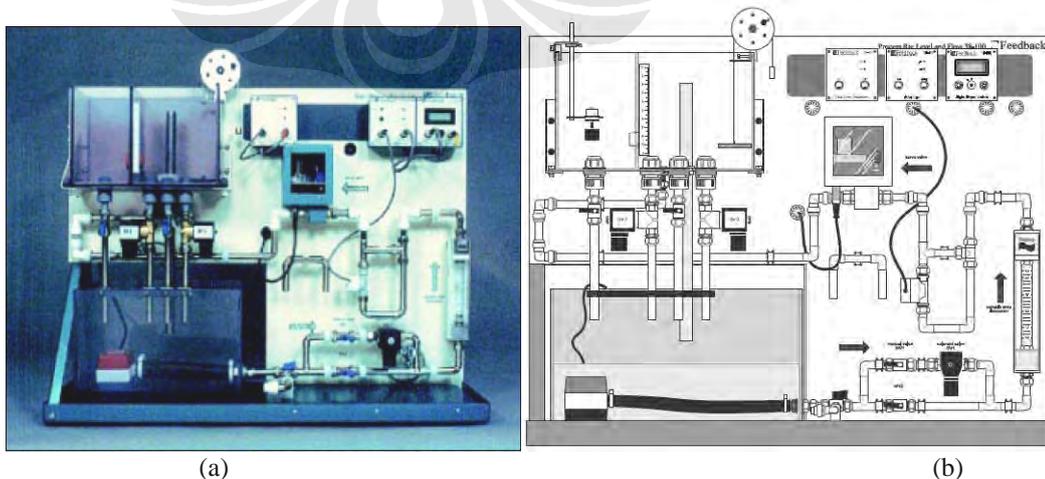
### PERANCANGAN SISTEM

#### 3.1 DESKRIPSI PROSES

##### 3.1.1 Sistem Dua Tangki Berhubungan

Sistem dua tangki berhubungan yang digunakan sebagai *plant* pada skripsi ini adalah *Coupled-Tank Basic Process Rig 38-100*. *Coupled-Tank Basic Process Rig 38-100* adalah alat yang sengaja dirancang untuk pengajaran laboratorium dari teori sistem kendali yang merupakan contoh dari sistem suatu pabrik.

*Coupled-Tank Basic Process Rig 38-100* terdiri dari dua menara tangki yang terletak di atas sebuah *reservoir* yang digunakan untuk menyimpan air (gambar 3.1). Air dari *reservoir* dapat dipompa ke dalam tangki pertama (sebelah kanan) dan bila ketinggian air cukup maka air akan mengalir ke tangki kedua (sebelah kiri). Ketinggian air pada masing-masing tangki dapat dilihat pada mistar yang terletak di depan kedua tangki. Pada setiap tangki dari alat tersebut dilengkapi juga oleh saluran keluar air yang letaknya di bagian bawahnya. Besarnya saluran keluar tersebut dapat diatur dengan cara mengatur katup pada pipa keluaran menggunakan sebuah katup pada bagian tangki kedua. Kondisi katup yang berbeda akan menyebabkan model alat menjadi berbeda. Oleh sebab itu, kondisi bukaan katup harus selalu dibuat tetap selama pembuatan model.

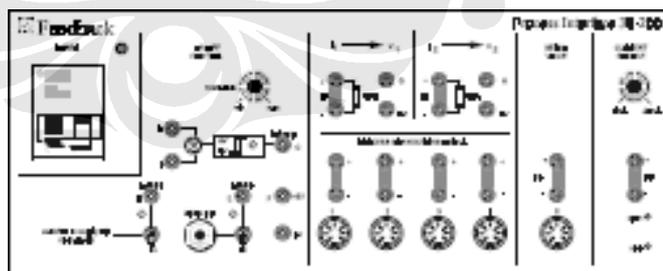


Gambar 3.1. (a) Bentuk fisik *Coupled-Tank Basic Process Rig 38-100*. (b) Skema *Coupled-Tank Basic Process Rig 38-100*.

Ketinggian air pada setiap tangki dapat diketahui dengan menggunakan sebuah penggaris mistar atau dengan menggunakan sensor tipe potensiometer yang terdapat pada tangki kedua saja [5]. Sensor tersebut mengkonversi ketinggian air menjadi tegangan listrik. Tegangan keluaran 0,4 volt diberikan pada posisi ketinggian air terendah yaitu di bawah mistar pengukur, sedangkan tegangan keluaran 2 volt diberikan pada posisi ketinggian air tertinggi tepat sebelum pipa keluaran atau di atas 100%. Pipa keluaran berfungsi untuk menjaga air agar air tidak meluap dari atas tangki.

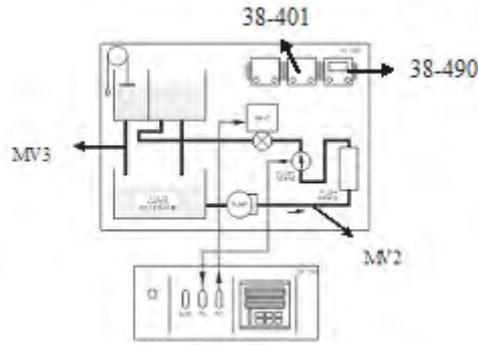
Pada skripsi ini, *plant* yang digunakan adalah *Single-Input Single-Output* (SISO) yang berbentuk model ruang keadaan. Output yang akan dikendalikan adalah ketinggian air pada tangki kedua, sedangkan aktuator yang digunakan hanya pompa pada tangki pertama. Oleh karena itu diperlukan *Reduced-Order State Observer* jika ingin mengetahui besarnya ketinggian air pada tangki pertama. Untuk menjaga agar model tidak berubah, maka besarnya sekat antara kedua tangki dan katup keluaran dari setiap tangki dijaga tetap.

Selain komponen-komponen diatas, terdapat sebuah sistem yang disebut *Process Interface 38-200* yang dihubungkan ke sistem *Basic Process Rig 38-100*, *Process Interface* ini bertugas menyediakan semua outlet daya yang dibutuhkan. *Process Interface* juga memiliki input 4 – 20 mA, sebuah sumber arus 4 – 20 mA, konverter arus ke tegangan, komparator tegangan dengan *variable hysteresis*. Sistem proteksi nya disediakan oleh *residual current circuit breaker*.



Gambar 3.2. Sketsa *Process Interface 38-200*

Berikut ini adalah sketsa dari gabungan *Basic Process Rig 38-100* dengan *Process Interface* :



Gambar 3.3. Sketsa gabungan *Basic Process Rig 38-100* dan *Process Interface*

### 3.1.2 Kalibrasi Komponen

Kalibrasi merupakan langkah pertama yang mutlak dilakukan pada sebuah sistem sebelum sistem tersebut dipakai. Berikut ini beberapa kalibrasi yang harus dilakukan pada sistem *Coupled-Tank Basic Process Rig 38-100*.

#### 3.1.2.1 Kalibrasi Servo Valve

Untuk melakukan kalibrasi *servo valve*, buka MV2 pada kondisi terbuka penuh dan MV3 pada kondisi setengah terbuka. Nyalakan *Process Interface* dan pompa kemudian naikan keluaran sumber arus pada *Process Interface* dari nilai minimal ke maksimal secara perlahan.

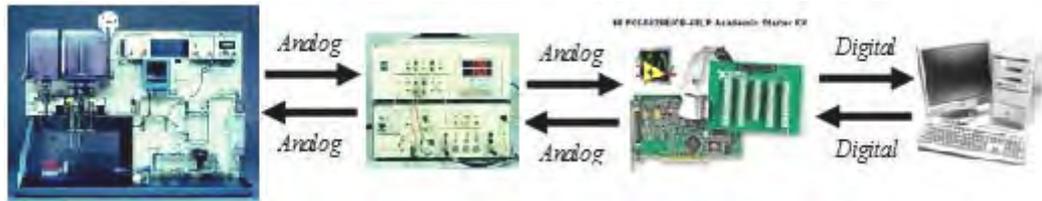
#### 3.1.2.2 Kalibrasi Sensor tipe Potensiometer

Untuk melakukan kalibrasi *Sensor tipe Potensiometer* dengan mengatur bagian *zero* pada 38-401 agar tampilan pada 38-490 menjadi 0,4 ketika *sensor tipe potensiometer* berada pada level terbawah dan mengatur bagian *span* pada 38-401 agar tampilan pada 38-490 menjadi 0,4 ketika *sensor tipe potensiometer* berada pada level tertinggi.

### 3.1.3 Interkoneksi Alat

Pengambilan data masukan dan keluaran dilakukan dengan menggunakan blok SIMULINK yang terdapat di MATLAB sehingga diperlukan interkoneksi antara *Coupled-Tank Basic Process Rig 38-100* dengan sebuah PC. Koneksi antara PC dengan *Coupled-Tank Basic Process Rig 38-100* memerlukan *Process Interface 38-200* yang dihubungkan dengan PCI-6024E yang kemudian dipasang pada slot PCI yang berada di *Motherboard* PC. PCI-6024E berfungsi sebagai *Analog-to-Digital Converter* (ADC) dan *Digital-to-Analog Converter* (DAC)

antara PC dengan alat. Skema interkoneksi antara *Coupled-Tank Basic Process Rig 38-100* dengan PC seperti gambar 3.2.



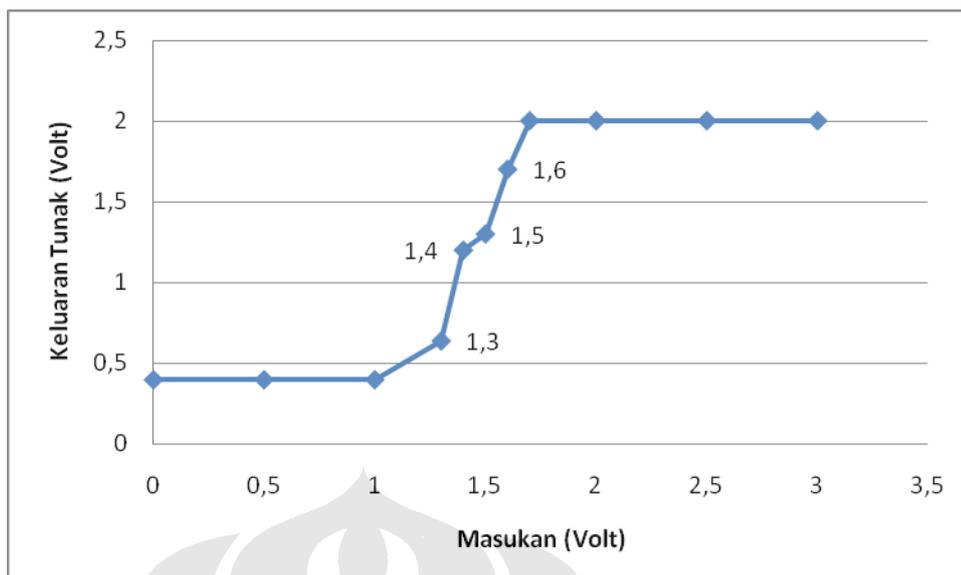
Gambar 3.4. Skema interkoneksi antara *Coupled-Tank Basic Process Rig 38-100* dengan sebuah PC

## 3.2 PEMBUATAN MODEL *COUPLED-TANK BASIC PROCESS RIG 38-100*

### 3.2.1 Penentuan Daerah Kerja

Sebelum melakukan pengambilan data untuk membuat model, hal yang harus diketahui adalah letak daerah kerja (daerah operasi) yaitu daerah dimana respon suatu sistem dianggap linear terhadap masukannya. Dengan adanya daerah kerja ini, diasumsikan pula bahwa sistem hanya beroperasi di sekitar daerah kerja ini saja.

Untuk mengetahui daerah kerja alat, maka dilakukan serangkaian percobaan dengan menggunakan uji *Step Response* dengan amplitudo sinyal masukan yang berbeda-beda. Pada skripsi ini, amplitudo sinyal masukan yang diberikan pada uji *Step Response* berkisar di antara 0 Volt dan 3 Volt. Dari hasil percobaan didapatkan grafik keluaran tunak dari sensor ketinggian  $y_{ss}$  terhadap masukan fungsi *step*  $u(t)$  seperti yang terdapat pada gambar 3.5.



Gambar 3.5. Grafik hubungan keluaran tunak terhadap masukan *Unit Step* dengan amplitudo yang berbeda-beda.

Dari grafik karakteristik keluaran tunak terhadap masukan pada gambar 3.5, dapat ditentukan titik kerja sistem, yaitu pada nilai masukan 1,5 volt. Dari acuan titik kerja ini maka ditentukanlah daerah kerja yaitu daerah yang berada di sekitar titik kerja ini. Pada skripsi ini, daerah kerja ditentukan berada dalam rentang masukan antara 1,3 volt hingga 1,6 volt. Terlihat bahwa pada sekitar daerah ini, karakteristik alat membentuk garis yang mendekati linear.

Walaupun ada bagian yang linier pada karakteristik alat, model ruang keadaan pada persamaan (2.10) tidak dapat digunakan untuk mewakili karakteristik alat. Hal ini disebabkan adanya daerah mati atau *dead zone* dan daerah saturasi pada rentang masukan tertentu. Dengan mengabaikan daerah saturasi, maka model linier dapat dibuat dengan menggunakan konstanta kompensasi masukan pada model ruang keadaan, sehingga persamaan ruang keadaan menjadi

$$\underline{x}(k+1) = \underline{A}\underline{x}(k) + \underline{B}u(k) + \underline{K} \quad (3.1)$$

Model ruang keadaan pada persamaan (3.1) nantinya akan selalu digunakan dalam perhitungan sinyal kendali dengan menggunakan MPC.

### 3.2.2 Pencarian Model Parametrik

Untuk mencari model parametrik dari *Coupled-Tank Basic Process Rig 38-100* diperlukan pengamatan terhadap kondisi fisik dari alat tersebut. Parameter-parameter sistem dua tangki terhubung yang digunakan dapat dilihat pada tabel berikut :

Tabel 3.1. Tabel parameter *Coupled-Tank Basic Process Rig 38-100*

Parameter	Nilai
Luas penampang tangki 1 ( $A_1$ )	196 cm <sup>2</sup>
Luas penampang tangki 1 ( $A_2$ )	130.9 cm <sup>2</sup>
Luas pipa keluaran 1 ( $a_1$ )	8.54865 cm <sup>2</sup>
Luas pipa keluaran 2 ( $a_2$ )	0.883125 cm <sup>2</sup>
Konstanta gravitasi (g)	980 cm/s <sup>2</sup>
Konstanta pengubah tegangan ke debit air ( $k_{VQ}$ )	18.23974
Titik kerja ( $u_0$ )	1.5 volt

Perilaku sistem dua tangki terhubung, perubahan ketinggian fluida dapat diprediksi berdasarkan kondisi kecepatan alirannya dengan mengabaikan efek dari temperatur terhadap fluida sehingga massa jenis dari fluida akan selalu tetap. Persamaan kesetimbangan massa antara tangki pertama dan kedua :

$$A_1 \frac{\partial h_1}{\partial t} = q_{in} - q_{out1} = k_{VQ} \cdot u - \sqrt{2g |h_1 - h_2|} \cdot a_1^2 \quad (3.2)$$

dan

$$A_2 \frac{\partial h_2}{\partial t} = q_{out1} - q_{out2} = a_1 \sqrt{2g |h_1 - h_2|} - \sqrt{2g |h_2|} \cdot a_2 \quad (3.3)$$

Dengan menggunakan persamaan (3.2) dan (3.3), kita dapat menurunkan persamaan ruang keadaan dari sistem tangki terhubung tersebut berupa :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \frac{-a_1^2 g}{A_1 \cdot k_{VQ} \cdot u_0} & \frac{a_1^2 g}{A_1 \cdot k_{VQ} \cdot u_0} \\ \frac{a_1^2 g}{A_2 \cdot k_{VQ} \cdot u_0} & - \left[ \frac{a_1^2 g}{A_2 \cdot k_{VQ} \cdot u_0} + \frac{a_0^2 g}{A_2 \cdot k_{VQ} \cdot u_0} \right] \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{k_{VQ}}{A_1} \\ 0 \end{bmatrix} \bar{u} \quad (3.4)$$

$$\bar{y} = [0 \quad 1] \bar{x}$$

Dengan menggunakan konstanta dari *Coupled-Tank Basic Process Rig 38-100* tersebut dimasukkan ke dalam persamaan (3.4) maka didapat model ruang keadaan sebagai berikut :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -13.3553 & 13.3553 \\ 19.9973 & -20.2107 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0.09306 \\ 0 \end{bmatrix} u$$

$$\bar{y} = [0 \quad 1] \bar{x}$$

Karena model tersebut masih berupa model continuous maka kita perlu mendiskritkan model tersebut di atas dengan menggunakan MATLAB dengan perintah *tf2ss*. Hasil pendiskritan model tersebut adalah

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.1098 & 0.07288 \\ 0.1091 & 0.07241 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0.5399 \\ 0.5337 \end{bmatrix} u(k) \quad (3.5)$$

$$y(k) = [0 \quad 1] x(k)$$

### 3.2.3 Perancangan *Reduced-Order State Observer*

#### 3.2.3.1 Pengetesan *Observability* Sistem

Ada satu langkah penting yang harus dilakukan terlebih dahulu sebelum merancang sebuah *observer* yaitu pengetesan kondisi *observability* dari sebuah sistem. Pengecekan *observability* sistem dimaksudkan untuk mengetahui apakah sistem tersebut benar-benar dapat diobservasi dan untuk mengetahui apakah *state-state* yang diobservasi tersebut dapat mewakili keadaan sistem yang sebenarnya.

Asumsikan bahwa model ruang keadaan sistem *Basic Process Rig 38-100* yang ditunjukkan pada persamaan (3.5) dapat diwakili oleh persamaan berikut :

$$x((k+1)T) = Ax(kT) + Bu(kT) \quad (3.6)$$

$$y(kT) = Cx(kT) \quad (3.7)$$

Untuk melakukan pengetesan *observability* dari suatu sistem, langkah yang harus dilakukan adalah membentuk matriks *observability* seperti yang ditunjukkan oleh persamaan berikut :

$$\left[ C^T : A^T C^T : \dots : (A^T)^{n-1} C^T \right] \quad (3.8)$$

dimana :

- $n$  adalah jumlah *state* yang dimiliki oleh sebuah sistem.
- sistem *observable* jika matriks *observability* memiliki *rank* sebanyak  $n$  (jumlah *state*)

Berdasarkan persamaan (3.8) maka matriks *observability* dari *Basic Process Rig 38-100* adalah sebagai berikut :

$$\begin{bmatrix} 0 & 0.1091 \\ 1.000 & 0.0724 \end{bmatrix}$$

*Rank* dari matriks *observability* diatas adalah 2 sehingga sistem *Basic Process Rig 38-100* dikatakan *fully observable* atau dengan kata lain semua *state* dari sistem dapat diobservasi.

### 3.2.3.2. Pembentukan Persamaan Karakteristik *Observer*

Setelah perancangan model parametrik dari sistem tersebut, maka didapat model seperti pada persamaan (3.5). Untuk merancang *reduced-order observer*, maka persamaan (3.5) harus dipartisi sebagaimana yang dilakukan dalam persamaan (2.62) dan (2.63) dimana *state vector* yang dapat diukur adalah  $x_2(k)$  dan *state vector* yang tidak dapat diukur adalah  $x_1(k)$ .

Sehingga diperoleh :

$$x_1(k) = x_1(k) \text{ dan } x_2(k) = x_2(k)$$

$$A_{11} = [0.1098], A_{12} = [0.07288], A_{21} = [0.1091], A_{22} = [0.07241]$$

$$B_1 = [0.5399], B_2 = [0.5337]$$

Parameter  $K_e$  (matriks penguat umpan balik *observer*) dapat diperoleh dengan menggunakan persamaan (2.75) dengan letak kutub-kutub *observer* lingkaran tertutup yang diinginkan berada pada titik 0,1.

Berikut ini adalah persamaan karakteristik *observer* yang akan dirancang :

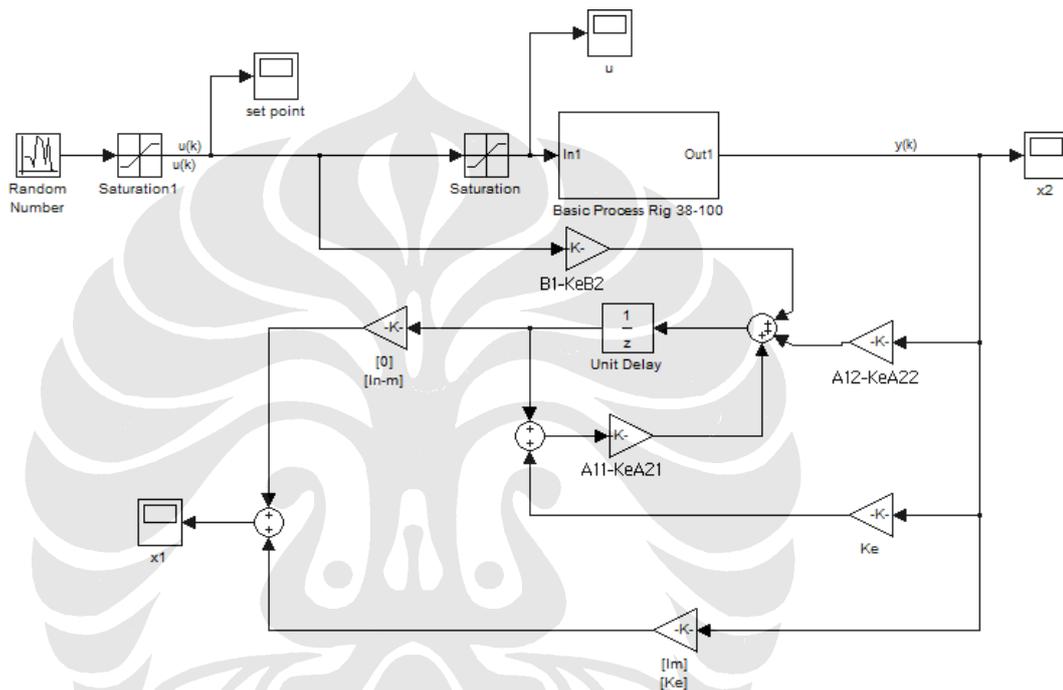
$$|zI - A_{11} + K_e A_{21}| = 0$$

didapat nilai  $K_e = 1.9230$

Rancangan blok simulink untuk *reduced-order observer* dapat dilihat pada gambar blok simulink untuk identifikasi model proses di bawah yaitu gambar 3.6.

### 3.2.4 Identifikasi Model Proses

Gambar blok SIMULINK yang digunakan seperti yang terdapat pada gambar 3.4.



Gambar 3.6. Blok SIMULINK yang digunakan untuk mengambil data masukan dan keluaran dari *Coupled-Tank Basic Process Rig 38-100*

Pada saat pengambilan data, terdapat satu buah masukan dan dua buah variabel keadaan yang dicatat sebagai informasi untuk menentukan nilai parameter-parameter estimasi model ruang keadaan dari alat tersebut. Model ruang keadaan sistem tangki terhubung adalah sebagai berikut.

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u(k) + \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \quad (3.9)$$

atau

$$\underbrace{\begin{bmatrix} x_1(k+1) & x_2(k+1) \end{bmatrix}}_{\underline{x}^T} = \underbrace{\begin{bmatrix} x_1(k) & x_2(k) & u(k) & 1^k \end{bmatrix}}_{\underline{\rho}^T} \underbrace{\begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ b_1 & b_2 \\ k_1 & k_2 \end{bmatrix}}_{\hat{\underline{\theta}}} \quad (3.10)$$

Untuk sejumlah  $N$  data, persamaan (3.10) dapat ditulis menjadi

$$\underbrace{\begin{bmatrix} x_1(2) & x_2(2) \\ x_1(3) & x_2(3) \\ \vdots & \vdots \\ x_1(N) & x_2(N) \end{bmatrix}}_{\underline{X}} = \underbrace{\begin{bmatrix} x_1(1) & x_2(1) & u(1) & 1 \\ x_1(2) & x_2(2) & u(2) & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_1(N-1) & x_2(N-1) & u(N-1) & 1 \end{bmatrix}}_{\underline{P}} \underbrace{\begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ b_1 & b_2 \\ k_1 & k_2 \end{bmatrix}}_{\hat{\underline{\theta}}} \quad (3.11)$$

Dari persamaan (3.11), maka dapat diturunkan rumus untuk menghitung nilai parameter-parameter estimasi  $\hat{\underline{\theta}}$ . Langkah-langkah untuk menghitung nilai parameter estimasi  $\hat{\underline{\theta}}$  adalah sebagai berikut

1. Memodifikasi fungsi kriteria pada persamaan (2.1) menjadi

$$\begin{aligned} J_{LS} &= (\underline{X} - \underline{P}\hat{\underline{\theta}})^T (\underline{X} - \underline{P}\hat{\underline{\theta}}) \\ &= \underline{X}^T \underline{X} - \hat{\underline{\theta}}^T \underline{P}^T \underline{X} - \underline{X}^T \underline{P} \hat{\underline{\theta}} + \hat{\underline{\theta}}^T \underline{P}^T \underline{P} \hat{\underline{\theta}} \end{aligned} \quad (3.12)$$

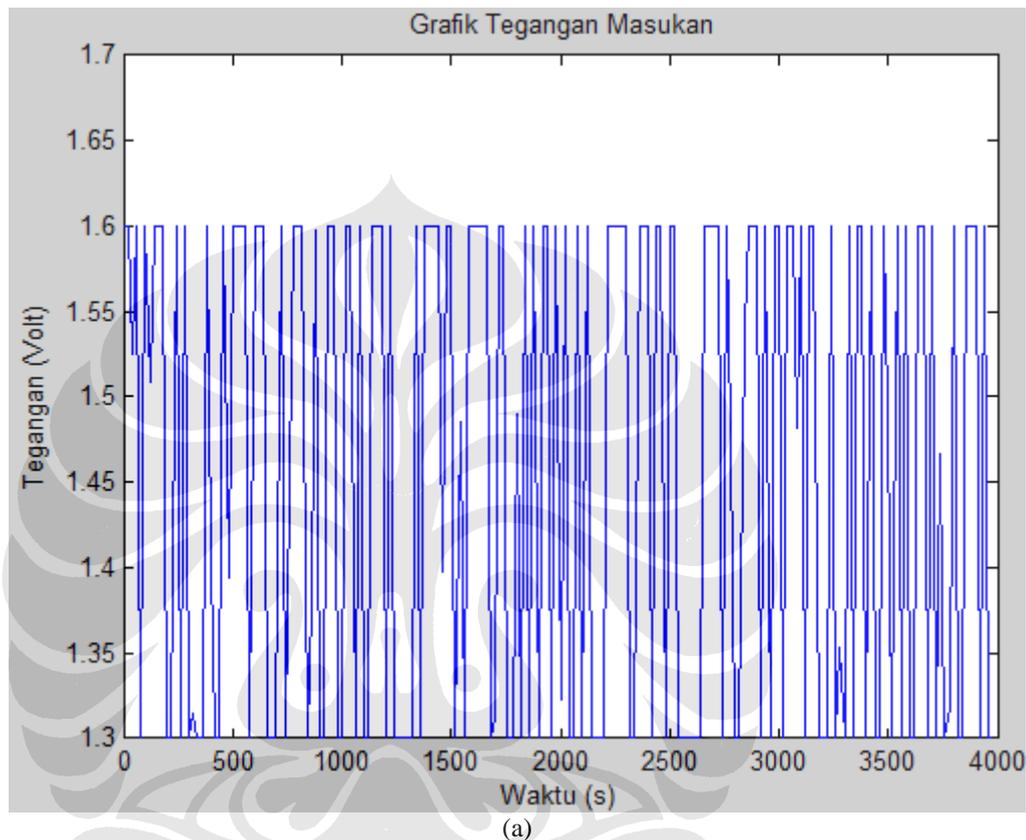
2. Dengan membuat turunan pertama dari  $J_{LS}$  terhadap  $\underline{\theta}$  bernilai nol, maka didapatkan persamaan

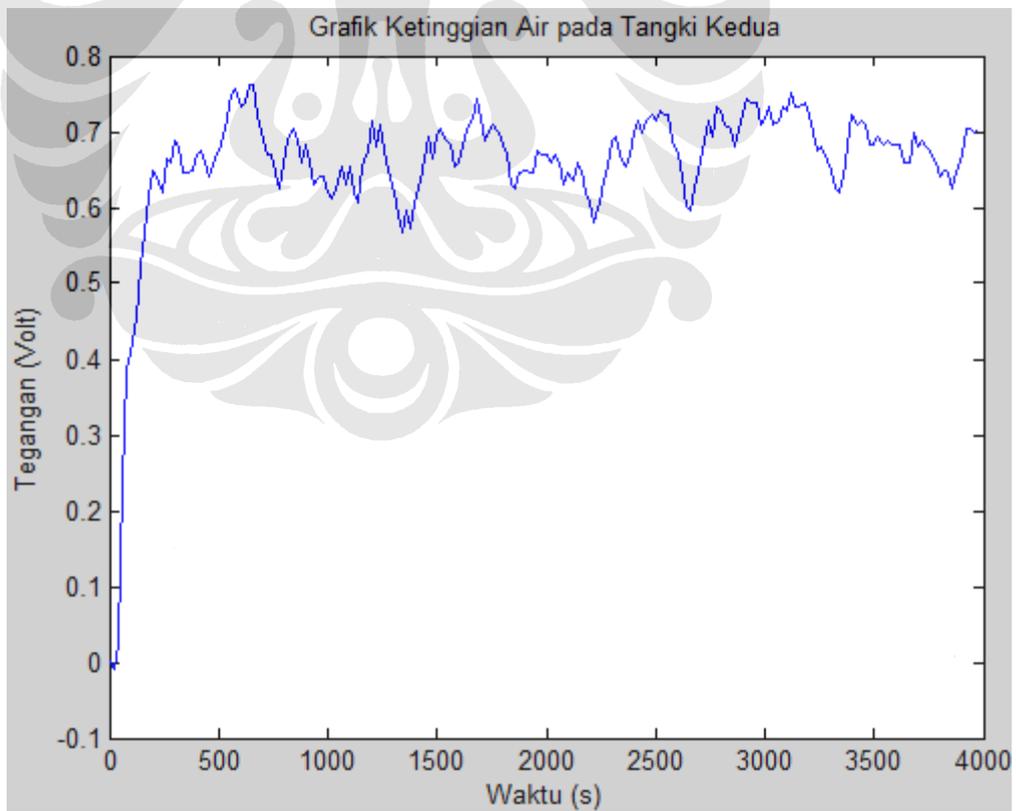
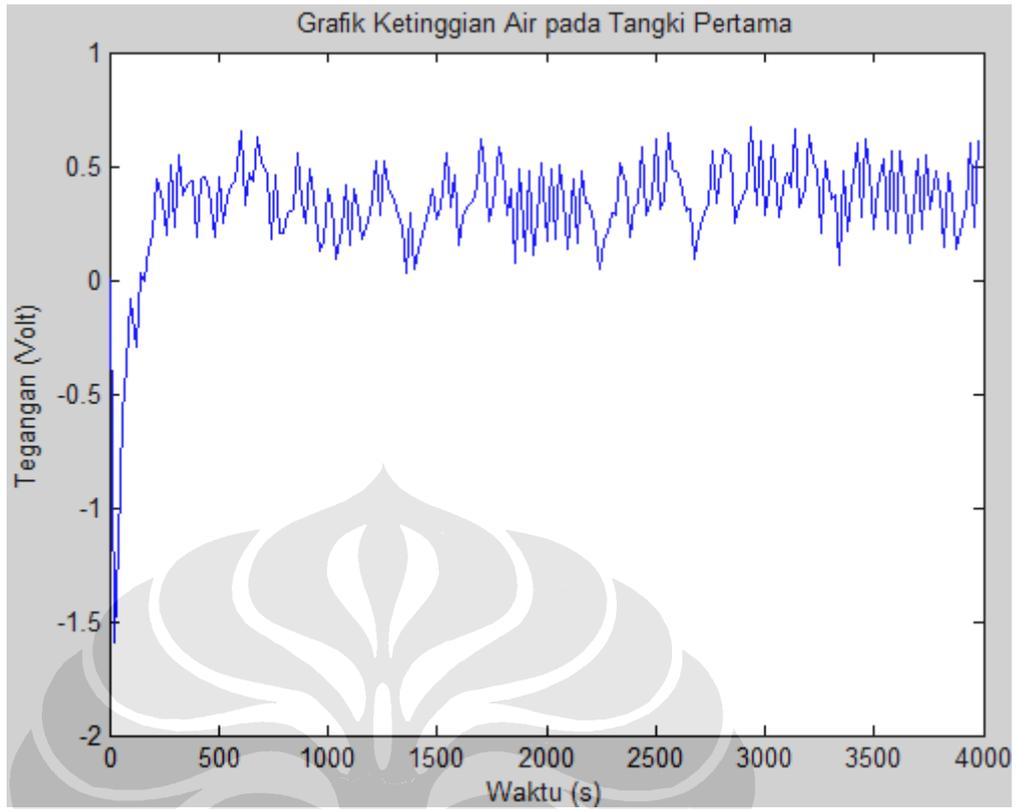
$$\left. \frac{\partial J_{LS}(\underline{\theta})}{\partial \underline{\theta}} \right|_{\underline{\theta}=\hat{\underline{\theta}}} = -2\underline{P}^T \underline{X} + 2\underline{P}^T \underline{P} \hat{\underline{\theta}} = 0 \quad (3.13)$$

3. Dari persamaan (3.13), maka didapat rumus untuk menghitung nilai parameter estimasi

$$\hat{\underline{\theta}} = (\underline{P}^T \underline{P})^{-1} \underline{P}^T \underline{X} \quad (3.14)$$

Pada percobaan, masukan yang digunakan adalah berupa *Random Number* dengan nilai rata-rata  $\bar{v}$  1,45 dan variansi  $\sigma_v^2$  bernilai 1, sedangkan nilai *sampling time*  $h$  yang digunakan adalah 20 detik. Data masukan dan keluaran terlihat pada gambar 3.7.





Gambar 3.7. (a) Grafik sinyal masukan saat identifikasi. (b) Grafik dari sensor ketinggian pada tangki pertama. (c) Grafik dari sensor ketinggian pada tangki kedua

Berdasarkan data masukan dan data keluaran, maka dengan menggunakan persamaan (3.14) didapatkan parameter estimasi model  $\hat{\theta}$

$$\hat{\theta} = \begin{bmatrix} 0.1875 & -0.0352 \\ 1.6781 & 0.9436 \\ -0.2586 & 0.0626 \\ -0.4617 & -0.0395 \end{bmatrix}$$

Setelah nilai  $\hat{\theta}$  diketahui, maka didapatkan persamaan model ruang keadaan linier dari *Coupled-Tank Basic Process Rig 38-100* sebagai berikut

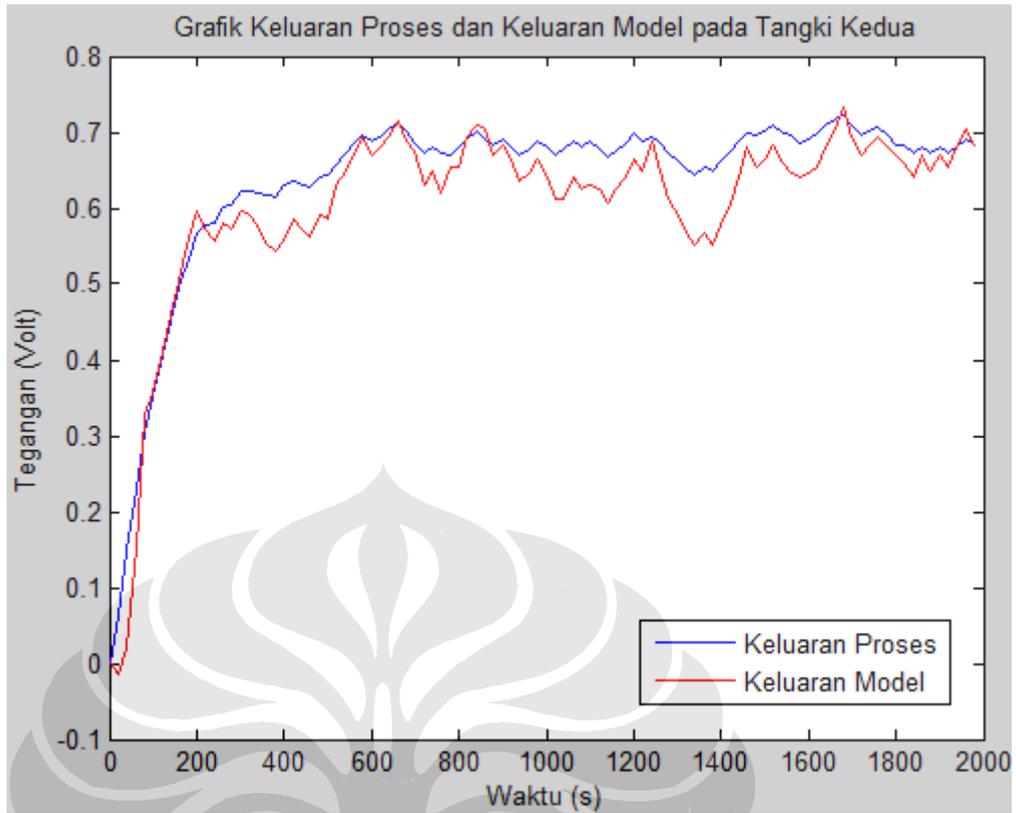
$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 0.1875 & 1.6781 \\ -0.0352 & 0.9436 \end{bmatrix}}_A \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \underbrace{\begin{bmatrix} -0.2586 \\ 0.0626 \end{bmatrix}}_B u(k) + \underbrace{\begin{bmatrix} -0.4617 \\ -0.0395 \end{bmatrix}}_K \quad (3.15)$$

Vektor  $\underline{K}$  pada persamaan (3.15) adalah vektor kompensasi nilai masukan.

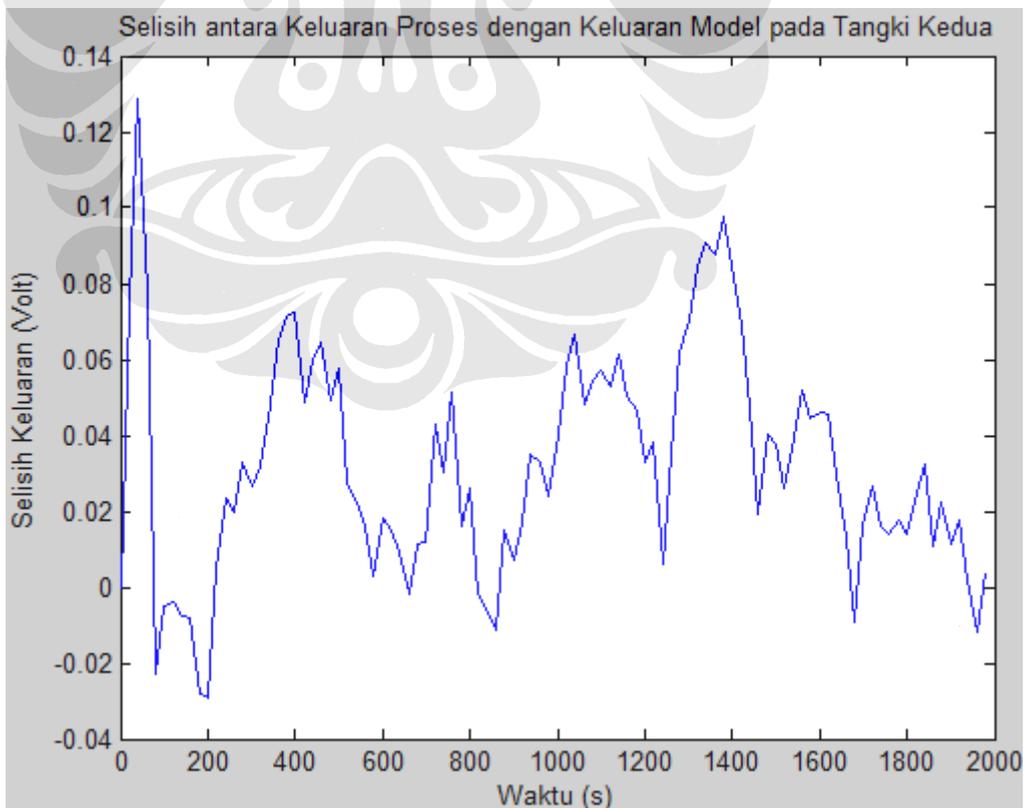
Karena keluaran sistem adalah ketinggian air pada tangki kedua, maka persamaan keluaran untuk model ruang keadaan adalah

$$y(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \quad (3.16)$$

Grafik keluaran proses dan keluaran model dapat dilihat pada gambar 3.8 sedangkan grafik selisih keluaran proses dan keluaran model pada tangki kedua dapat dilihat pada gambar 3.9.



Gambar 3.8. Grafik keluaran proses dan keluaran model pada tangki kedua.



Gambar 3.9. Selisih antara keluaran proses dengan keluaran model pada tangki kedua

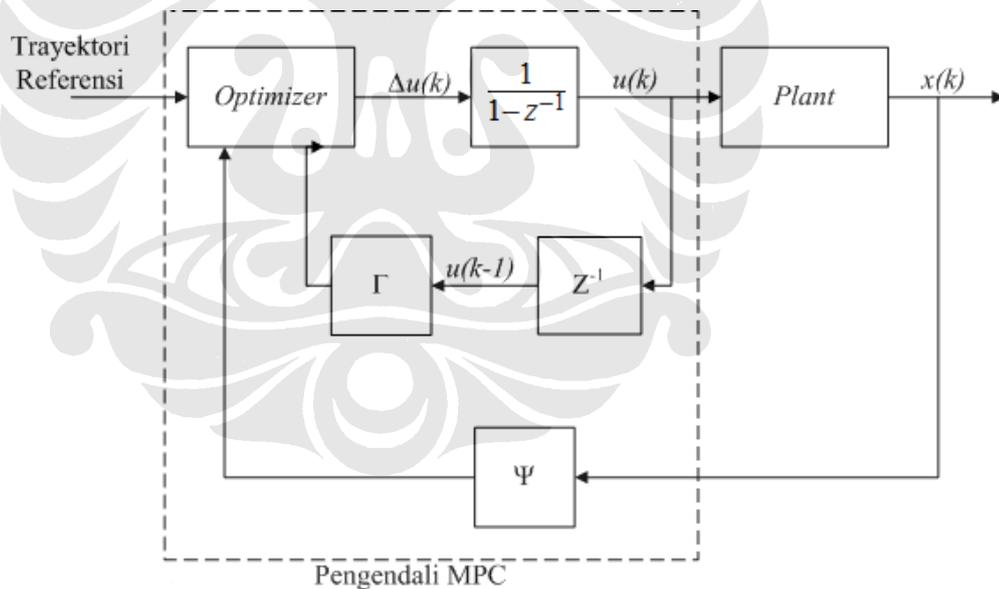
Dari gambar 3.8. dan gambar 3.9. terlihat bahwa model yang digunakan sudah cukup baik karena selisih antara keluaran proses dan keluaran model relatif kecil. Berdasarkan hasil estimasi, didapatkan nilai *lost function* ( $J_{LS}$ )

$$J_{LS} = \frac{1}{N} \sum_{i=1}^N (y_2 - \hat{y}_2)^2 = 0.00187997$$

Dari perhitungan ternyata didapatkan nilai  $J_{LS}$  yang cukup kecil, hal ini membuktikan bahwa model yang akan digunakan sudah cukup baik.

### 3.3 ALGORITMA MODEL PREDICTIVE CONTROL DENGAN CONSTRAINTS

Struktur pengendali MPC dengan *constraint* untuk model ruang keadaan terdapat pada gambar 3.8. Dari blok diagram tersebut, terlihat bahwa prediksi perubahan sinyal masukan sekarang ( $\Delta u(k)$ ) membutuhkan data dari variabel keadaan sekarang  $x(k)$  dan masukan satu langkah sebelumnya  $u(k-1)$ .



Gambar 3.10. Blok diagram pengendali MPC *with constraints*.

Algoritma perhitungan perubahan sinyal kendali pada MPC dengan *constraints* adalah sebagai berikut :

1. Parameter pengendali yang terlebih dahulu harus ditentukan antara lain horizon prediksi ( $H_p$ ), horizon kendali ( $H_u$ ), matriks faktor bobot kesalahan ( $Q$ ), dan matriks faktor bobot perubahan sinyal kendali ( $R$ ).

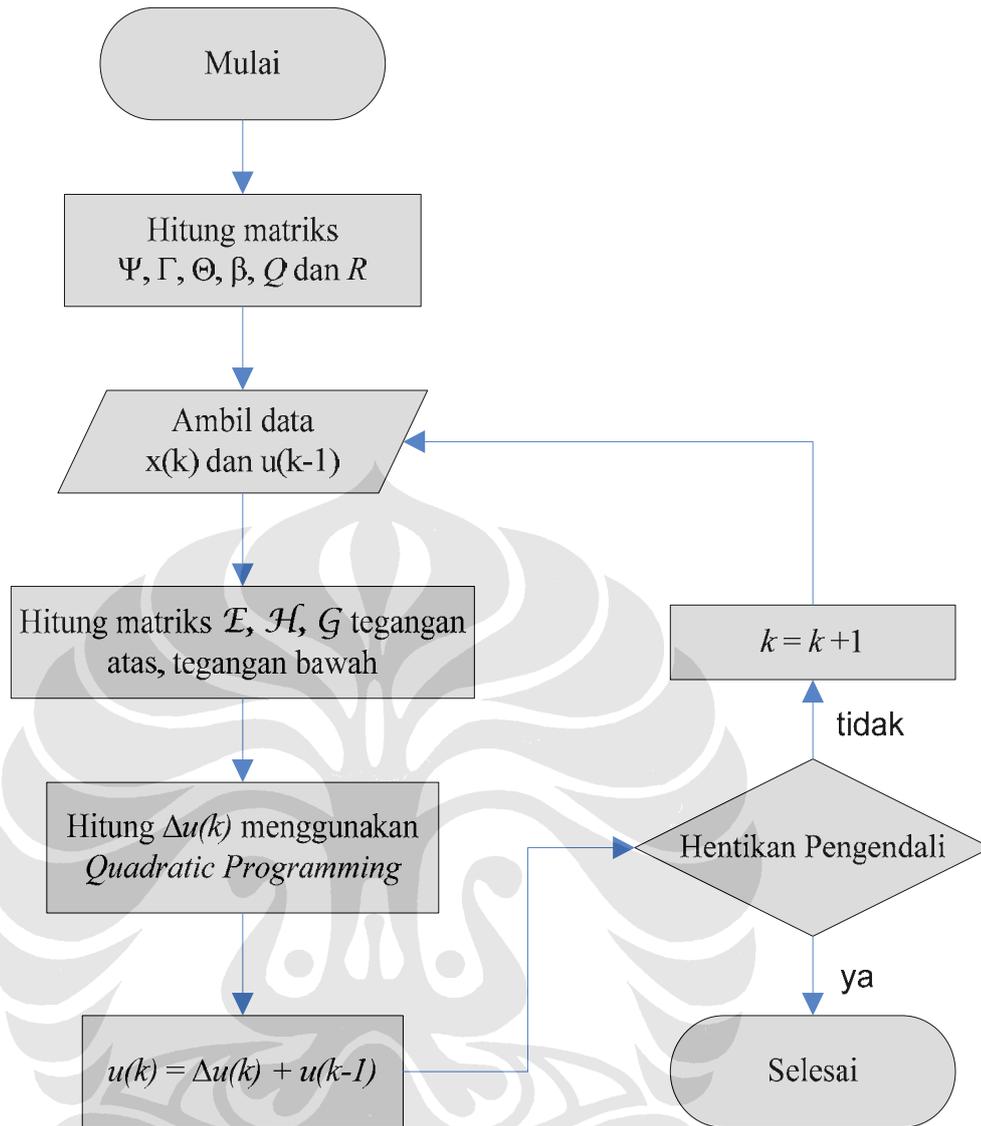
2. Matriks  $\underline{F}$  dihitung dengan menggunakan persamaan (2.33), serta matriks  $\underline{H}$  dan  $\underline{G}$  yang terdapat pada fungsi kriteria persamaan (2.37) dihitung masing-masing dengan menggunakan persamaan (2.39) dan (2.38).
3. Parameter batasan (*constraints*) fisik sistem diubah ke dalam bentuk pertidaksamaan yang memiliki hubungan dengan perubahan sinyal kendali ( $\Delta U$ ).

$$\underline{\Omega}\Delta U(k) \leq \underline{\omega} \quad (3.2)$$

4. Menghitung perubahan sinyal kendali optimal  $\Delta u_{opt}$  dengan menggunakan metode *Quadratic Programming*.
5. Menghitung sinyal kendali  $u(k)$  dimana

$$u(k) = \Delta u(k) + u(k-1) \quad (3.3)$$

Diagram alir untuk perhitungan sinyal kendali dengan menggunakan MPC dengan *constraints* adalah seperti pada gambar 3.11.



Gambar 3.11. Diagram alir algoritma MPC *with constraints*.

Metode yang digunakan pada *Quadratic Programming* dalam menghitung nilai  $\underline{\Delta U}$  adalah *Active Set* dengan alur operasi seperti dijelaskan berikut ini [2].

1. Fungsi kriteria pada persamaan (2.37), diubah menjadi seperti berikut

$$V(\underline{\Delta U}(k)) = \frac{1}{2} \underline{\Delta U}^T(k) 2\underline{\mathcal{H}} \underline{\Delta U}(k) - \underline{\Delta U}^T(k) \underline{\mathcal{G}} \quad (3.4)$$

berdasarkan *constraints*

$$\underline{\Omega} \underline{\Delta U}(k) \leq \underline{\omega} \quad (3.20)$$

2. Nilai  $\underline{\Delta U}_r$  dipilih sedemikian sehingga pertidaksamaan *constraints* (3.20) menjadi sebuah persamaan seperti berikut

$$\underline{\Omega}_r \underline{\Delta U}_r = \underline{w}_r \quad (3.21)$$

Elemen yang membuat pertidaksamaan menjadi persamaan disebut elemen aktif .

3. Menghitung nilai  $\underline{d}$  yang merupakan pergerakan  $\underline{\Delta U}_r$  dalam meminimasi fungsi kriteria sehingga fungsi kriteria pada persamaan (3.19) berubah menjadi

$$\begin{aligned} V(\underline{\Delta U}_r + \underline{d}) &= \frac{1}{2} (\underline{\Delta U}_r + \underline{d})^T 2\underline{\mathcal{H}}(\underline{\Delta U}_r + \underline{d}) - (\underline{\Delta U}_r + \underline{d})^T \underline{\mathcal{G}} \\ &= \frac{1}{2} \underline{d}^T \underbrace{2\underline{\mathcal{H}}}_{\underline{\Phi}} \underline{d} + \underline{d}^T \underbrace{(2\underline{\mathcal{H}} \underline{\Delta U}_r - \underline{\mathcal{G}})}_{\underline{\phi}_r} + V(\underline{\Delta U}_r) \end{aligned} \quad (3.22)$$

Nilai  $\underline{d}$  tidak boleh mempengaruhi pertidaksamaan *constraints* (3.13), sehingga persamaan *constraints* untuk persamaan (3.22) adalah

$$\underline{\Omega}_r \underline{d} = \underline{0} \quad (3.23)$$

4. Dari persamaan (3.22) dan (3.23), nilai optimal  $\underline{d}$  sepanjang *constraints* yang aktif dapat dihitung dengan menyelesaikan fungsi kuadratik berikut

$$\min \frac{1}{2} \underline{d}^T \underline{\Phi} \underline{d} + \underline{d}^T \underline{\phi}_r \quad (3.24)$$

dengan *constraints*

$$\underline{\Omega}_r \underline{d} = \underline{0} \quad (3.25)$$

Nilai pengali *Lagrange*  $\underline{\lambda}_r$  untuk persamaan (3.24) dan (3.25) dihitung berdasarkan kondisi *Karush-Kuhn-Tucker* (KKT) seperti berikut

$$\begin{bmatrix} \underline{\Phi} & \underline{\Omega}^T \\ \underline{\Omega} & \underline{0} \end{bmatrix} \begin{bmatrix} \underline{d} \\ \underline{\lambda}_r \end{bmatrix} = \begin{bmatrix} -\underline{\phi}_r \\ \underline{0} \end{bmatrix} \quad (3.26)$$

dimana nilai  $\underline{\Delta U}_r$  yang terdapat pada matriks  $\underline{\phi}_r$  ditentukan pada langkah (2).

Hasil perhitungan  $\underline{d}$  dan  $\underline{\lambda}_r$  akan mempengaruhi tahapan berikutnya, yaitu :

- a. Jika semua  $\underline{\lambda}_r > 0$  dan  $\underline{d} = 0$ , maka proses komputasi selesai dan nilai  $\underline{\Delta U}_r$  merupakan nilai optimal untuk  $\underline{\Delta U}(k)$ .
- b. Jika semua  $\underline{\lambda}_r > 0$  dan ada nilai  $d \neq 0$ , maka lanjut ke langkah (5).
- c. Jika ada nilai  $\lambda_r < 0$ , maka *constraint* yang memiliki nilai  $\lambda_r$  paling negatif dibuang, kemudian lanjut ke langkah (5).

5. Nilai faktor koreksi pergerakan nilai optimal  $\alpha_r$  dihitung dengan menggunakan rumus

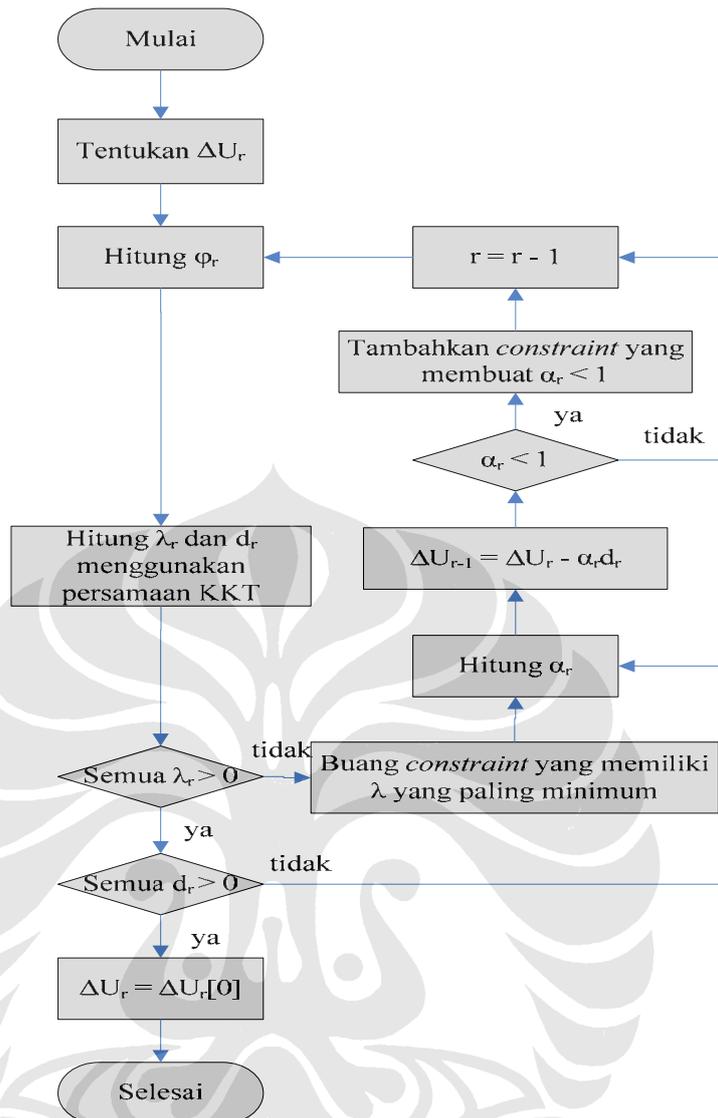
$$\alpha_r = \min \left( 1, \min_{\substack{i \in \Omega_r \\ a_i d > 0}} \frac{b_i - a_i \underline{\Delta U}_r}{a_i d} \right) \quad (3.5)$$

dengan  $a_i$  adalah baris dari pertidaksamaan batasan yang tidak aktif dan  $b_i$  adalah batasannya. Selanjutnya, nilai  $\underline{\Delta U}_r$  dalam arah  $\underline{d}$  dihitung sebagai berikut

$$\underline{\Delta U}_{r+1} = \underline{\Delta U}_r + \alpha_r \underline{d} \quad (3.6)$$

6. Jika nilai  $\alpha_r < 1$ , maka *constraint* yang membuat nilai  $\alpha_r < 1$  ditambahkan ke  $\underline{\Omega}_r$ .
7. Tetapkan  $r = r + 1$  dan kembali ke langkah (3) untuk proses iterasi berikutnya.

Diagram alir metode *Active Set* untuk menyelesaikan *Quadratic Programming* seperti yang terdapat pada gambar 3.12.



Gambar 3.12. Diagram alir metode *Active Set* untuk menyelesaikan *Quadratic Programming*.

### 3.4 PERHITUNGAN SINYAL KENDALI

Berikut ini adalah contoh perhitungan sinyal kendali dengan metode MPC dengan *constraints*. Spesifikasi pengendali yang digunakan pada pengendali MPC berikut ini adalah sebagai berikut :

- Nilai *control horizon*  $H_u = 2$
- Nilai *prediction horizon*  $H_p = 40$
- Faktor bobot kesalahan  $\underline{Q} = 11\underline{I}_{H_p}$
- Faktor bobot perubahan sinyal kendali  $\underline{R} = 12\underline{I}_{H_u}$
- Trayektori acuan  $r(k) = 0.8$

- Matriks variabel keadaan

$$\underline{A} = \begin{bmatrix} 0,989057 & -0,104357 \\ 0,787474 & 0,119354 \end{bmatrix}, \underline{B} = \begin{bmatrix} 0,270662 \\ 0,252388 \end{bmatrix}, \text{ dan } \underline{K} = \begin{bmatrix} -0,375062 \\ -0,298039 \end{bmatrix}$$

Karena pada persamaan ruang keadaan terdapat faktor kompensasi untuk nilai masukan, yakni vektor  $\underline{K}$ , maka perhitungan nilai prediksi variabel keadaan pada persamaan (2.16) berubah menjadi

$$\begin{bmatrix} \hat{x}(k+1|k) \\ \vdots \\ \hat{x}(k+Hu|k) \\ \vdots \\ \hat{x}(k+Hp|k) \end{bmatrix} = \underbrace{\begin{bmatrix} \underline{A} \\ \vdots \\ \underline{A}^{Hu} \\ \vdots \\ \underline{A}^{Hp} \end{bmatrix}}_{\underline{\Psi}} \underline{x}(k) + \underbrace{\begin{bmatrix} \underline{B} \\ \vdots \\ \sum_{i=0}^{Hu-1} \underline{A}^i \underline{B} \\ \vdots \\ \sum_{i=0}^{Hp-1} \underline{A}^i \underline{B} \end{bmatrix}}_{\underline{\Gamma}} \underline{u}(k-1) + \begin{bmatrix} \underline{K} \\ \vdots \\ \sum_{i=0}^{Hu-1} \underline{A}^i \underline{K} \\ \vdots \\ \sum_{i=0}^{Hp-1} \underline{A}^i \underline{K} \end{bmatrix} \quad (3.7)$$

$$\underbrace{\begin{bmatrix} \underline{B} & \cdots & \mathbf{0}_{n \times l} \\ \underline{A}\underline{B} + \underline{B} & \cdots & \vdots \\ \vdots & \cdots & \vdots \\ \sum_{i=0}^{Hp-1} \underline{A}^i \underline{B} & \cdots & \sum_{i=0}^{Hp-Hu} \underline{A}^i \underline{B} \end{bmatrix}}_{\underline{\Theta}} \begin{bmatrix} \Delta \hat{u}(k) \\ \vdots \\ \Delta \hat{u}(k+Hu-1) \end{bmatrix} + \underbrace{\begin{bmatrix} \underline{K} \\ \vdots \\ \sum_{i=0}^{Hu-1} \underline{A}^i \underline{K} \\ \vdots \\ \sum_{i=0}^{Hp-1} \underline{A}^i \underline{K} \end{bmatrix}}_{\underline{\beta}}$$

Contoh dari setiap tahap untuk menghitung sinyal kendali dengan algoritma *Model Predictive Control* dengan *constraints* adalah seperti berikut

1. Matriks  $\underline{C}_y \underline{\Psi}$ ,  $\underline{C}_y \underline{\Gamma}$ ,  $\underline{C}_y \underline{\Theta}$ , dan  $\underline{C}_y \underline{\beta}$  dihitung dengan menggunakan persamaan (3.22)

$$\underline{C}_y \underline{\Psi} = \begin{bmatrix} \underline{C} & \underline{0}_{1 \times 2} & \cdots & \underline{0}_{1 \times 2} \\ \underline{0}_{1 \times 2} & \underline{C} & \cdots & \underline{0}_{1 \times 2} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{0}_{1 \times 2} & \underline{0}_{1 \times 2} & \cdots & \underline{C} \end{bmatrix} \begin{bmatrix} \underline{A} \\ \vdots \\ \underline{A}^2 \\ \vdots \\ \underline{A}^{10} \end{bmatrix} = \begin{bmatrix} -0.0352 & 0.9436 & -0.0023 & 0.0431 \\ -0.0398 & 0.8313 & -0.0019 & 0.0368 \\ -0.0367 & 0.7176 & -0.0017 & 0.0315 \\ -0.0321 & 0.6155 & -0.0014 & 0.0269 \\ -0.0277 & 0.5268 & -0.0012 & 0.0230 \\ -0.0237 & 0.4507 & -0.0010 & 0.0197 \\ -0.0203 & 0.3854 & -0.0009 & 0.0168 \\ -0.0174 & 0.3296 & -0.0008 & 0.0144 \\ -0.0149 & 0.2818 & -0.0006 & 0.0123 \\ -0.0127 & 0.2410 & -0.0006 & 0.0105 \\ -0.0109 & 0.2061 & -0.0005 & 0.0090 \\ -0.0093 & 0.1762 & -0.0004 & 0.0077 \\ -0.0079 & 0.1507 & -0.0003 & 0.0066 \\ -0.0068 & 0.1289 & -0.0003 & 0.0056 \\ -0.0058 & 0.1102 & -0.0003 & 0.0048 \\ -0.0050 & 0.0942 & -0.0002 & 0.0041 \\ -0.0042 & 0.0806 & -0.0002 & 0.0035 \\ -0.0036 & 0.0689 & -0.0002 & 0.0030 \\ -0.0031 & 0.0589 & -0.0001 & 0.0026 \\ -0.0027 & 0.0504 & -0.0001 & 0.0022 \end{bmatrix}$$

$$\underline{C}_y \underline{\Gamma} = \begin{bmatrix} \underline{C} & \underline{0}_{1 \times 2} & \cdots & \underline{0}_{1 \times 2} \\ \underline{0}_{1 \times 2} & \underline{C} & \cdots & \underline{0}_{1 \times 2} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{0}_{1 \times 2} & \underline{0}_{1 \times 2} & \cdots & \underline{C} \end{bmatrix} \begin{bmatrix} \underline{B} \\ \vdots \\ \sum_{i=0}^1 \underline{A}^i \underline{B} \\ \vdots \\ \sum_{i=0}^9 \underline{A}^i \underline{B} \end{bmatrix} = \begin{bmatrix} 0.0626 & 0.0038 \\ 0.0682 & 0.0033 \\ 0.0623 & 0.0028 \\ 0.0544 & 0.0024 \\ 0.0468 & 0.0021 \\ 0.0401 & 0.0018 \\ 0.0343 & 0.0015 \\ 0.0294 & 0.0013 \\ 0.0251 & 0.0011 \\ 0.0215 & 0.0009 \\ 0.0184 & 0.0008 \\ 0.0157 & 0.0007 \\ 0.0134 & 0.0006 \\ 0.0115 & 0.0005 \\ 0.0098 & 0.0004 \\ 0.0084 & 0.0004 \\ 0.0072 & 0.0003 \\ 0.0061 & 0.0003 \\ 0.0053 & 0.0002 \\ 0.0045 & 0.0002 \end{bmatrix}$$

$$\underline{C}_y \underline{\Theta} = \begin{bmatrix} \underline{C} & \underline{0}_{1 \times 2} & \cdots & \underline{0}_{1 \times 2} \\ \underline{0}_{1 \times 2} & \underline{C} & \cdots & \underline{0}_{1 \times 2} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{0}_{1 \times 2} & \underline{0}_{1 \times 2} & \cdots & \underline{C} \end{bmatrix} \begin{bmatrix} \underline{B} & \cdots & \underline{0}_{2 \times 1} \\ \underline{AB} + \underline{B} & \ddots & \vdots \\ \vdots & \cdots & \vdots \\ \sum_{i=0}^9 \underline{A}^i \underline{B} & \cdots & \sum_{i=0}^8 \underline{A}^i \underline{B} \end{bmatrix} = \begin{bmatrix} 0.0626 & 0 & 0.0038 & 0.0045 \\ 0.0682 & 0.0626 & 0.0033 & 0.0038 \\ 0.0623 & 0.0682 & 0.0028 & 0.0033 \\ 0.0544 & 0.0623 & 0.0024 & 0.0028 \\ 0.0468 & 0.0544 & 0.0021 & 0.0024 \\ 0.0401 & 0.0468 & 0.0018 & 0.0021 \\ 0.0343 & 0.0401 & 0.0015 & 0.0018 \\ 0.0294 & 0.0343 & 0.0013 & 0.0015 \\ 0.0251 & 0.0294 & 0.0011 & 0.0013 \\ 0.0215 & 0.0251 & 0.0009 & 0.0011 \\ 0.0184 & 0.0215 & 0.0008 & 0.0009 \\ 0.0157 & 0.0184 & 0.0007 & 0.0008 \\ 0.0134 & 0.0157 & 0.0006 & 0.0007 \\ 0.0115 & 0.0134 & 0.0005 & 0.0006 \\ 0.0098 & 0.0115 & 0.0004 & 0.0005 \\ 0.0084 & 0.0098 & 0.0004 & 0.0004 \\ 0.0072 & 0.0084 & 0.0003 & 0.0004 \\ 0.0061 & 0.0072 & 0.0003 & 0.0003 \\ 0.0053 & 0.0061 & 0.0002 & 0.0003 \\ 0.0045 & 0.0053 & 0.0002 & 0.0002 \end{bmatrix}$$

$$\underline{C}_y \underline{\beta} = \begin{bmatrix} \underline{C} & \underline{0}_{1 \times 2} & \cdots & \underline{0}_{1 \times 2} \\ \underline{0}_{1 \times 2} & \underline{C} & \cdots & \underline{0}_{1 \times 2} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{0}_{1 \times 2} & \underline{0}_{1 \times 2} & \cdots & \underline{C} \end{bmatrix} \begin{bmatrix} \underline{K} \\ \vdots \\ \sum_{i=0}^1 \underline{A}^i \underline{K} \\ \vdots \\ \sum_{i=0}^9 \underline{A}^i \underline{K} \end{bmatrix} = \begin{bmatrix} -0.0395 & -0.0008 \\ -0.0210 & -0.0007 \\ -0.0145 & -0.0006 \\ -0.0114 & -0.0005 \\ -0.0095 & -0.0004 \\ -0.0080 & -0.0003 \\ -0.0068 & -0.0003 \\ -0.0058 & -0.0003 \\ -0.0050 & -0.0002 \\ -0.0043 & -0.0002 \\ -0.0037 & -0.0002 \\ -0.0031 & -0.0001 \\ -0.0027 & -0.0001 \\ -0.0023 & -0.0001 \\ -0.0020 & -0.0001 \\ -0.0017 & -0.0001 \\ -0.0014 & -0.0001 \\ -0.0012 & -0.0001 \\ -0.0010 & -0.0000 \\ -0.0009 & -0.0000 \end{bmatrix}$$

2. Batasan sinyal kendali dengan nilai *control horizon* sama dengan dua dapat dinyatakan dalam bentuk pertidaksamaan seperti berikut

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} 1,3 \leq \begin{bmatrix} u(k) \\ u(k+1) \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix} 3 \quad (3.30)$$

Pertidaksamaan (3.30) harus diubah kedalam bentuk pertidaksamaan yang mengandung  $\Delta u(k)$ , dimana

$$u(k) = \Delta u(k) + u(k-1) \text{ dan}$$

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

Sehingga didapatkan hasil transformasi pertidaksamaan (3.30) sebagai berikut

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} 1,3 \leq \begin{bmatrix} \Delta u(k) \\ \Delta u(k) + \Delta u(k+1) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k-1) \quad (3.31)$$

$$\begin{bmatrix} \Delta u(k) \\ \Delta u(k) + \Delta u(k+1) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k-1) \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix} 3 \quad (3.32)$$

Dengan menggeser semua elemen yang mengandung  $\Delta u$  ke sebelah kiri dan yang tidak mengandung  $\Delta u$  ke sebelah kanan tanda pertidaksamaan ( $\leq$ ), maka persamaan (3.31) dan (3.32) masing-masing dapat ditulis kembali menjadi

$$\begin{bmatrix} -\Delta u(k) \\ -\Delta u(k) - \Delta u(k+1) \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix} [-1,3 + u(k-1)] \quad (3.33)$$

$$\begin{bmatrix} \Delta u(k) \\ \Delta u(k) + \Delta u(k+1) \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix} [3 - u(k-1)] \quad (3.34)$$

atau

$$\underbrace{\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ -1 & 0 \\ -1 & -1 \end{bmatrix}}_{\underline{\Omega}} \underbrace{\begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \end{bmatrix}}_{\underline{\Delta U}} \leq \underbrace{\begin{bmatrix} 3 - u(k-1) \\ 3 - u(k-1) \\ -1,3 + u(k-1) \\ -1,3 + u(k-1) \end{bmatrix}}_{\underline{\omega}} \quad (3.35)$$

3. Matriks  $\underline{G}$ , dan  $\underline{H}$  masing-masing dihitung dengan menggunakan persamaan (2.38), dan (2.39). Dengan membuat nilai matriks  $\underline{Q}$  sama dengan  $11\underline{I}_{40}$  dan matriks  $\underline{R}$  bernilai  $12\underline{I}_2$ , maka matriks  $\underline{H}$  dapat dihitung sebagai berikut

$$\underline{H} = (\underline{C}_y \underline{\Theta})^T \underline{Q} \underline{C}_y \underline{\Theta} + \underline{R}$$

$$\underline{H} = \begin{bmatrix} 12.2597 & 0.2362 \\ 0.2362 & 12.2597 \end{bmatrix}$$

sedangkan nilai matriks  $\underline{G}$  untuk  $k = 1$  adalah

$$\underline{G} = 2(\underline{\Theta} \underline{C}_y)^T \underline{Q} \underline{E}(1)$$

$$\underline{G} = \begin{bmatrix} 10.1965 \\ 10.1459 \end{bmatrix}$$

Perhitungan untuk matriks  $\underline{E}$  agak berbeda dengan persamaan (2.33) dimana

$$\underline{E}(k) = \underline{T}(k) - \underline{C}_y \underline{\Psi} \underline{x}(k) - \underline{C}_y \underline{\Gamma} \underline{u}(k-1) - \underline{C}_y \underline{\beta} \quad (3.36)$$

Dengan memisalkan pada  $k = 1$   $\underline{x}(1) = [0 \ 0]^T$ ,  $\underline{u}(0) = 0$  dan

$$\underline{T}(1) = \begin{bmatrix} 0.8000 & 0.8000 \\ 0.8000 & 0.8000 \\ 0.8000 & 0.8000 \\ 0.8000 & 0.8000 \\ 0.8000 & 0.8000 \\ 0.8000 & 0.8000 \\ 0.8000 & 0.8000 \\ 0.8000 & 0.8000 \\ 0.8000 & 0.8000 \\ 0.8000 & 0.8000 \\ 0.8000 & 0.8000 \\ 0.8000 & 0.8000 \\ 0.8000 & 0.8000 \\ 0.8000 & 0.8000 \\ 0.8000 & 0.8000 \\ 0.8000 & 0.8000 \\ 0.8000 & 0.8000 \\ 0.8000 & 0.8000 \\ 0.8000 & 0.8000 \\ 0.8000 & 0.8000 \end{bmatrix}$$

maka didapatkan

$$\underline{E}(1) = \begin{bmatrix} 0.8395 & 0.8008 \\ 0.8210 & 0.8007 \\ 0.8145 & 0.8006 \\ 0.8114 & 0.8005 \\ 0.8095 & 0.8004 \\ 0.8080 & 0.8003 \\ 0.8068 & 0.8003 \\ 0.8058 & 0.8003 \\ 0.8050 & 0.8002 \\ 0.8043 & 0.8002 \\ 0.8037 & 0.8002 \\ 0.8031 & 0.8001 \\ 0.8027 & 0.8001 \\ 0.8023 & 0.8001 \\ 0.8020 & 0.8001 \\ 0.8017 & 0.8001 \\ 0.8014 & 0.8001 \\ 0.8012 & 0.8001 \\ 0.8010 & 0.8000 \\ 0.8009 & 0.8000 \end{bmatrix}$$

4. Menghitung nilai  $\underline{d}$  dan  $\underline{\lambda}_r$  sebagai berikut :

- a. Nilai matriks  $\underline{\Omega}_1$  dan  $\underline{\Delta U}_1$  dipilih sedemikian rupa sehingga isi matriks  $\underline{\Delta U}_1$  membuat pertidaksamaan  $\underline{\Omega}_1 \underline{\Delta U} \leq \underline{\omega}_1$  menjadi aktif dan memenuhi persamaan

$$\underline{\Omega}_1 \underline{\Delta U}_1 = \underline{\omega}_1 \quad (3.8)$$

Matriks  $\underline{\Omega}_1$  yang dipilih adalah

$$\underline{\Omega}_1 = [1 \quad 0],$$

Untuk menentukan nilai matriks  $\underline{\omega}_1$ , ada beberapa hal yang harus diperhatikan, yaitu :

- i. Jika selisih antara batas tegangan maksimum dan nilai sinyal masukan sebelumnya ( $u(k-1)$ ) lebih besar dari *slew rate* maksimumnya ( $\Delta u_{maks}$ ), maka batas tegangan maksimum  $u_{maks}$  harus diubah menjadi

$$u_{maks} = u(k-1) + \Delta u_{maks} \quad (3.9)$$

- ii. Hal yang sama juga berlaku untuk batas tegangan minimum dimana ketika selisih antara nilai sinyal masukan sebelumnya ( $u(k-1)$ ) dengan nilai batas tegangan minimum lebih besar daripada *slew rate* maksimumnya ( $\Delta u_{maks}$ ), maka batas tegangan minimum  $u_{min}$  harus diubah menjadi

$$u_{min} = u(k-1) - \Delta u_{maks} \quad (3.10)$$

Besarnya *slew rate* maksimum pada percobaan ini adalah satu ( $\Delta u_{maks} = 1$ ), sehingga isi matriks  $\underline{\omega}_1$  yang memenuhi kedua syarat di atas dan bersesuaian dengan matriks  $\underline{\Omega}_1$  adalah

$$\underline{\omega}_1 = [1 - u(0)] = [1]$$

Supaya persamaan (3.37) terpenuhi, maka isi matriks  $\underline{\Delta U}_1$  yang harus digunakan adalah

$$\underline{\Delta U}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

b. Nilai  $\underline{d}$  dan  $\underline{\lambda}_r$  dihitung dengan menggunakan persamaan

$$\begin{bmatrix} \underline{d} \\ \underline{\lambda}_1 \end{bmatrix} = \begin{bmatrix} \underline{\Phi} & \underline{\Omega}_1^T \\ \underline{\Omega}_1 & \underline{0} \end{bmatrix}^{-1} \begin{bmatrix} -\underline{\phi}_1 \\ \dots \\ \underline{0} \end{bmatrix} \quad (3.40)$$

dengan  $\underline{\Phi}$  dan  $\underline{\phi}$  merupakan bagian dari persamaan (3.15), maka

$$\begin{bmatrix} \underline{d} \\ \underline{\lambda}_1 \end{bmatrix} = \begin{bmatrix} 24.5194 & 0.4724 & 1.0000 \\ 0.4724 & 24.5194 & 0 \\ 1.0000 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} -14.3229 \\ 9.6735 \\ \dots \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \underline{d} \\ \underline{\lambda}_1 \end{bmatrix} = \begin{bmatrix} -0.0000 \\ 0.3945 \\ \dots \\ -14.5093 \end{bmatrix}$$

Karena semua  $\underline{\lambda}_1$  bernilai negatif maka *constraint* yang memiliki nilai  $\underline{\lambda}_1$  paling negatif harus dibuang.

c. Nilai  $\alpha_1$  dihitung dengan menggunakan persamaan (3.27) dan didapatkan nilai  $\alpha_1$  sama dengan nol. Karena nilai  $\alpha_1$  kurang dari satu ( $\alpha_1 < 1$ ) maka ada *constraint* yang membuat nilai  $\alpha_1$  menjadi nol ditambahkan ke matriks  $\underline{\Omega}_1$ . *Constraint* yang ditambahkan ke matriks  $\underline{\Omega}_1$  adalah *constraint* yang terdapat pada baris kedua dari matriks  $\underline{\Omega}$  pada persamaan (3.35). Selanjutnya, matriks  $\underline{\Omega}_1$  berubah menjadi matriks  $\underline{\Omega}_2$ , dimana

$$\underline{\Omega}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

sedangkan isi matriks  $\underline{\Delta U}_2$  adalah seperti berikut

$$\underline{\Delta U}_2 = \underline{\Delta U}_1 + \alpha_1 \underline{d} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

- d. Dengan mengulang langkah (4.b), maka proses perhitungan untuk mendapatkan nilai  $\underline{d}$  dan  $\underline{\lambda}_2$  yang baru adalah

$$\begin{bmatrix} \underline{d} \\ \underline{\lambda}_2 \end{bmatrix} = \begin{bmatrix} 24.5194 & 0.4724 & 1.0000 & 1.0000 \\ 0.4724 & 24.5194 & 0 & 1.0000 \\ 1.0000 & 0 & 0 & 0 \\ 1.0000 & 1.0000 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 24.5194 \\ 0.4724 \\ \dots\dots\dots \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \underline{d} \\ \underline{\lambda}_2 \end{bmatrix} = \begin{bmatrix} 0.0000 \\ -0.0000 \\ \dots\dots\dots \\ 24.0470 \\ 0.4724 \end{bmatrix}$$

Karena semua nilai  $\underline{d}$  sama dengan nol dan semua isi matriks  $\underline{\lambda}_2$  lebih besar dari nol, maka proses perhitungan selesai dan nilai  $\underline{\Delta U}_2$  adalah nilai optimal yang membuat fungsi kriteria pada persamaan (3.19) menjadi minimum.

5. Nilai  $\Delta u(k)$  yang digunakan untuk memperbarui sinyal kendali hanya nilai pada baris pertama matriks  $\underline{\Delta U}$  sedangkan isi baris yang lainnya dibuang karena pada proses pencuplikan berikutnya sudah didapatkan nilai  $\Delta u(k)$  yang baru. Dari contoh perhitungan pada langkah (4), maka nilai  $u(k)$  yang harus diberikan ke *plant* adalah sebagai berikut :

$$u(1) = \Delta u(1) + u(0)$$

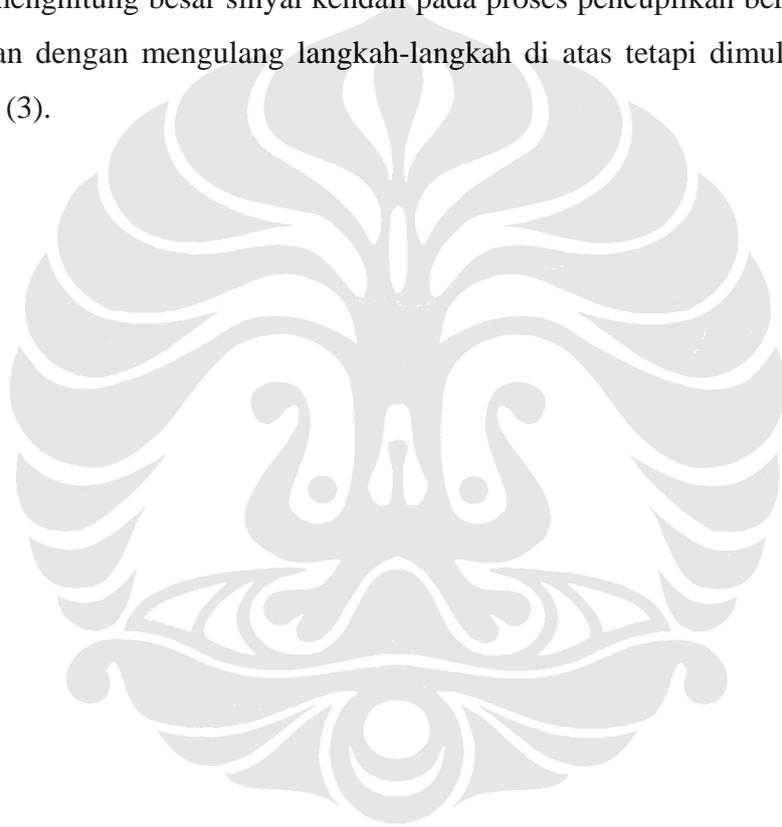
dengan

$$\underline{\Delta U}_{opt} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ dan } \Delta u(k) = \underline{\Delta U}_{opt} [1,1] = 1$$

maka

$$u(1) = 1 + 0 = 1 \text{ Volt}$$

Untuk menghitung besar sinyal kendali pada proses pencuplikan berikutnya dapat dilakukan dengan mengulang langkah-langkah di atas tetapi dimulai hanya dari langkah (3).



## BAB 4

### UJI EKSPERIMEN DAN ANALISA

Bab ini membahas analisa dari uji eksperimen pengendalian *Basic Process Rig 38-100* dengan metode MPC *with constraint* dengan beberapa parameter penalaan yang berbeda. Uji eksperimen dilakukan dengan menggunakan bantuan SIMULINK yang terdapat pada perangkat lunak MATLAB 7.

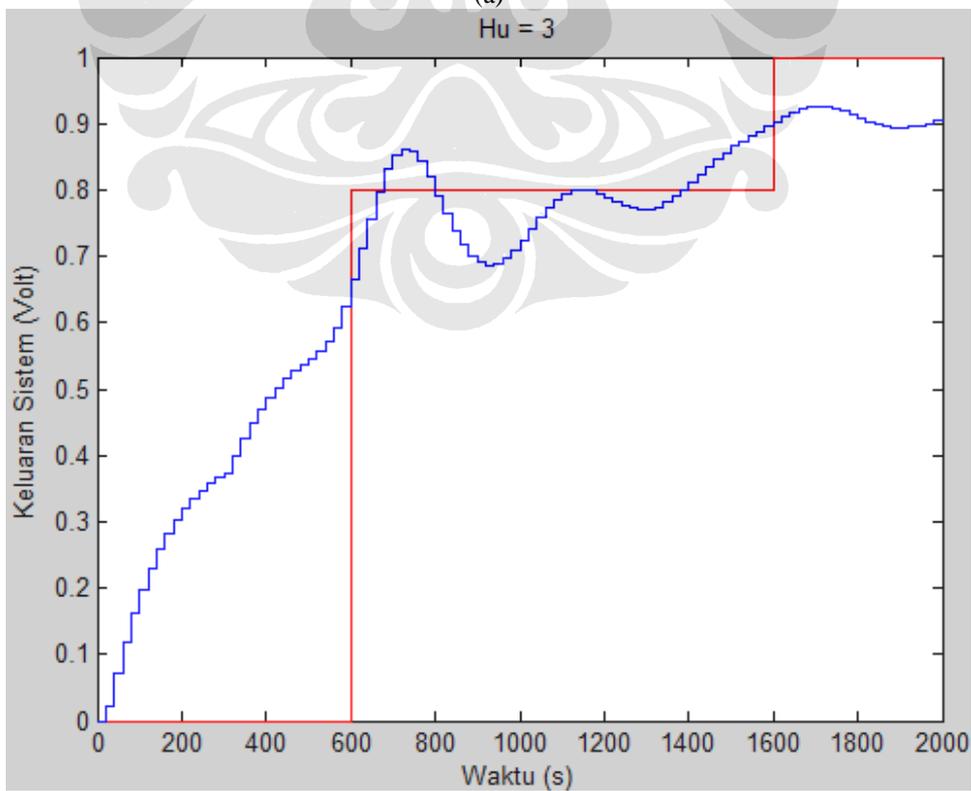
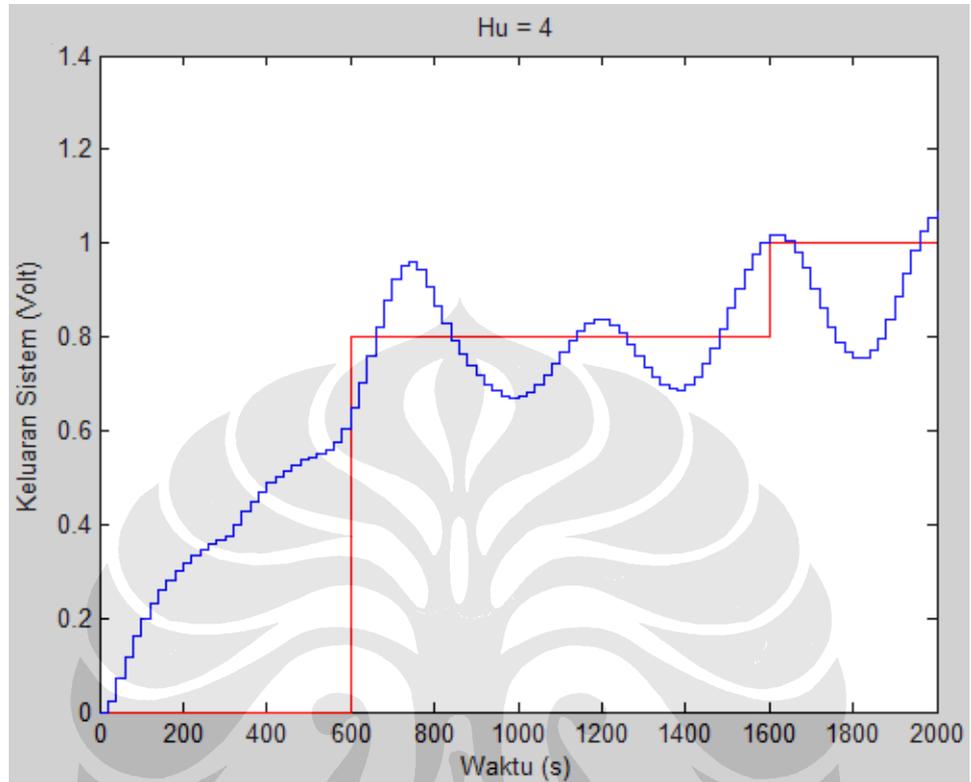
Tujuan dari uji eksperimen ini adalah untuk mengetahui kinerja MPC *with constraint* dengan parameter penalaan yang berbeda-beda. Selain membahas analisa hasil pengendali dengan menggunakan MPC *with constraint*, pada bab ini juga dibahas perbandingan antara hasil pengendali MPC *with constraint* dengan metode Aturan Kendali Ruang Keadaan.

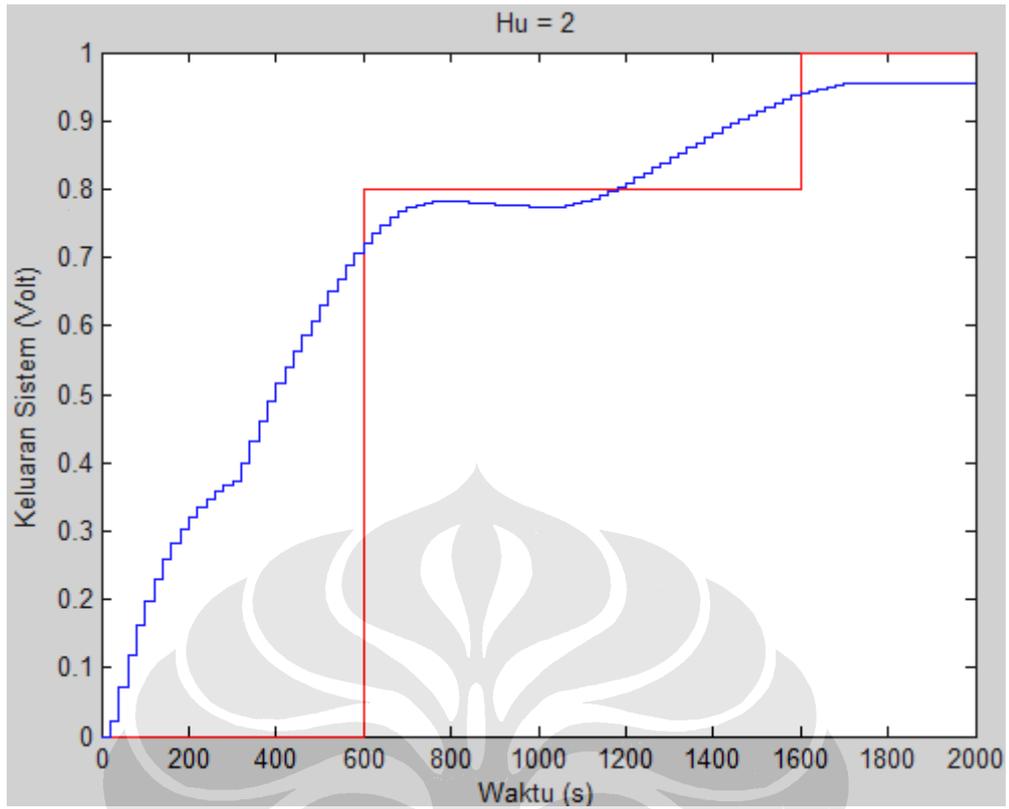
Sebelum dilakukan pengendali dengan menggunakan alat yang sebenarnya, terlebih dahulu dilakukan uji simulasi dengan menggunakan model proses. Model yang digunakan pada simulasi adalah model hasil identifikasi yang terdapat pada persamaan (3.15). Model tersebut kemudian disimulasikan dengan bantuan perangkat lunak MATLAB. Seluruh blok SIMULINK yang digunakan pada skripsi ini terdapat pada lampiran. Blok SIMULINK untuk uji simulasi dan uji eksperimen dengan pengendali MPC *with constraint* masing-masing terdapat pada gambar L.1 dan gambar L.2, sedangkan blok SIMULINK untuk uji eksperimen dengan metode Aturan Kendali Ruang Keadaan terdapat pada gambar L.3.

#### 4.1 PENGARUH NILAI *CONTROL HORIZON* PADA HASIL PENGENDALIAN MPC *WITH CONSTRAINT*

Untuk melihat pengaruh *control horizon* ( $H_u$ ) terhadap hasil pengendalian MPC, dilakukan uji simulasi dan uji eksperimen dengan nilai *prediction horizon* ( $H_p$ ) dibuat tetap, yaitu sebesar 30, dan nilai *control horizon* dibuat bervariasi. Variasi *control horizon* yang digunakan pada uji eksperimen adalah sebesar 2, 3, dan 4. Faktor bobot kesalahan ( $Q$ ) yang digunakan adalah  $L_{H_p}$  sedangkan faktor bobot perubahan sinyal kendali ( $R$ ) yang digunakan adalah  $L_{H_u}$ .

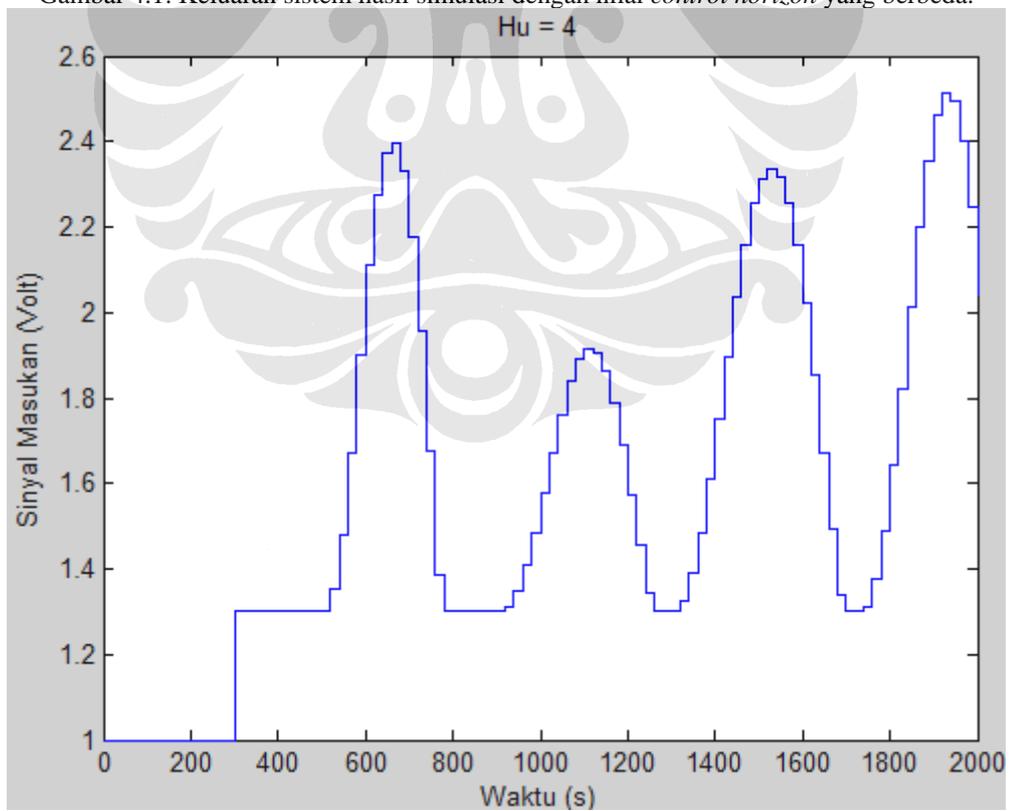
Hasil uji simulasi dengan *control horizon* yang berbeda dan parameter penalaan lainnya tetap terdapat pada gambar 4.1, gambar 4.2 dan gambar 4.3.



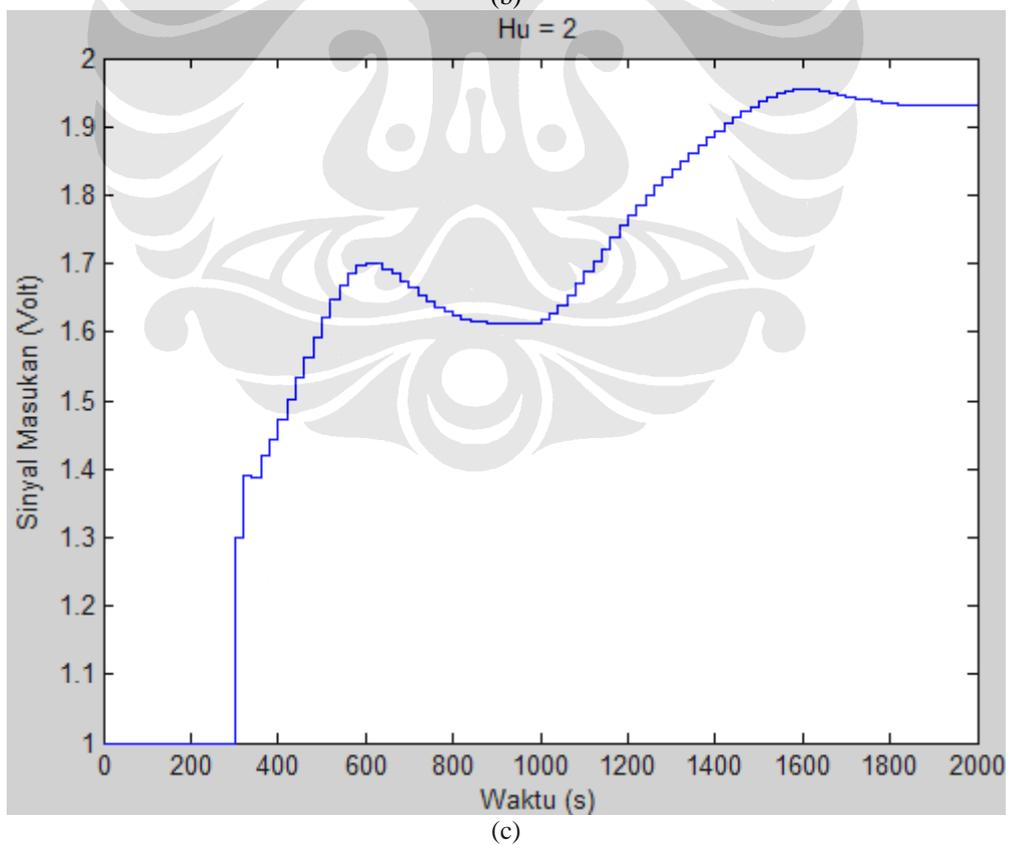
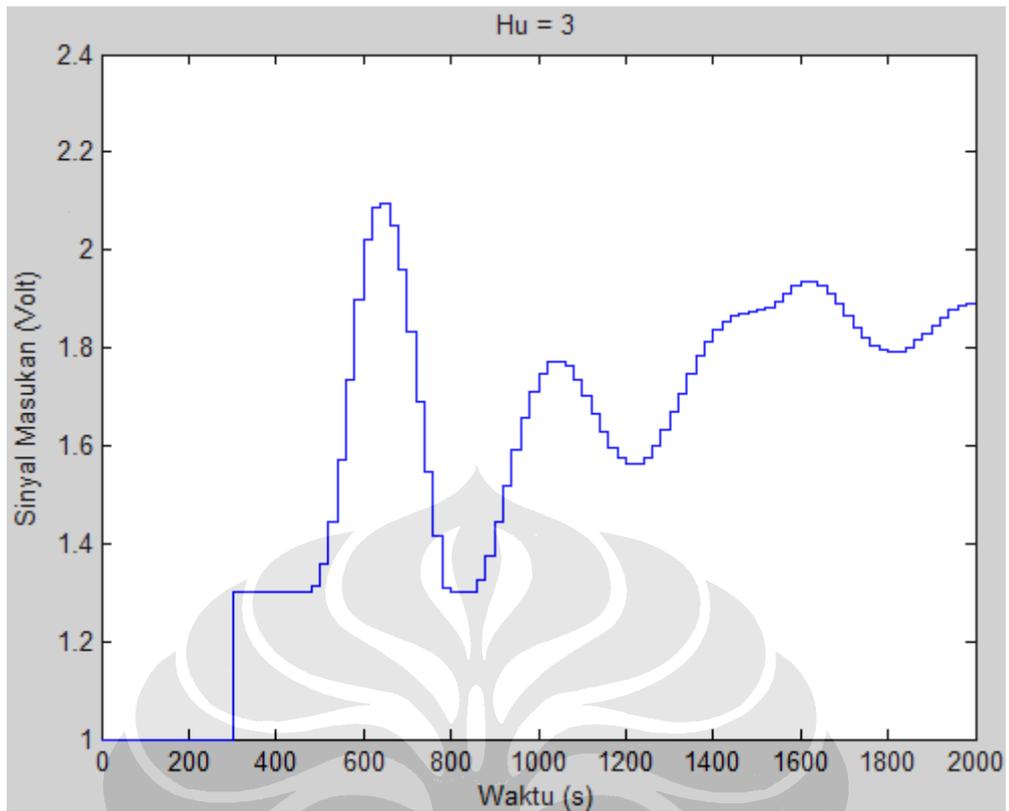


(c)

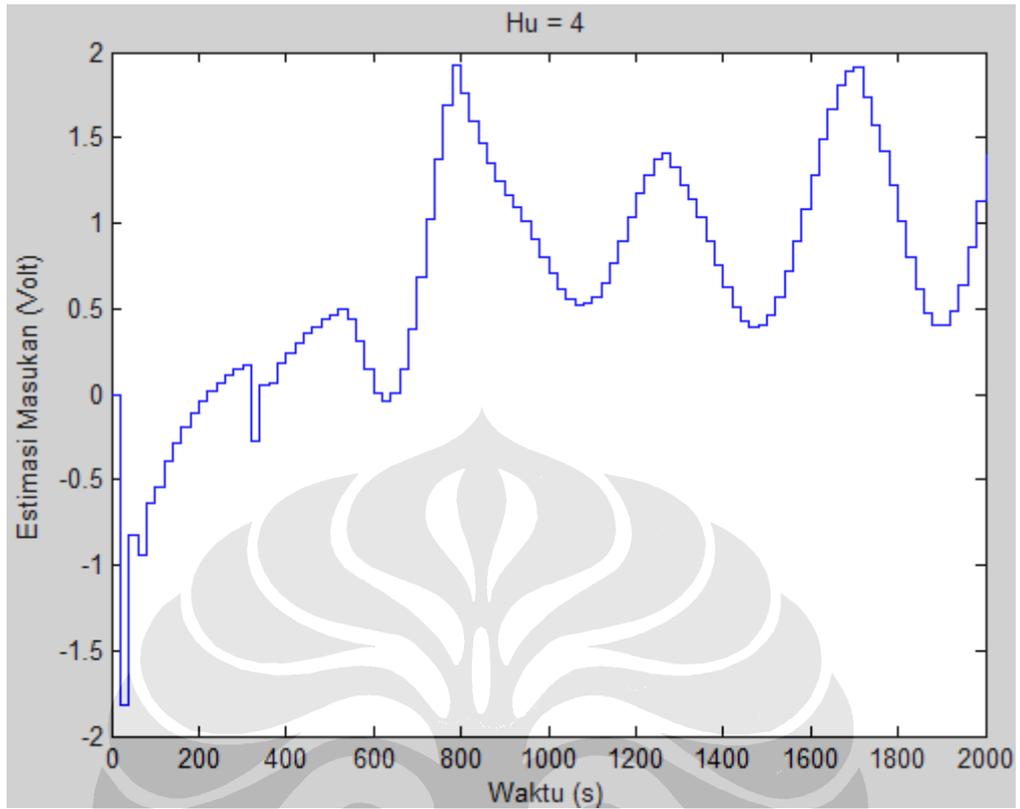
Gambar 4.1. Keluaran sistem hasil simulasi dengan nilai *control horizon* yang berbeda.



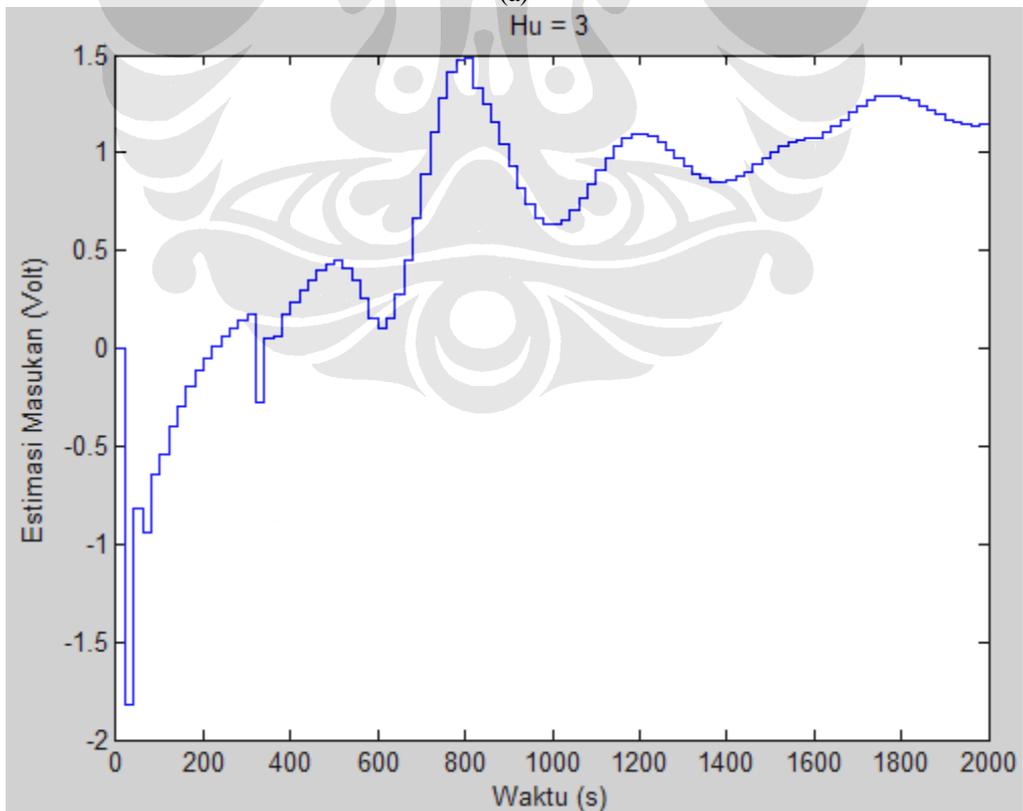
(a)



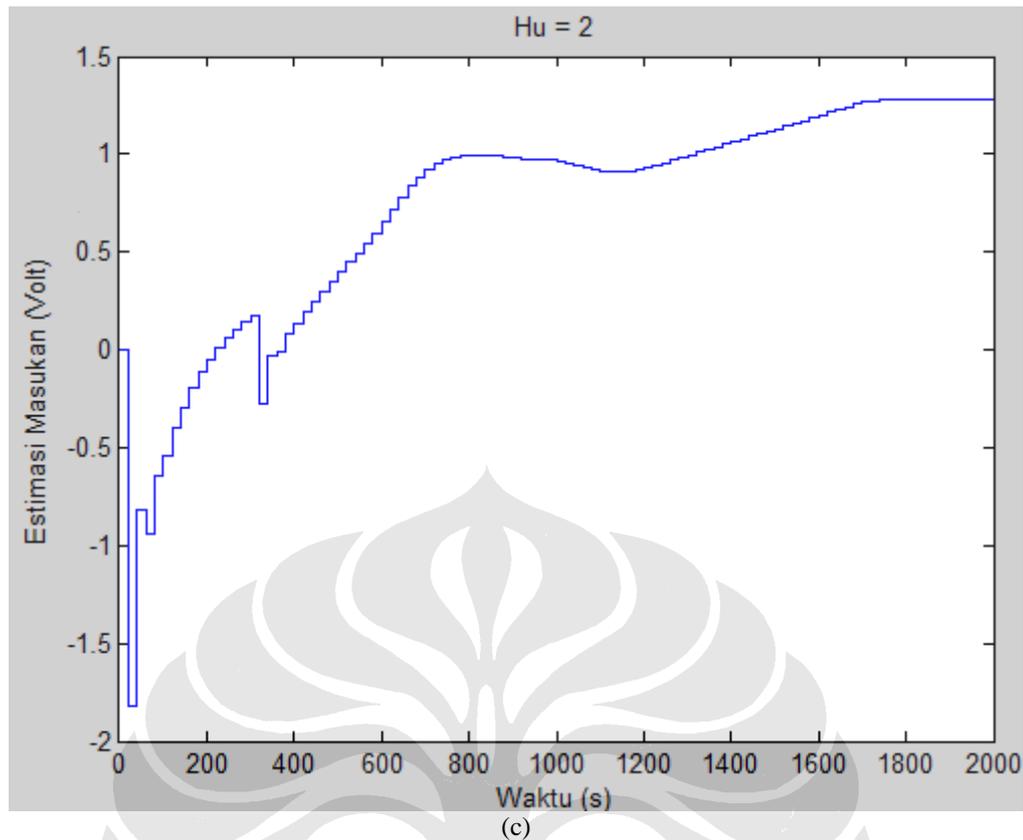
Gambar 4.2. Sinyal kendali hasil uji simulasi dengan nilai *control horizon* yang berbeda.



(a)



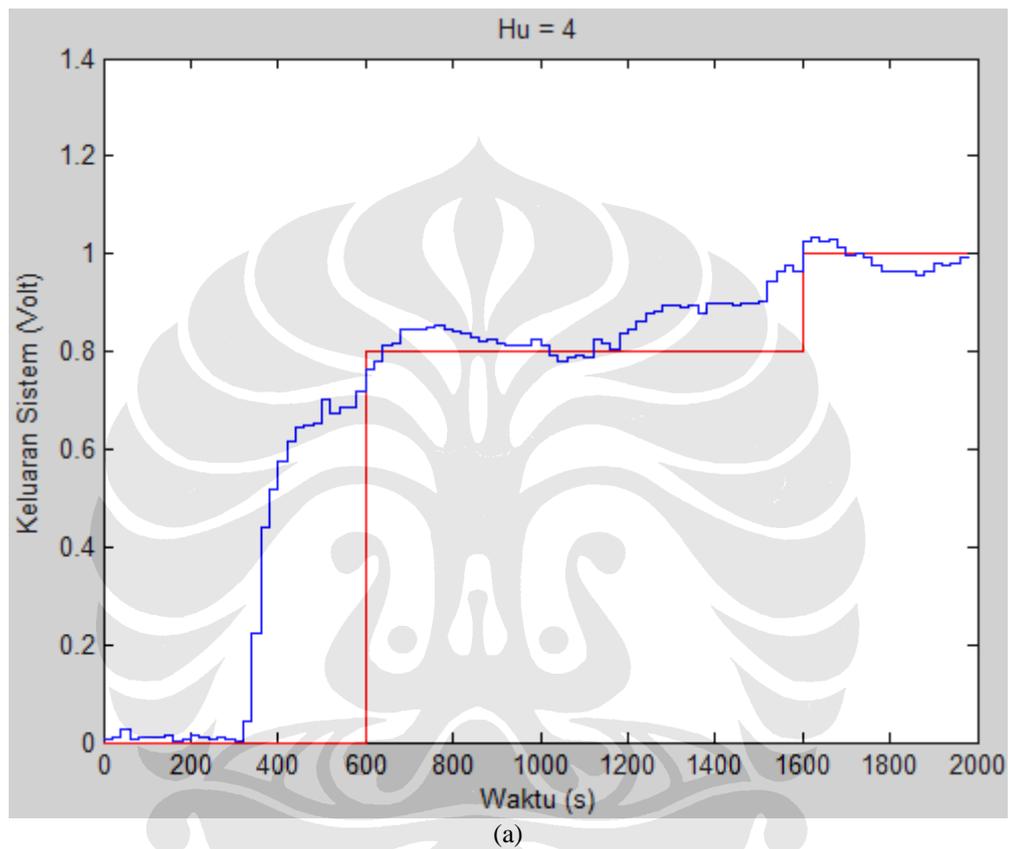
(b)

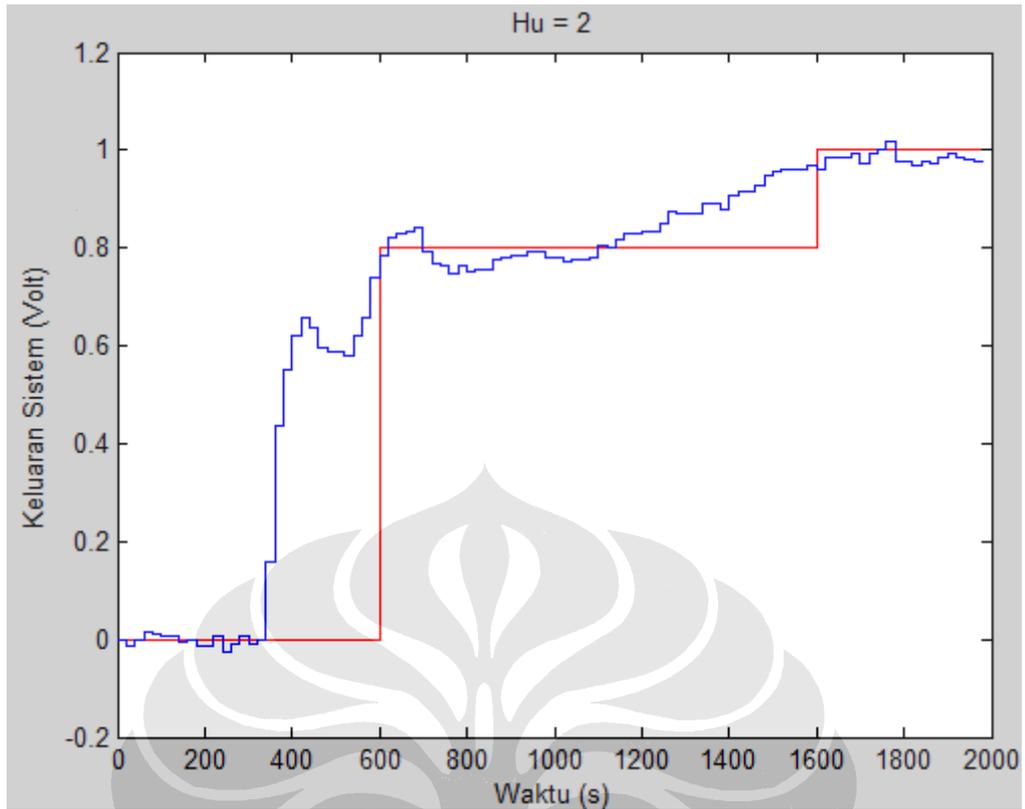


(c)  
Gambar 4.3. Estimasi masukan hasil simulasi dengan nilai *control horizon* yang berbeda.

Berdasarkan gambar 4.1, ternyata hasil pengendalian terlihat bagus jika besarnya *control horizon* menjauhi nilai *prediction horizon*. Jika nilai *control horizon* besar, maka sinyal kendali selalu ingin berubah sebesar-besarnya supaya keluaran sistem dapat dengan segera mencapai nilai yang sama dengan nilai trayektori acuan, sehingga perubahan sinyal kendali akan memiliki variansi yang besar pada saat akan terjadi perubahan nilai trayektori acuan (gambar 4.2.a). Perubahan sinyal kendali dengan variansi yang besar tersebut menyebabkan keluaran sistem menjadi tidak bagus (gambar 4.1.a). Semakin dekat nilai *control horizon* dengan nilai *prediction horizon*, variansi perubahan sinyal kendali akan semakin kecil. Hal tersebut dapat terlihat pada saat akan terjadi perubahan trayektori acuan. Dengan nilai *control horizon* yang hampir sama dengan nilai *prediction horizon*, prediksi perubahan sinyal kendali menyesuaikan dengan nilai prediksi keluaran sehingga variansi perubahan sinyal kendali tidak terlalu besar (gambar 4.2.b dan gambar 4.2.c).

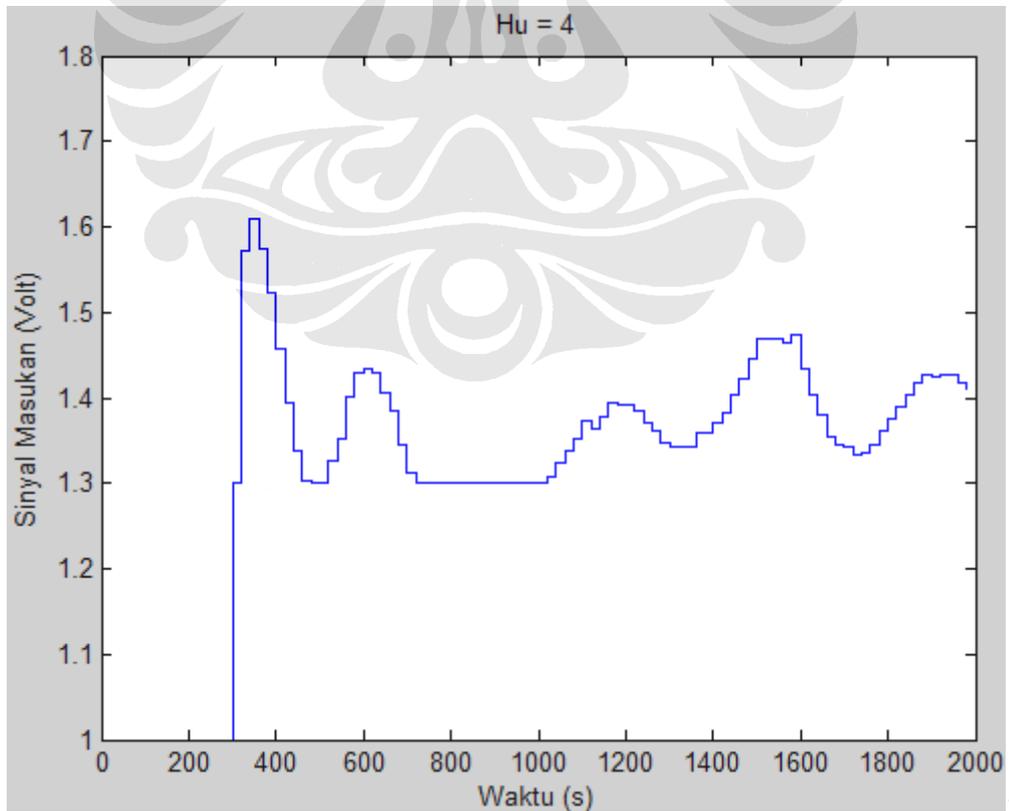
Untuk mempertegas hasil uji simulasi, dilakukan uji eksperimen pada *Basic Process Rig 38-100* untuk nilai *control horizon* yang berbeda, yaitu 2 dan 4, sedangkan nilai *prediction horizon* tetap, yaitu 30. Dari hasil uji eksperimen pada gambar 4.3 terlihat bahwa keluaran sistem akan semakin bagus jika nilai *control horizon* dibuat menjauhi nilai *prediction horizon*.



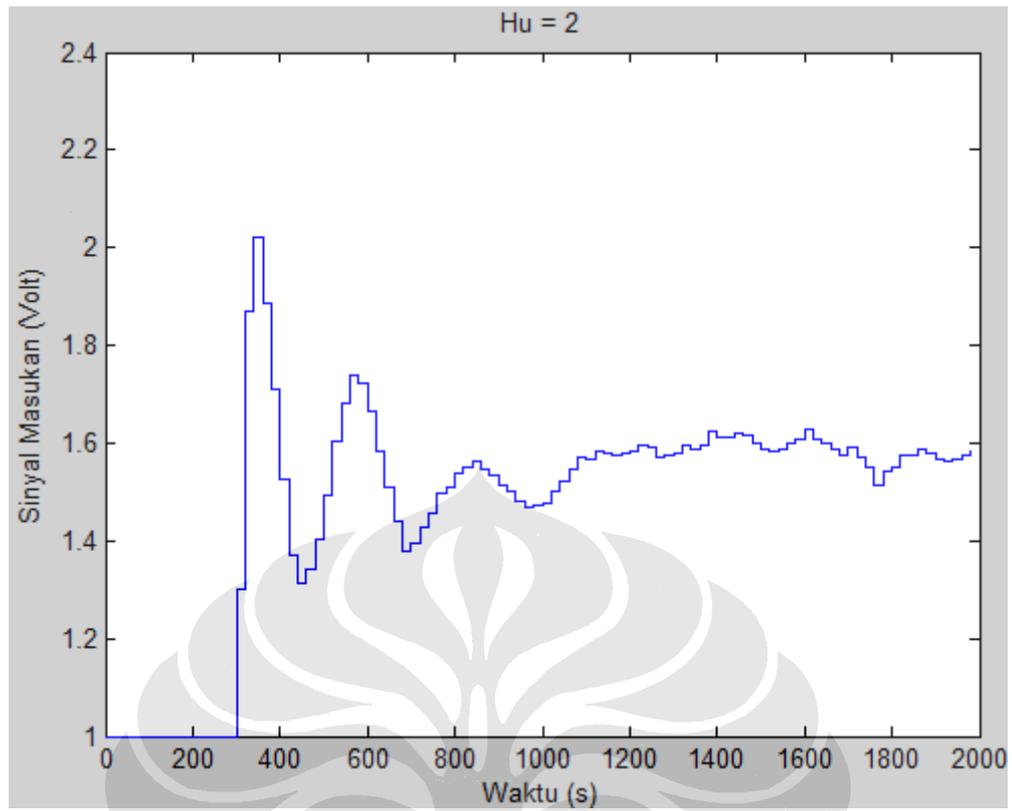


(b)

Gambar 4.4. Keluaran sistem hasil uji eksperimen dengan nilai *control horizon* yang berbeda.

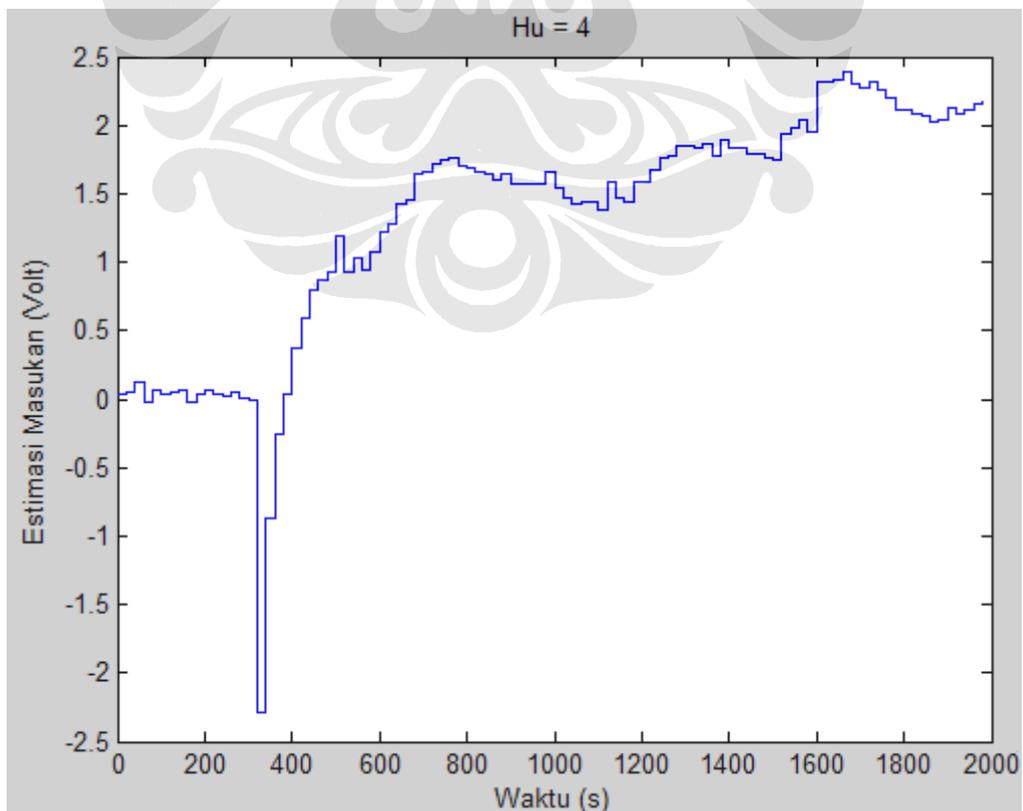


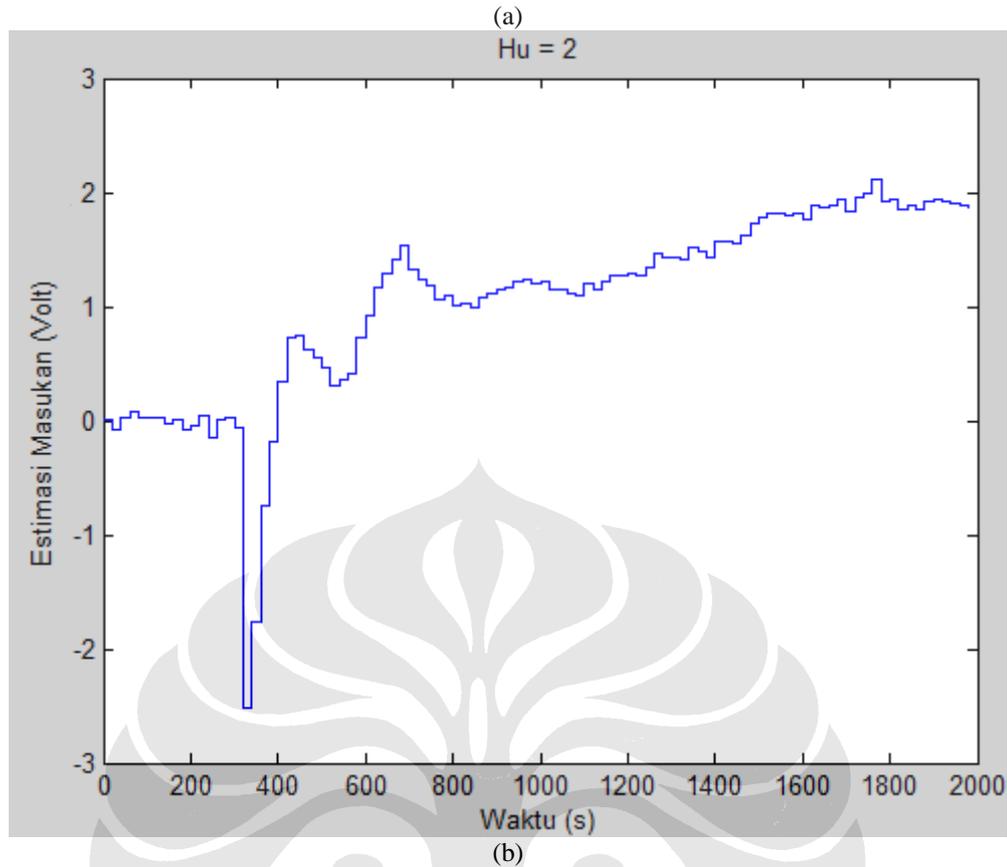
(a)



(b)

Gambar 4.5. Sinyal masukan hasil uji eksperimen dengan nilai *control horizon* yang berbeda.





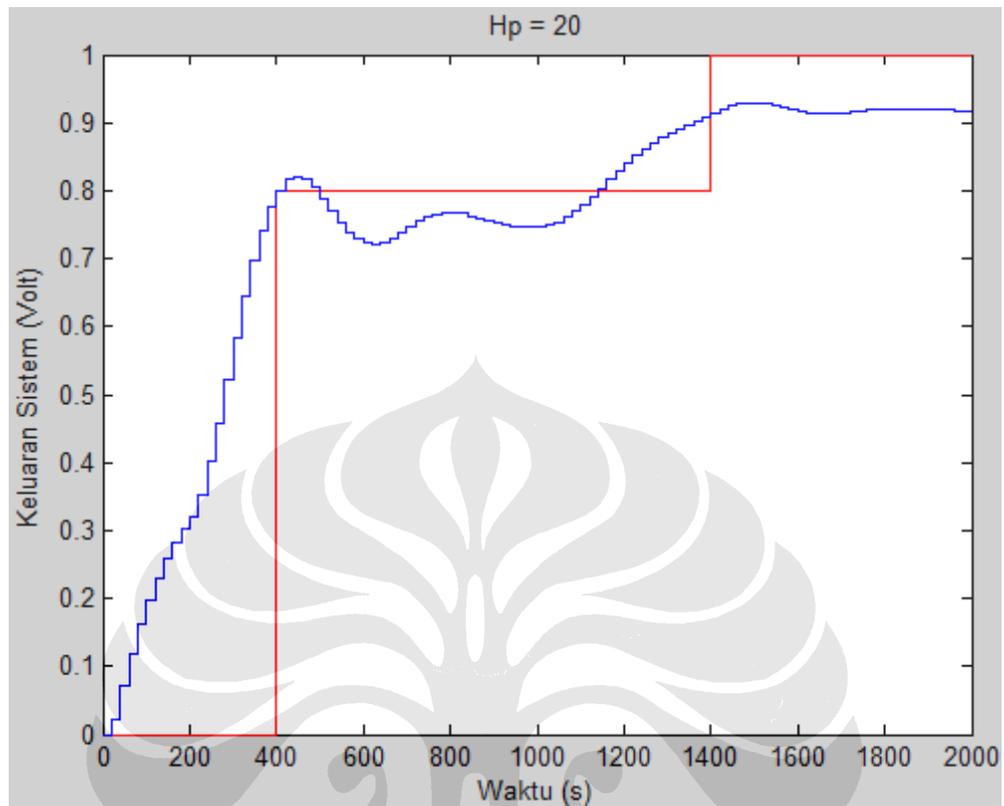
Gambar 4.6. Estimasi masukan hasil uji eksperimen dengan nilai *control horizon* yang berbeda.

#### 4.2 PENGARUH NILAI *PREDICTION HORIZON* PADA HASIL PENGENDALIAN MPC WITH CONSTRAINT

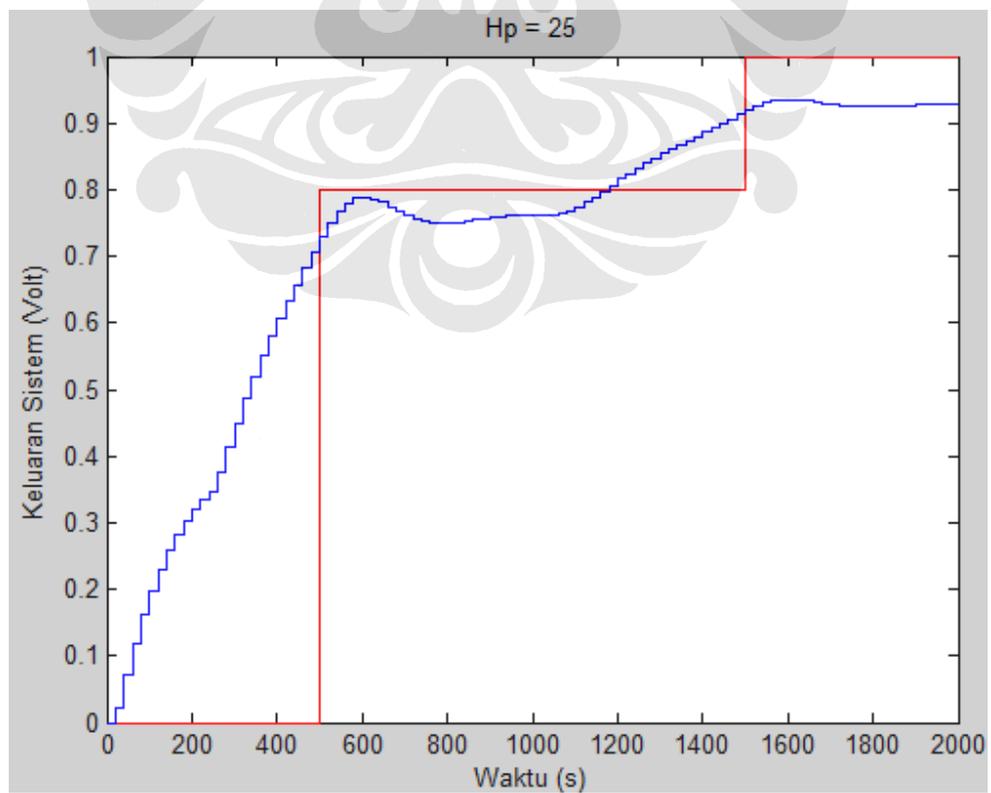
Untuk melihat pengaruh *prediction horizon* ( $H_p$ ) terhadap hasil pengendalian MPC with constraint, dilakukan uji simulasi dan uji eksperimen dengan nilai *control horizon* ( $H_u$ ) dibuat tetap, yaitu sebesar 2, dan nilai *prediction horizon* dibuat bervariasi. Variasi *prediction horizon* yang digunakan pada uji simulasi adalah sebesar 20, 25, dan 30. Faktor bobot kesalahan ( $Q$ ) yang digunakan adalah  $I_{H_p}$  sedangkan faktor bobot perubahan sinyal kendali ( $R$ ) yang digunakan adalah  $I_{H_u}$ .

Hasil uji simulasi dengan parameter penalaan yang tetap dan *prediction horizon* yang berbeda terdapat pada gambar 4.7, gambar 4.8 dan gambar 4.9. Dari hasil uji simulasi terlihat bahwa untuk nilai faktor bobot perubahan sinyal kendali  $R$  yang kecil, maka hasil pengendali akan tetap bagus jika nilai *prediction horizon* dibuat lebih besar nilainya atau nilainya menjauhi besar dari *control horizon*. Keluaran dari sistem akan stabil bila besar dari *prediction horizon* mencukupi [3],

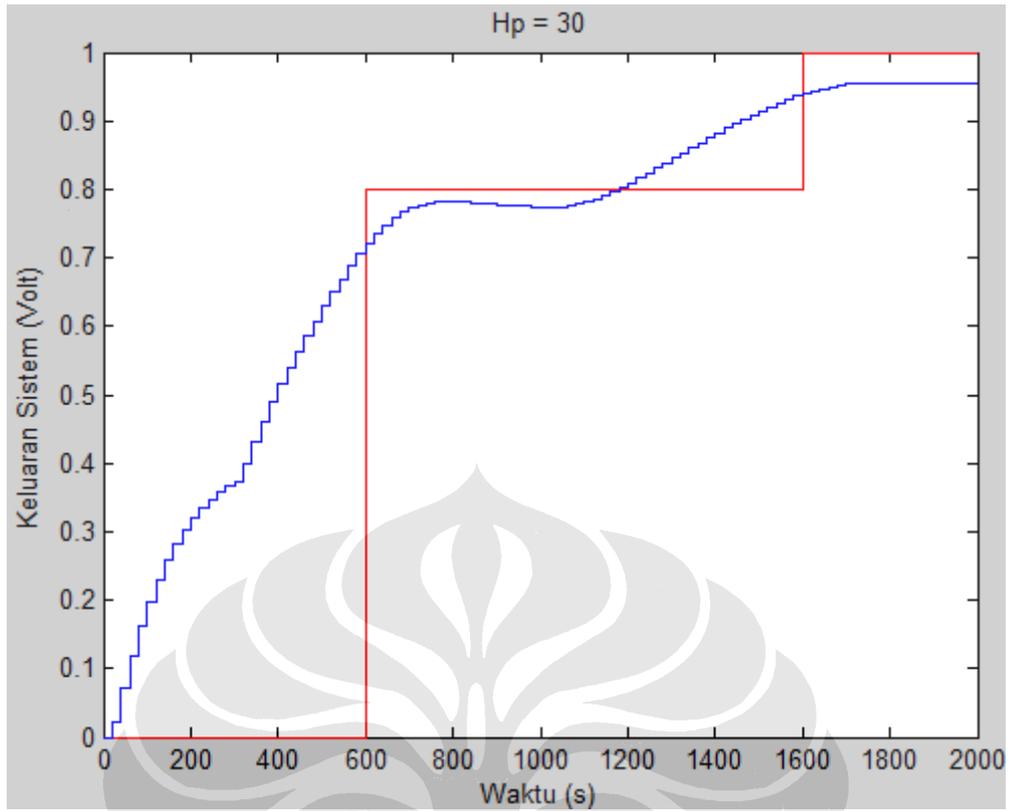
maka variansi perubahan sinyal kendali saat akan terjadi perubahan trayektori acuan semakin kecil.



(a)

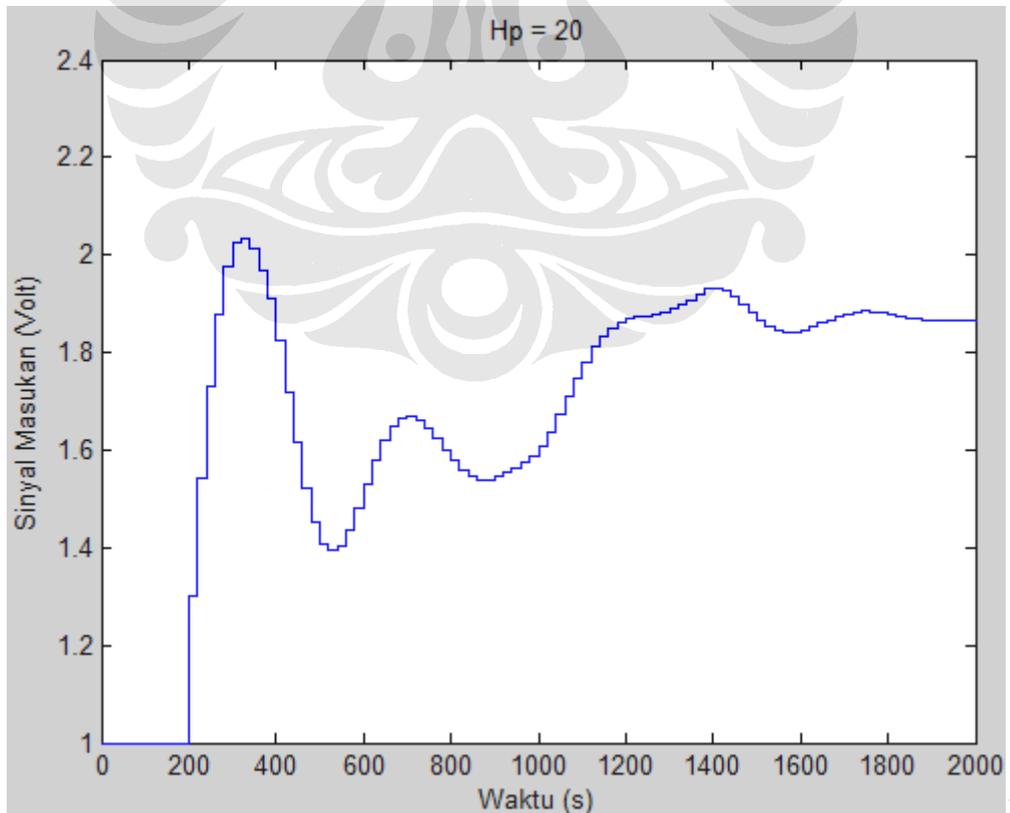


(b)

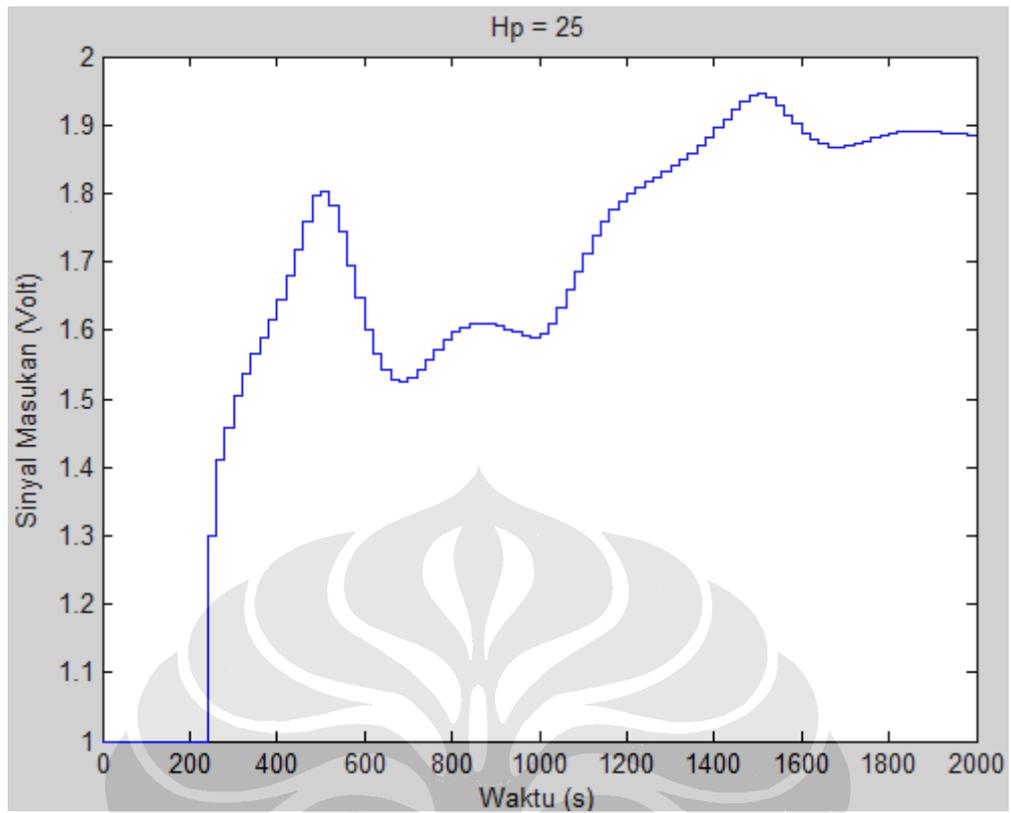


(c)

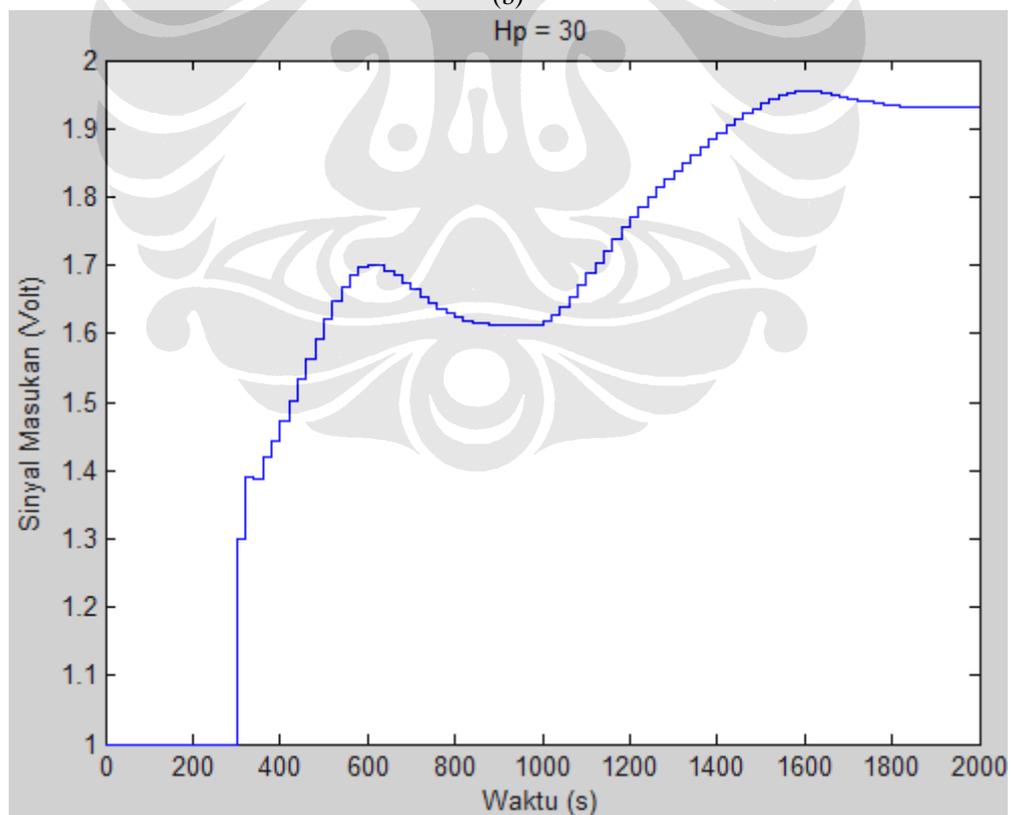
Gambar 4.7. Keluaran sistem hasil uji simulasi dengan nilai *prediction horizon* yang berbeda.



(a)

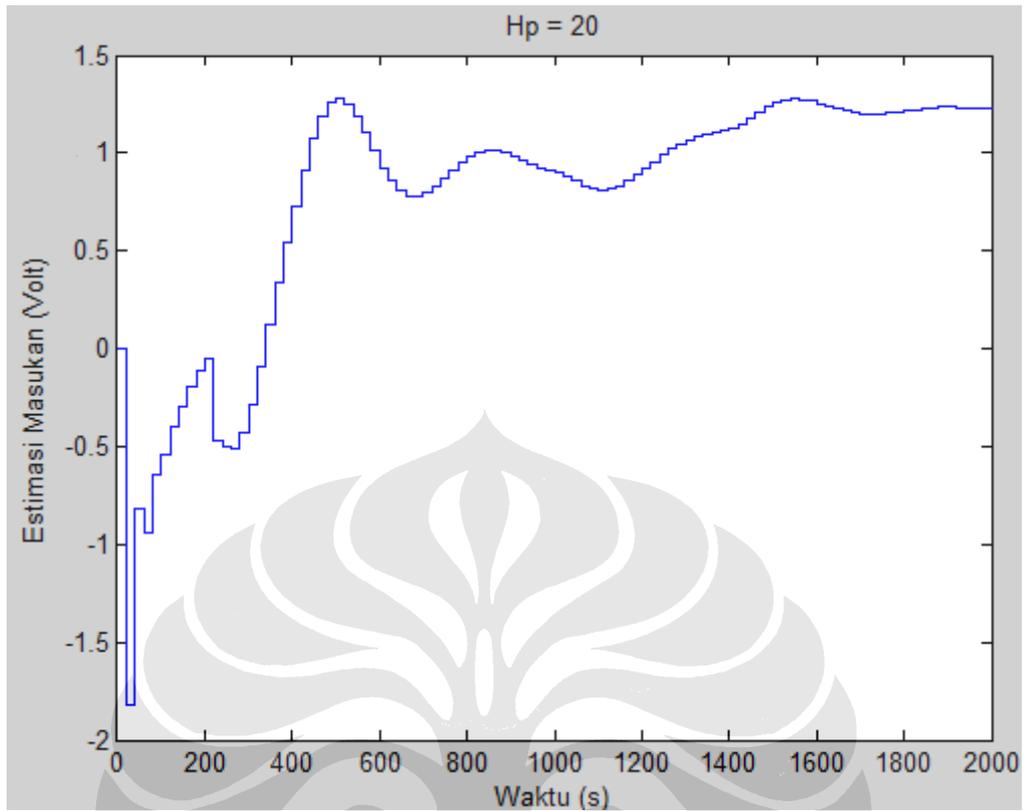


(b)

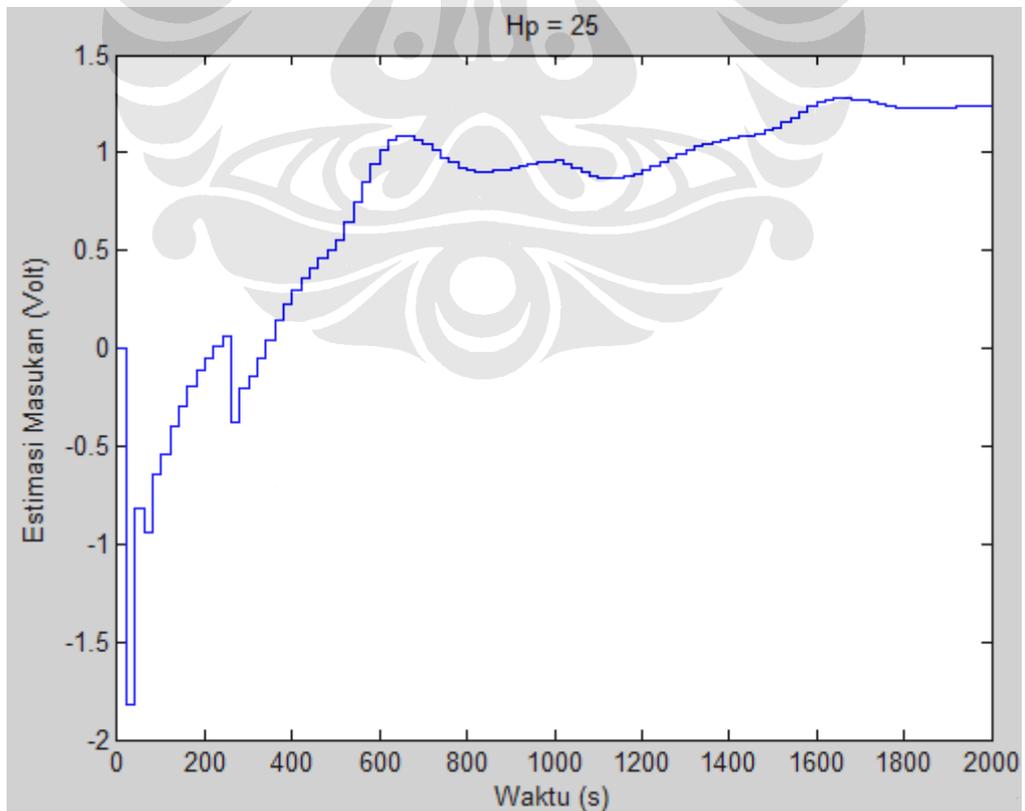


(c)

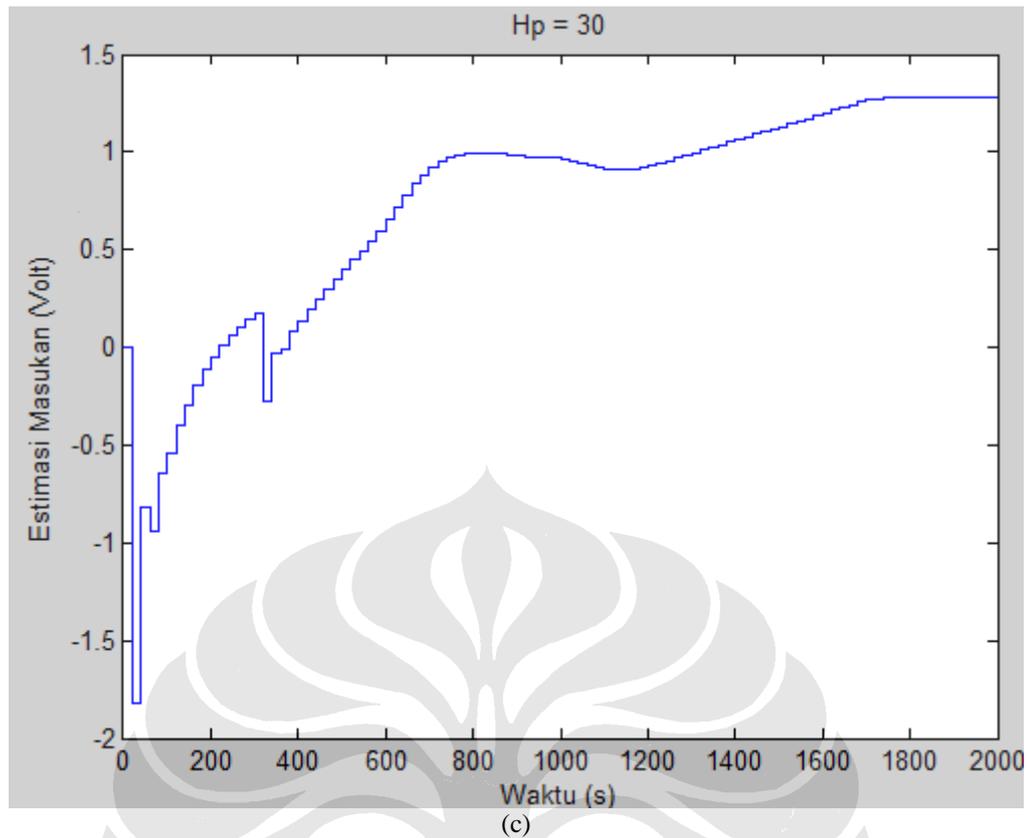
Gambar 4.8. Sinyal masukan hasil uji simulasi dengan nilai *prediction horizon* yang berbeda.



(a)

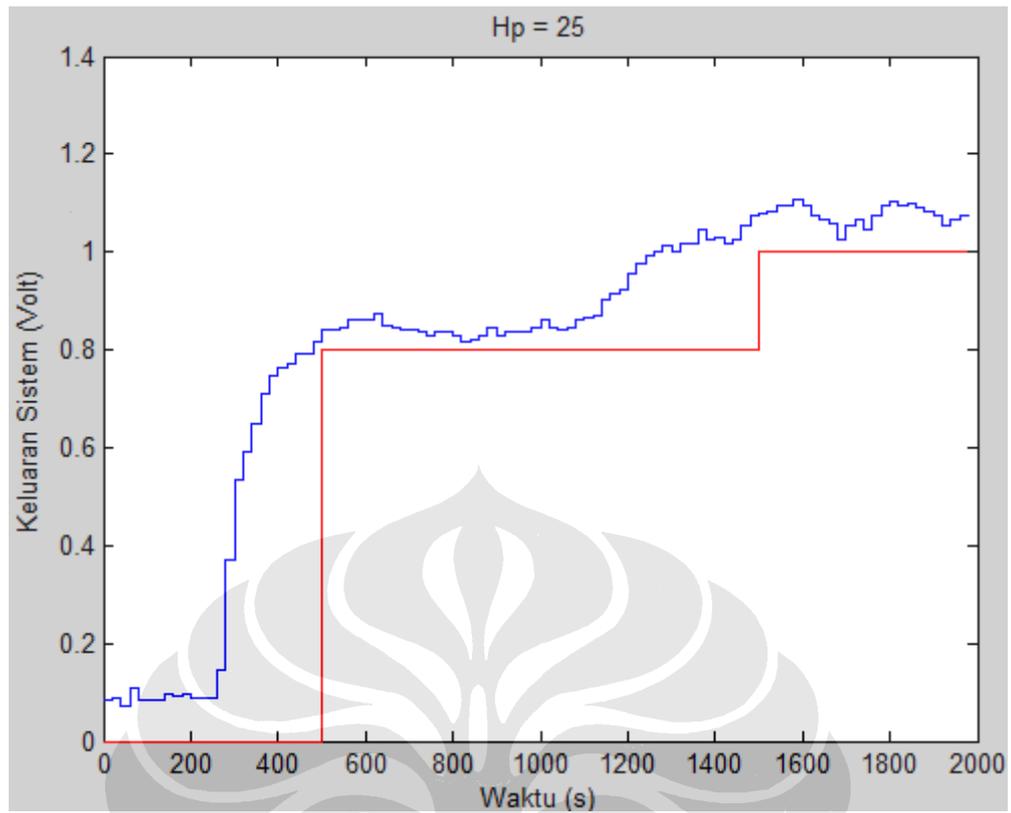


(b)

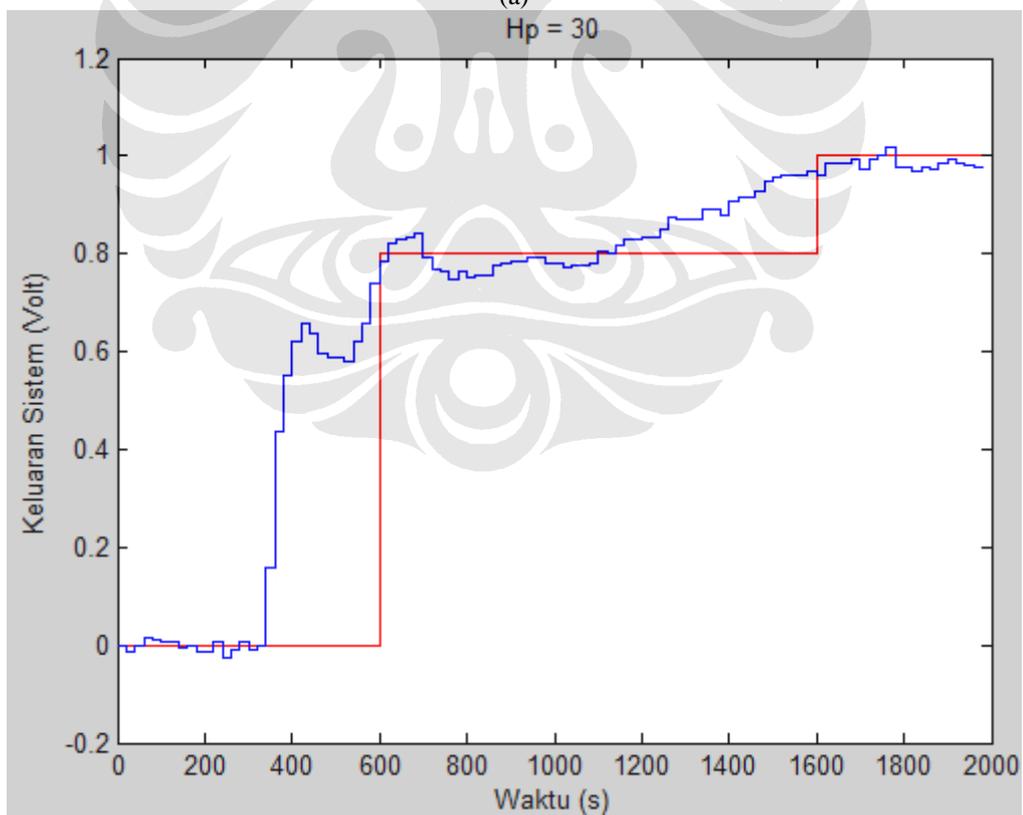


Gambar 4.9. Estimasi masukan hasil uji simulasi dengan nilai *prediction horizon* yang berbeda.

Pada uji eksperimen *Basic Process Rig 38-100* terbukti bahwa ketika nilai *prediction horizon* dibuat semakin jauh dengan nilai *control horizon* yaitu dari 25 menjadi 30, variansi perubahan sinyal kendali saat akan terjadi perubahan trayektori acuan tidak terlalu besar, sehingga keluaran sistem menjadi lebih halus seperti yang terlihat pada gambar 4.7 dan gambar 4.8.

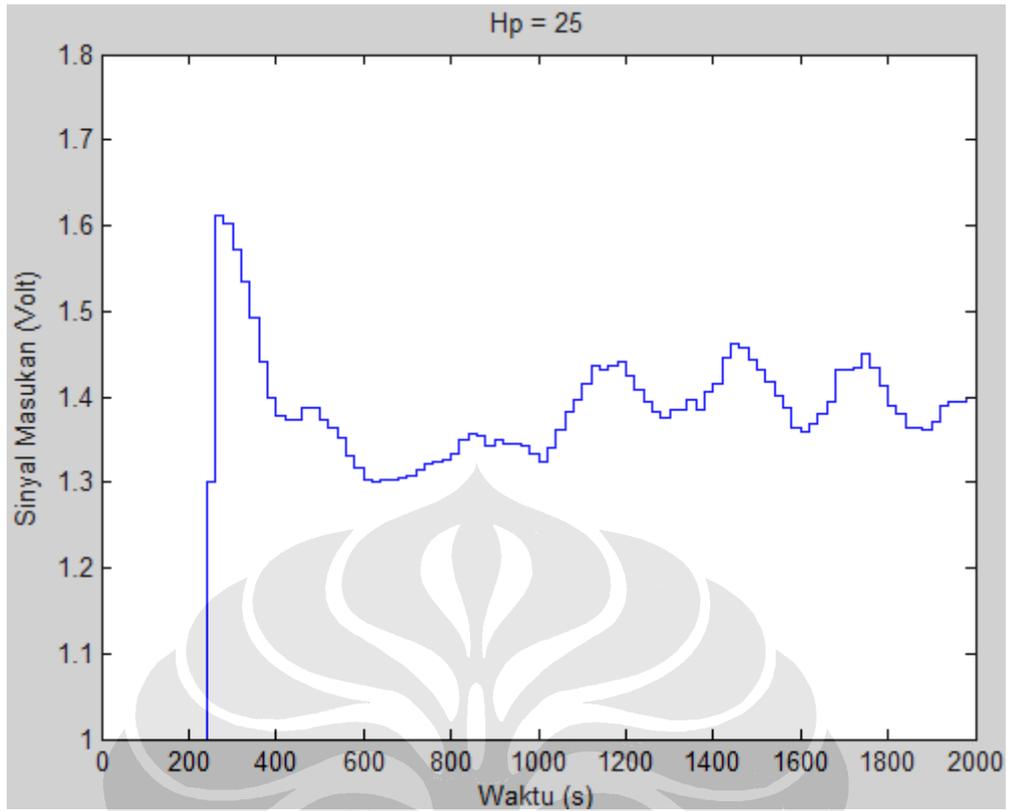


(a)

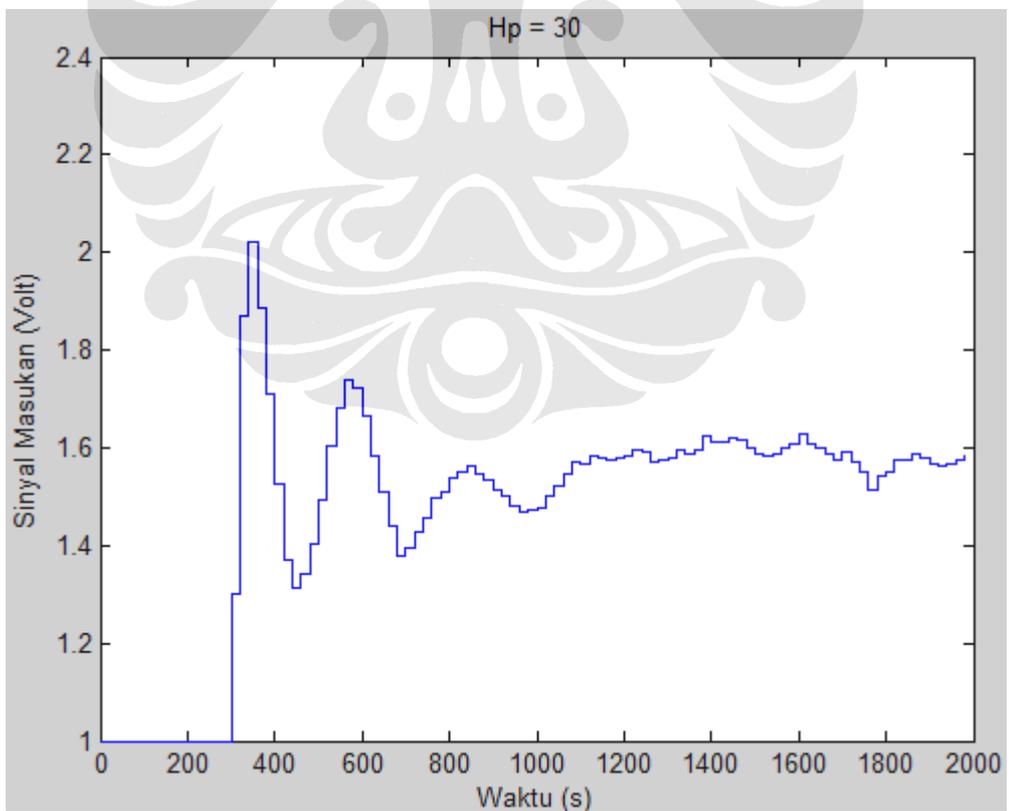


(b)

Gambar 4.10. Keluaran sistem hasil uji eksperimen dengan nilai *prediction horizon* yang berbeda.

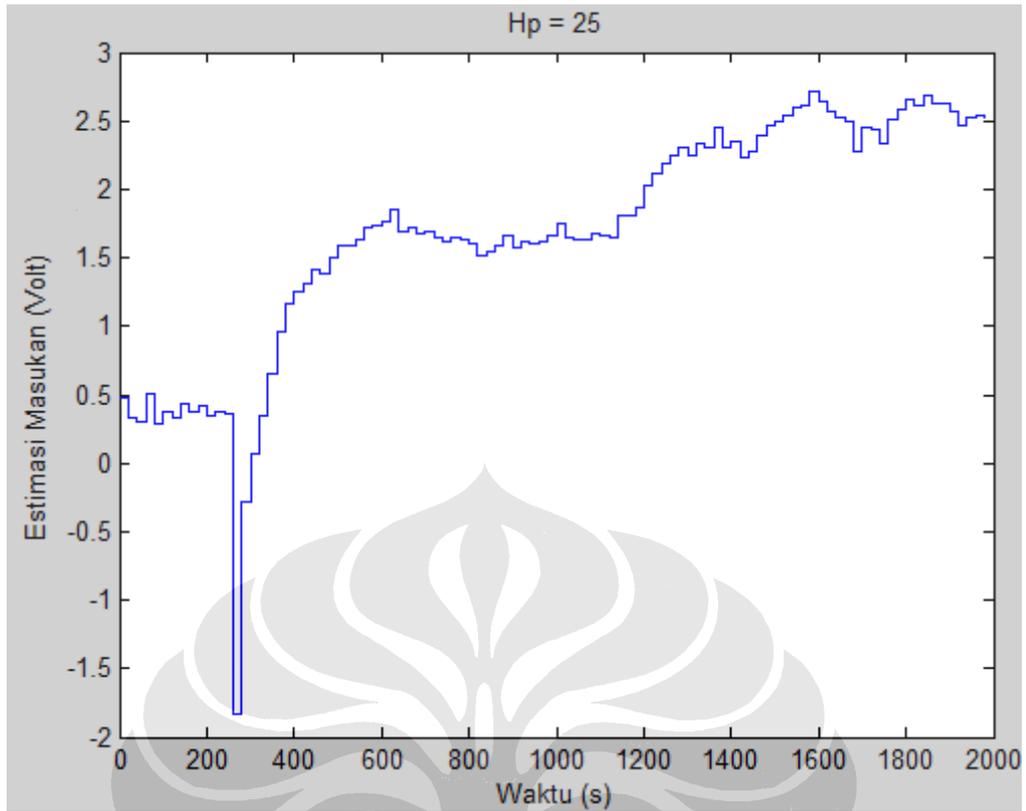


(a)

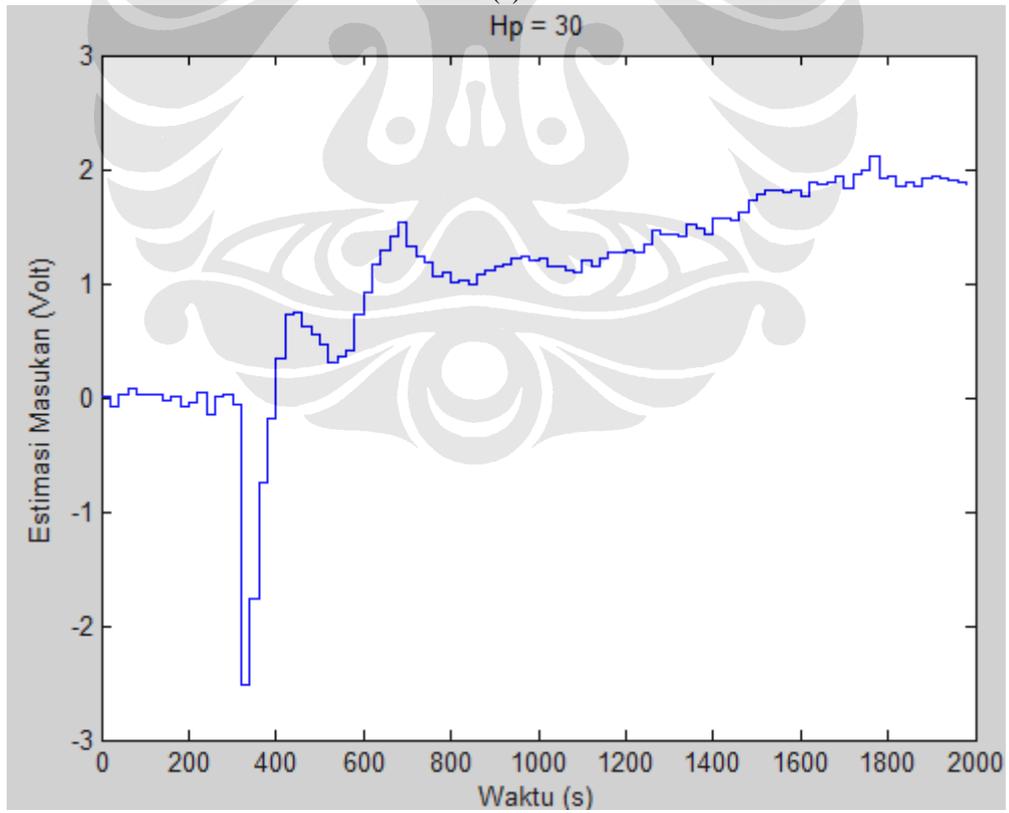


(b)

Gambar 4.11. Sinyal kendali hasil uji eksperimen dengan nilai *prediction horizon* yang berbeda.



(a)

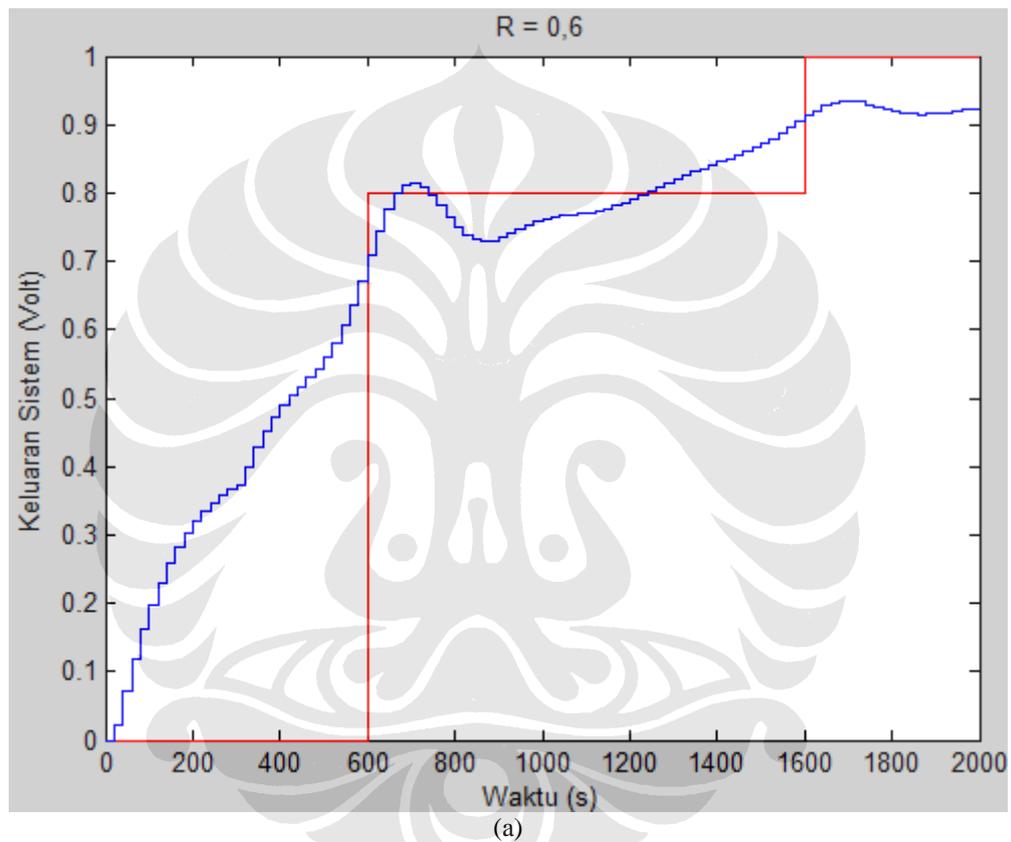


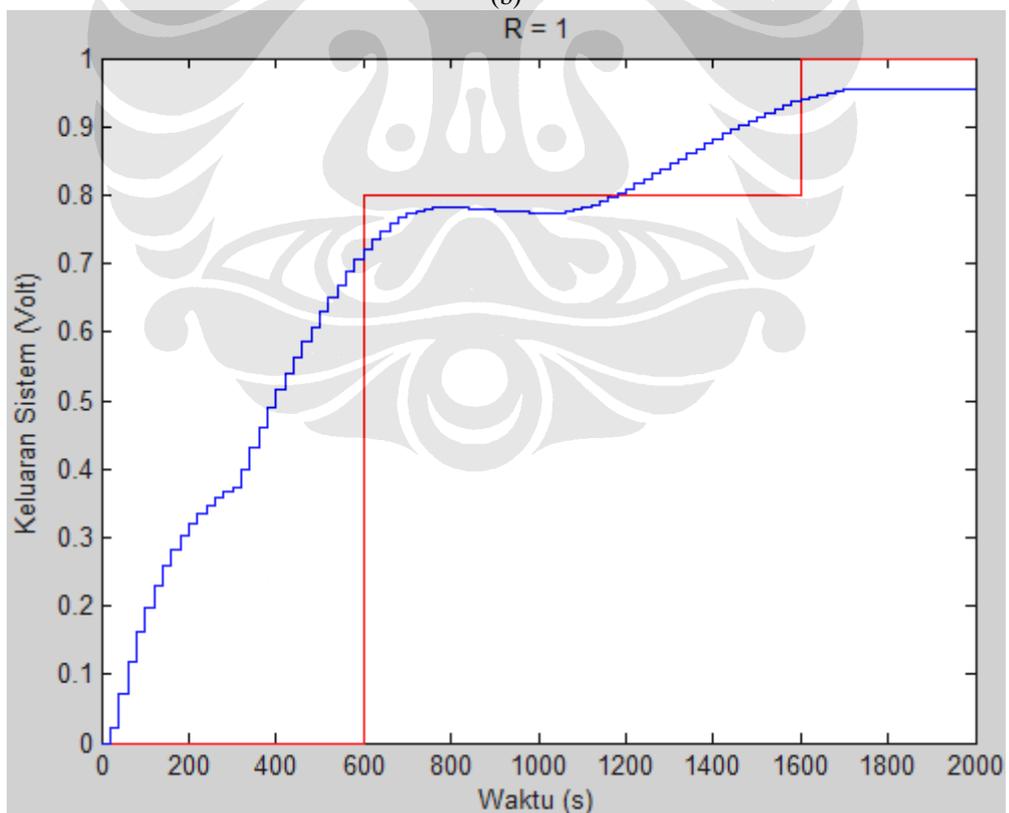
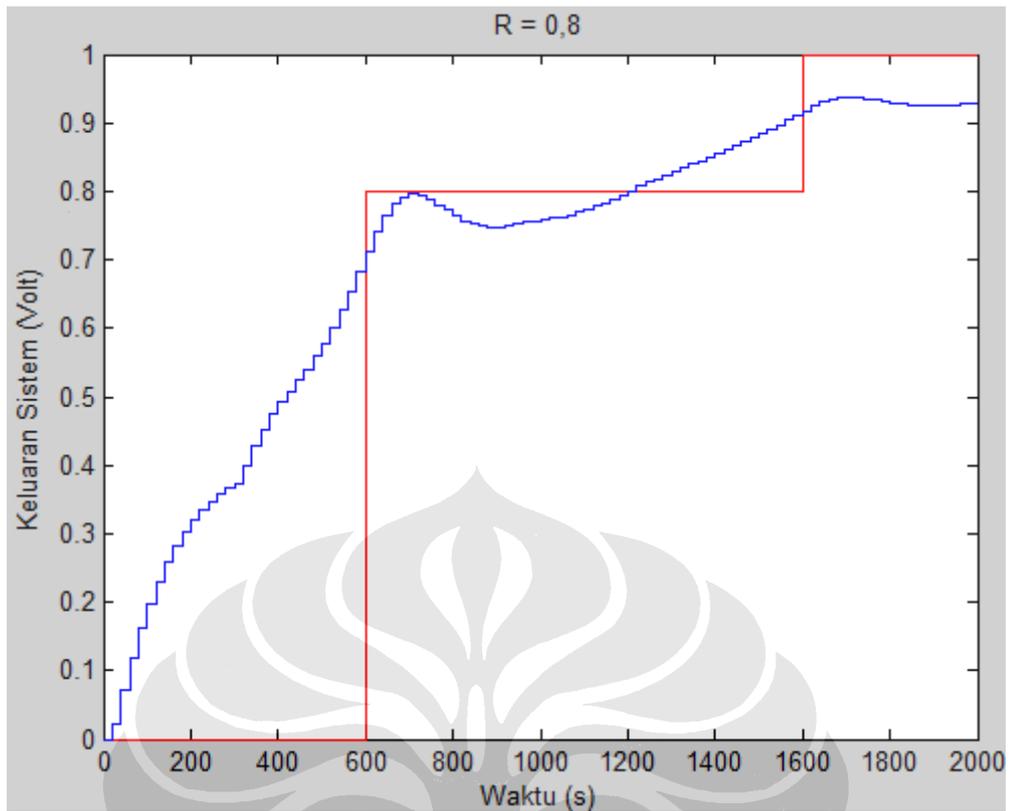
(b)

Gambar 4.12. Estimasi masukan hasil uji eksperimen dengan nilai *prediction horizon* yang berbeda.

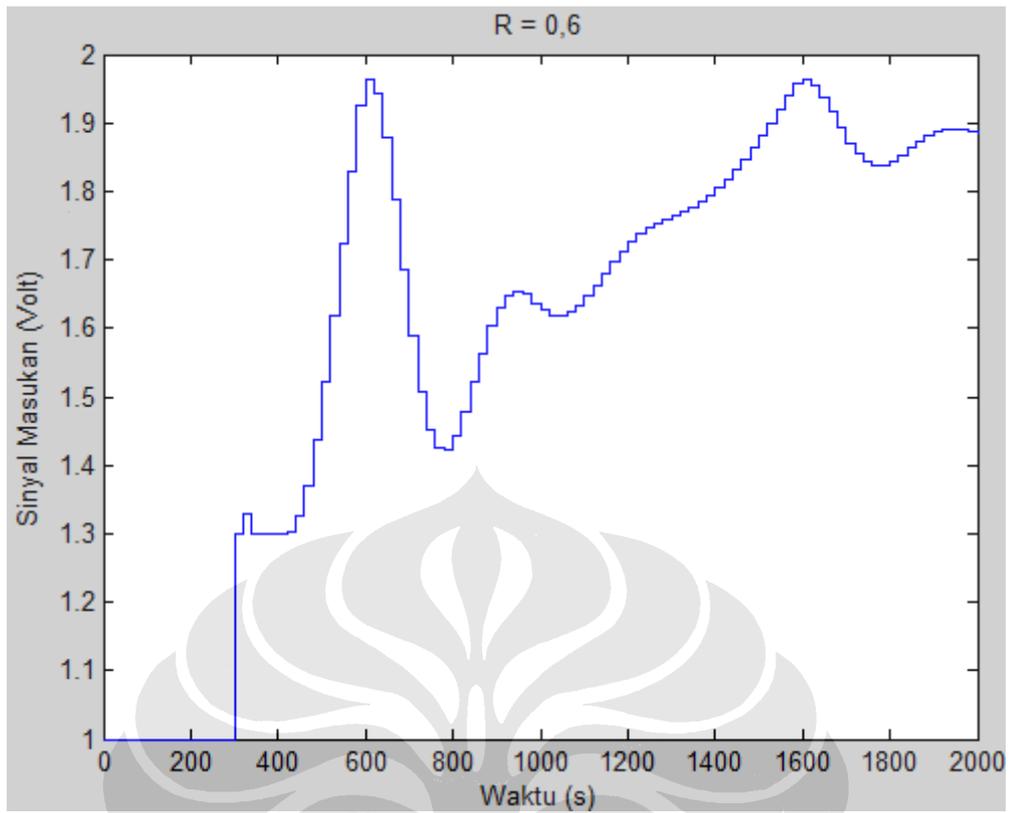
### 4.3 PENGARUH NILAI FAKTOR BOBOT PERUBAHAN SINYAL KENDALI ( $R$ ) PADA HASIL PENGENDALIAN MPC WITH CONSTRAINT

Untuk melihat pengaruh faktor bobot perubahan sinyal kendali  $R$  pada hasil pengendali, dilakukan uji eksperimen pada sistem dengan membuat nilai diagonal matriks  $R$  berbeda-beda, yaitu 0,6; 0,8; dan 1, sedangkan nilai parameter pengendali lainnya dibuat tetap, dengan  $H_u = 2$ ,  $H_p = 30$ , dan  $Q = I_{Hp}$

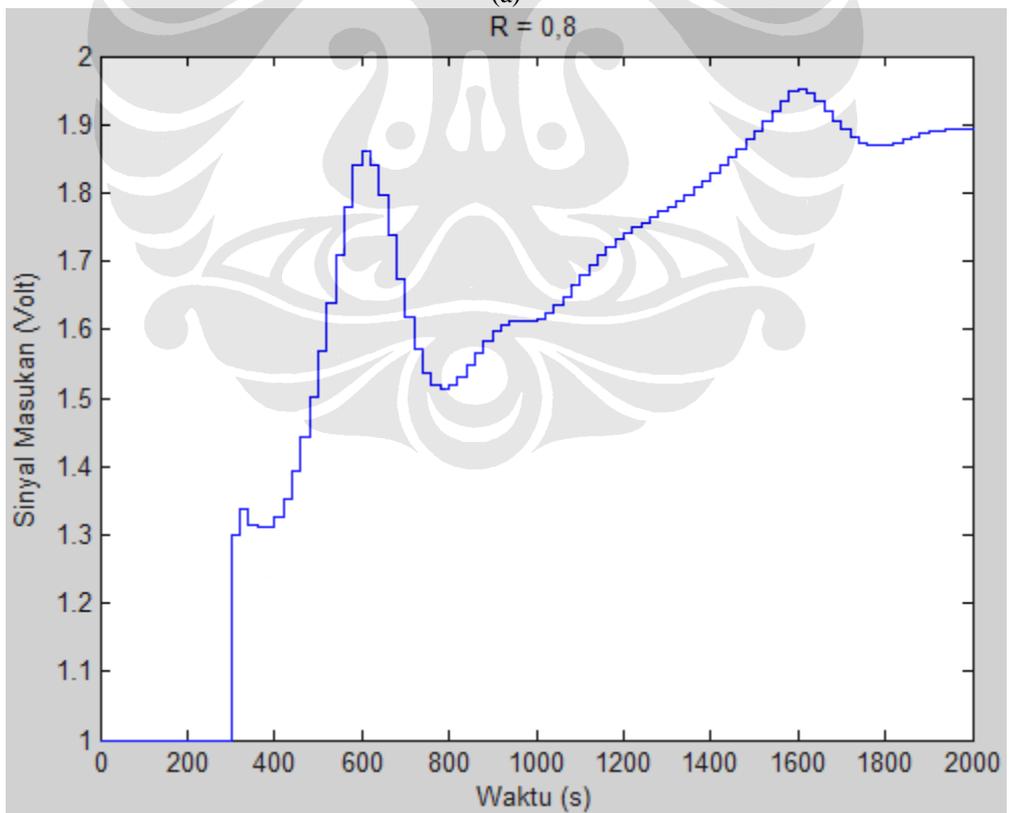




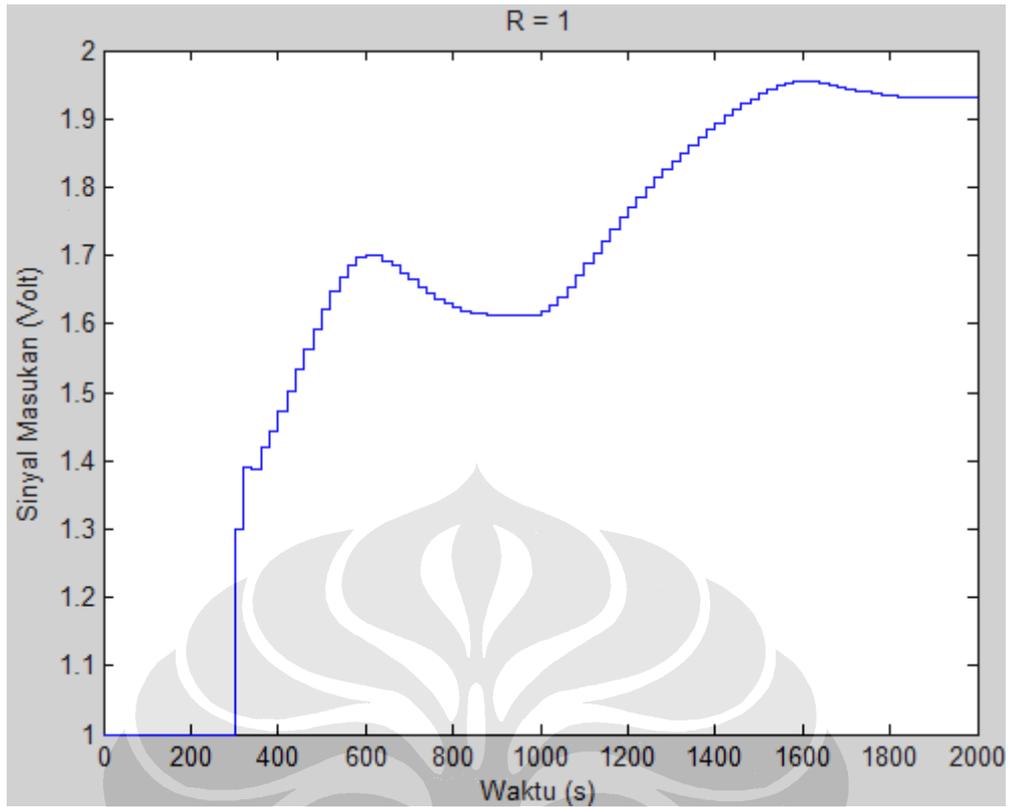
Gambar 4.13. Keluaran sistem hasil uji simulasi untuk nilai matriks  $R$  yang berbeda-beda.



(a)

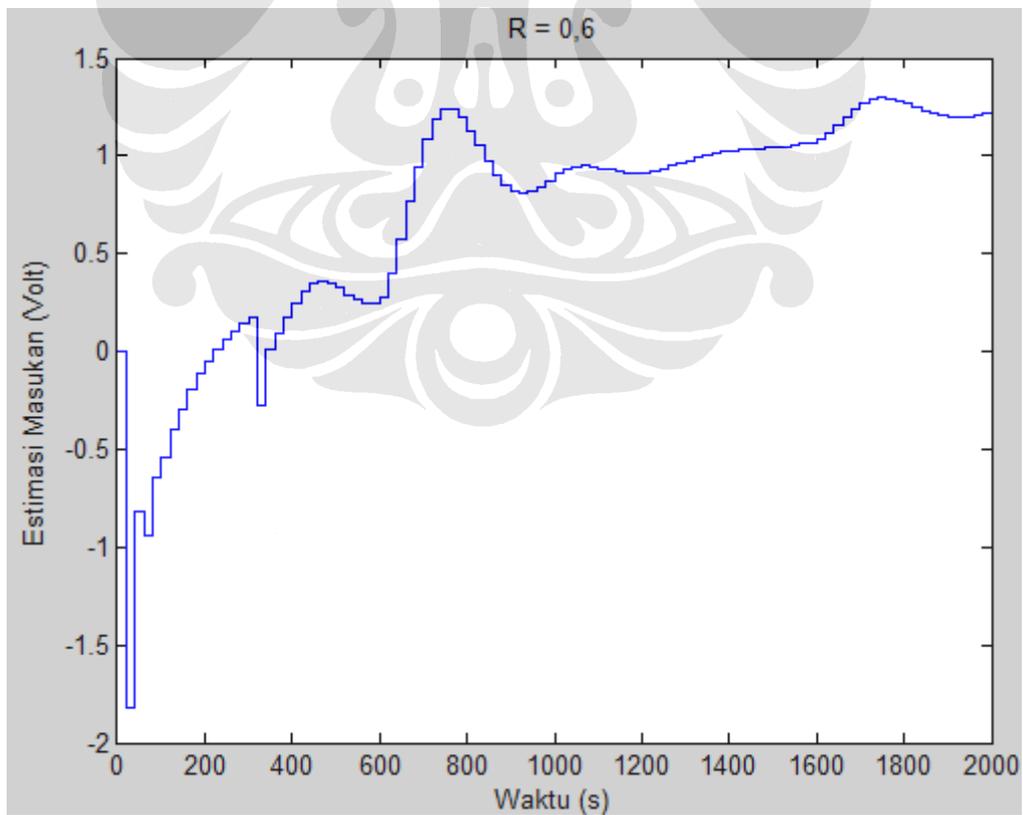


(b)

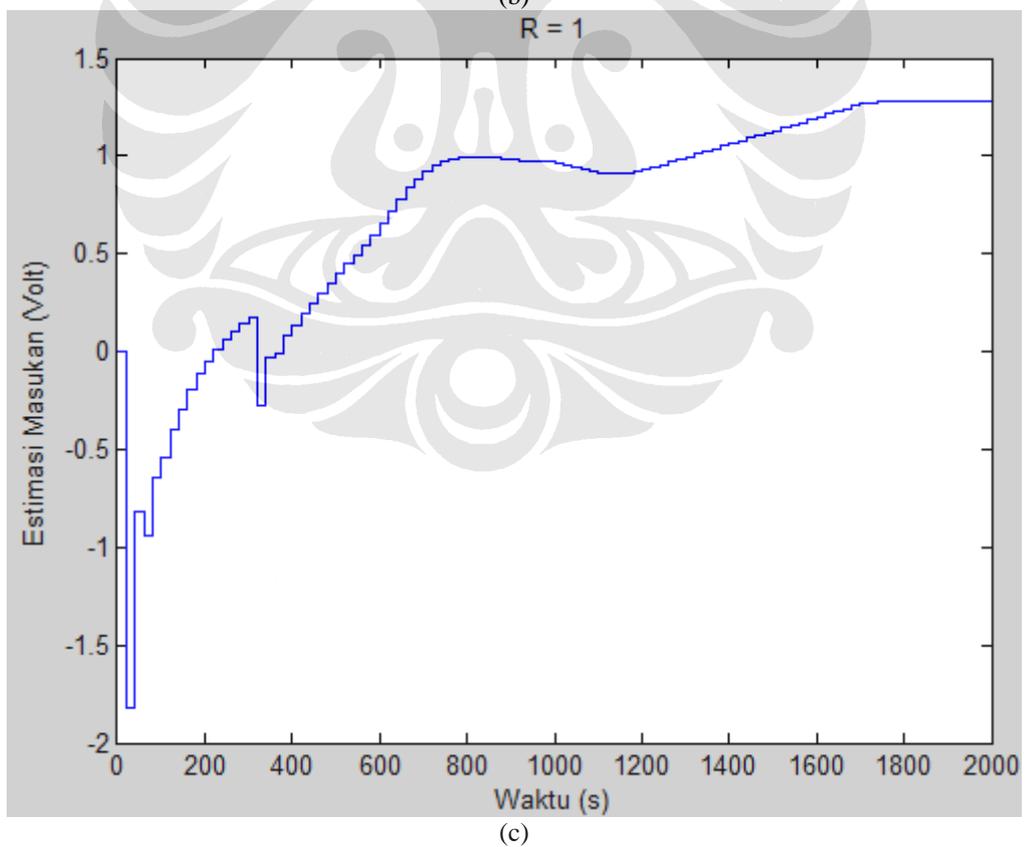
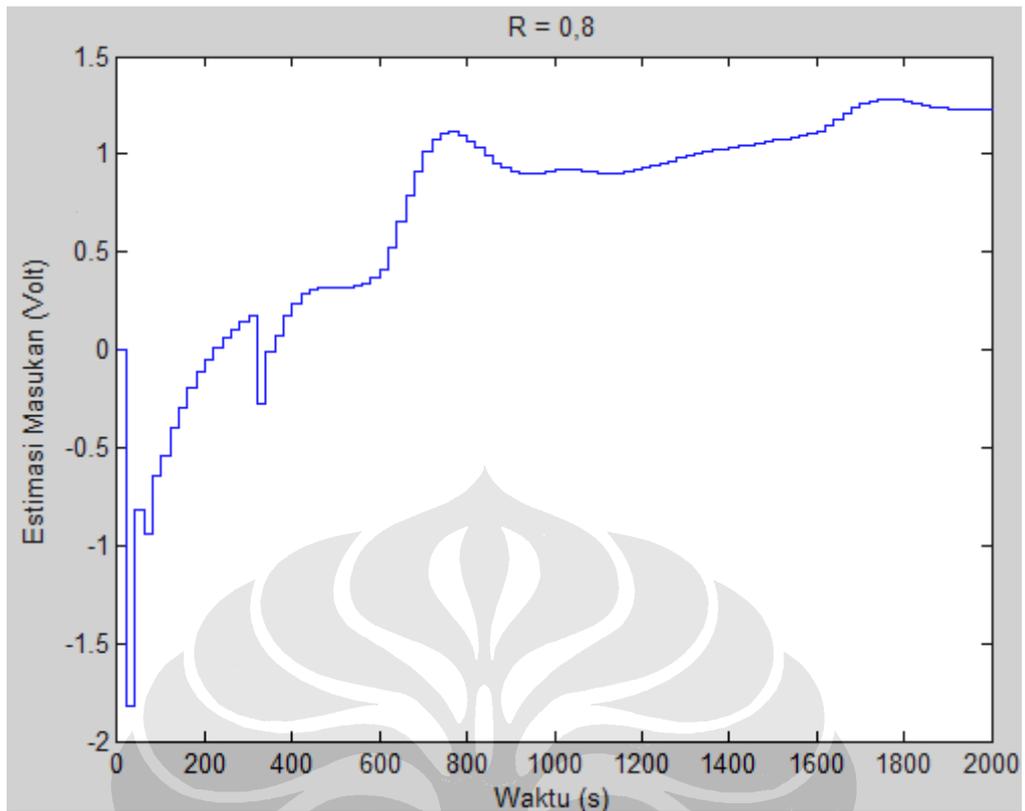


(c)

Gambar 4.14. Sinyal masukan hasil uji simulasi untuk nilai matriks  $R$  yang berbeda-beda.



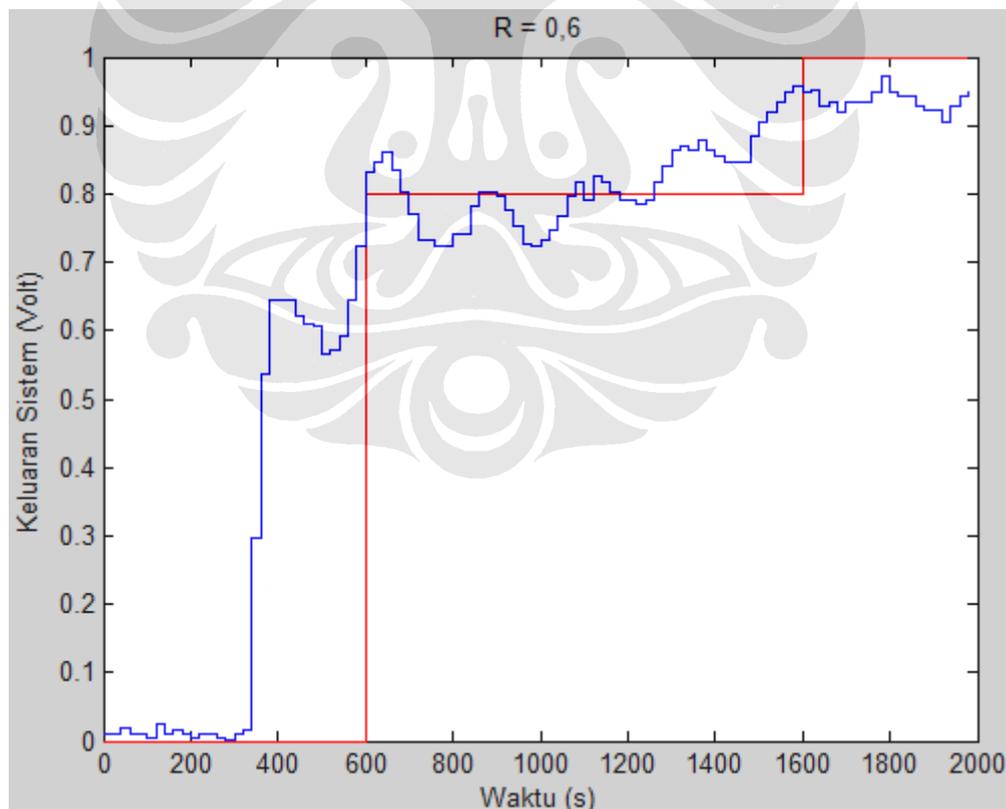
(a)



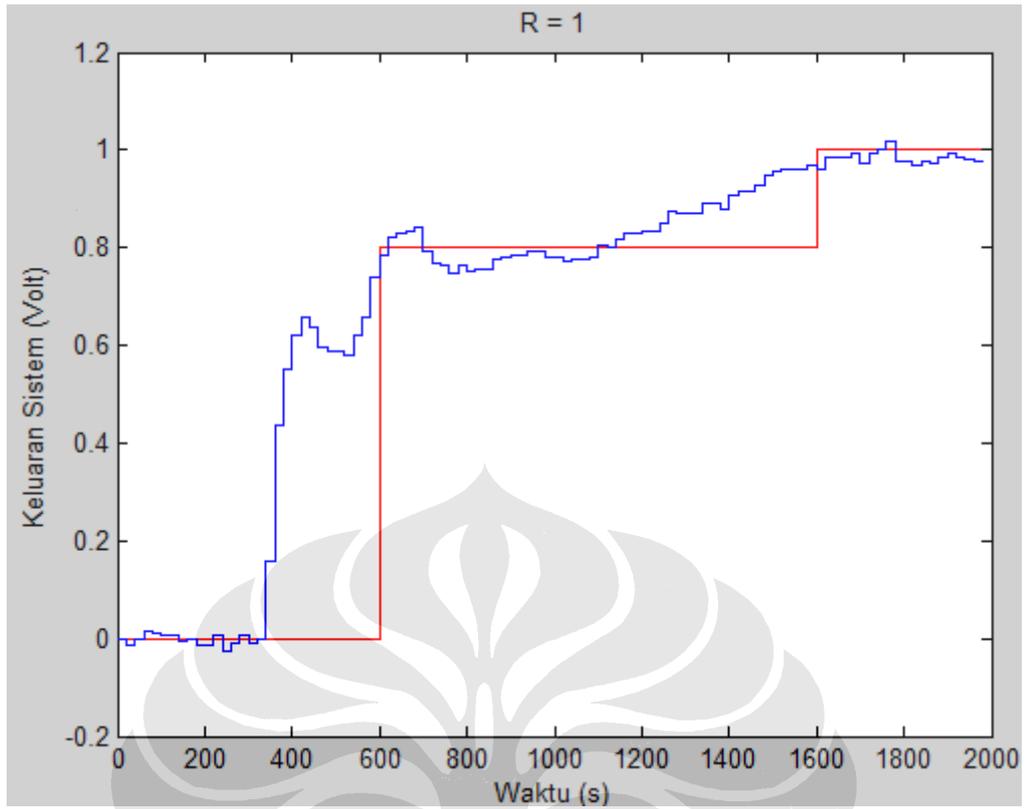
Gambar 4.15. Estimasi masukan hasil uji simulasi untuk nilai matriks  $R$  yang berbeda-beda.

Dari hasil simulasi pada gambar 4.13, gambar 4.14 dan gambar 4.15 terlihat bahwa semakin besar nilai matriks  $\underline{R}$ , perubahan sinyal kendali menjadi semakin ditekan sehingga menyebabkan keluaran sistem menjadi semakin halus. Dari simulasi terlihat bahwa dengan nilai matriks  $\underline{R}$  yang besar ( $I_{Hu}$ ), variansi perubahan sinyal kendali sangat kecil, bahkan saat akan terjadi perubahan trayektori acuan.

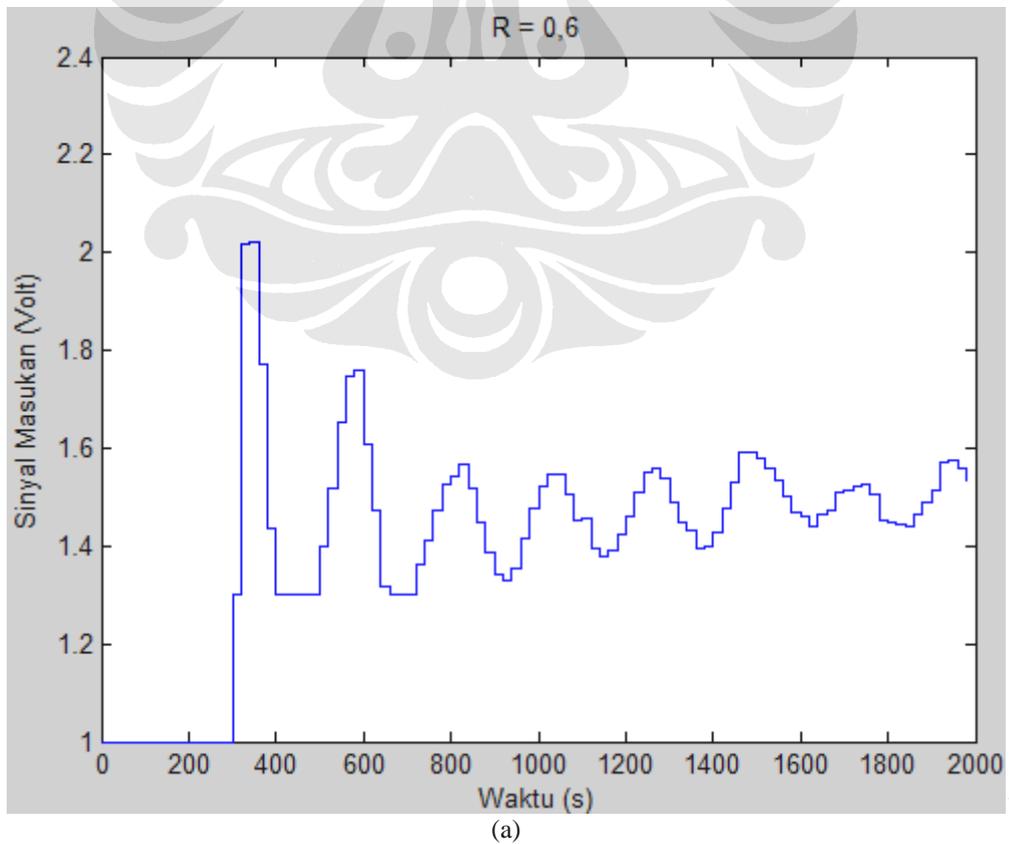
Untuk membuktikan hasil uji simulasi, dilakukan juga uji eksperimen pada *Basic Process Rig 38-100* dengan nilai matriks  $\underline{R}$  yang berbeda, yaitu sebesar  $0,6I_{Hu}$  dan  $I_{Hu}$ . Keluaran sistem dari uji eksperimen dengan nilai matriks  $\underline{R}$  yang berbeda terdapat pada gambar 4.16, sedangkan grafik sinyal kendali hasil uji eksperimen terdapat pada gambar 4.17 dan grafik estimasi masukan  $x_1$  hasil uji eksperimen terdapat pada gambar 4.18. Dari hasil uji eksperimen ternyata juga terlihat bahwa hasil pengendali akan semakin bagus jika nilai matriks  $\underline{R}$  semakin besar (mendekati  $I_{Hu}$ ).

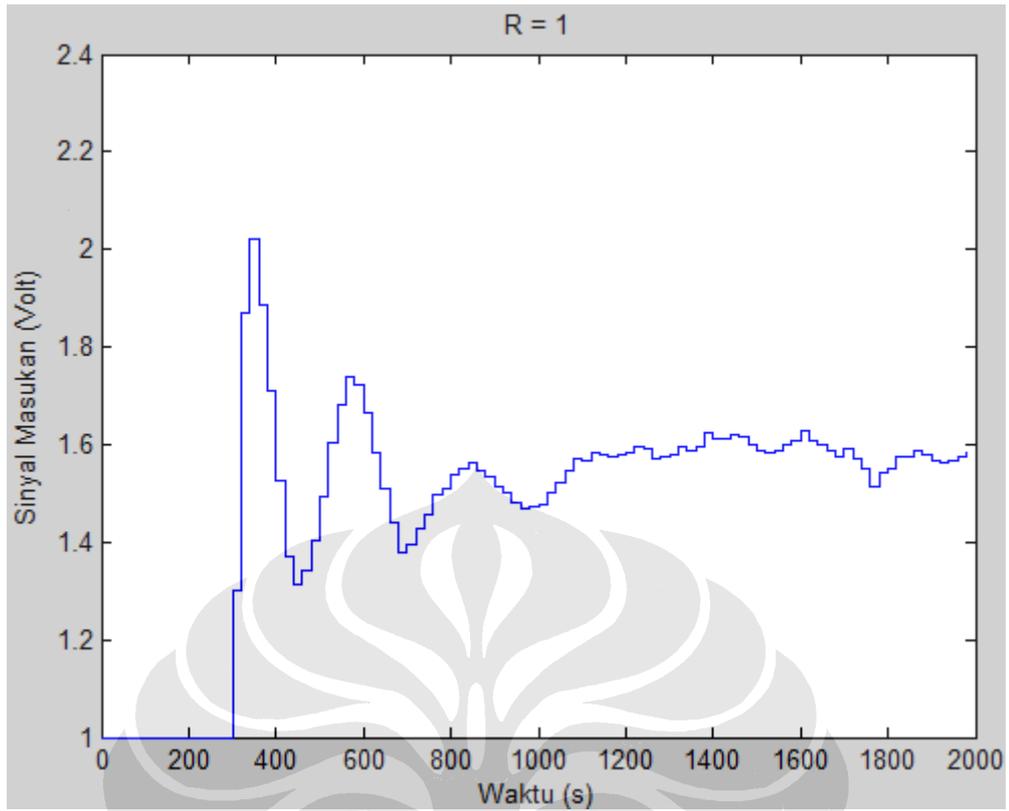


(a)



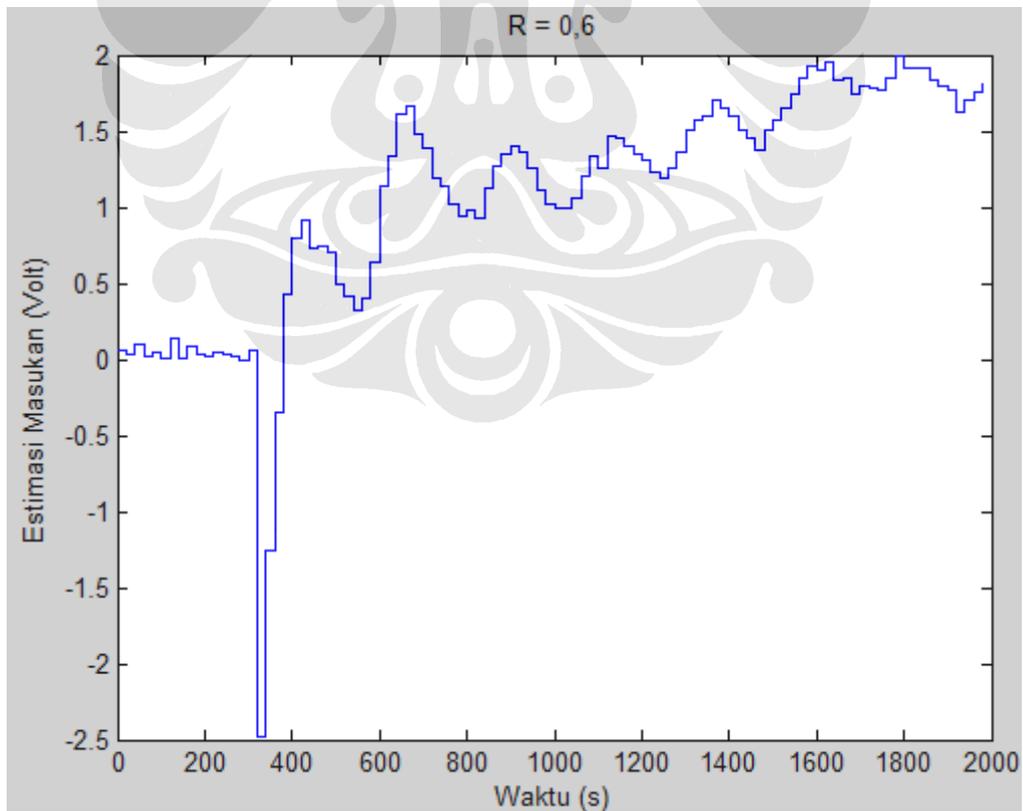
Gambar 4.16. Keluaran sistem hasil uji eksperimen untuk nilai matriks  $R$  yang berbeda-beda.



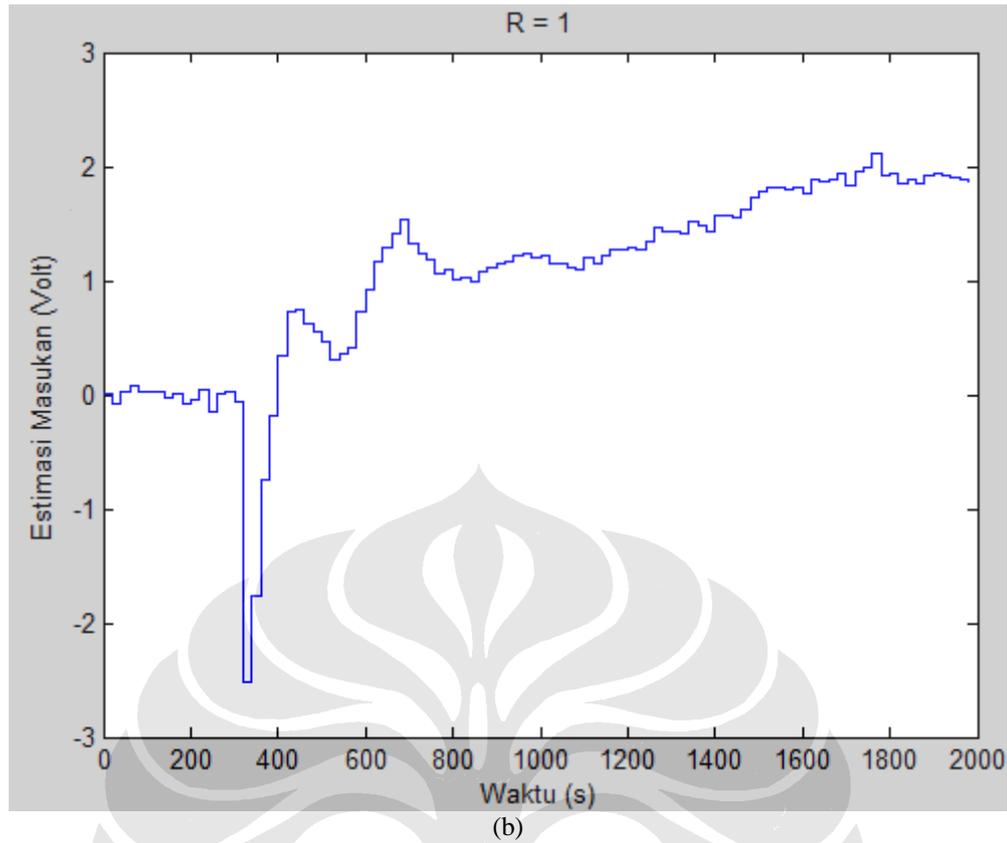


(b)

Gambar 4.17. Sinyal kendali hasil uji eksperimen untuk nilai matriks  $R$  yang berbeda-beda.



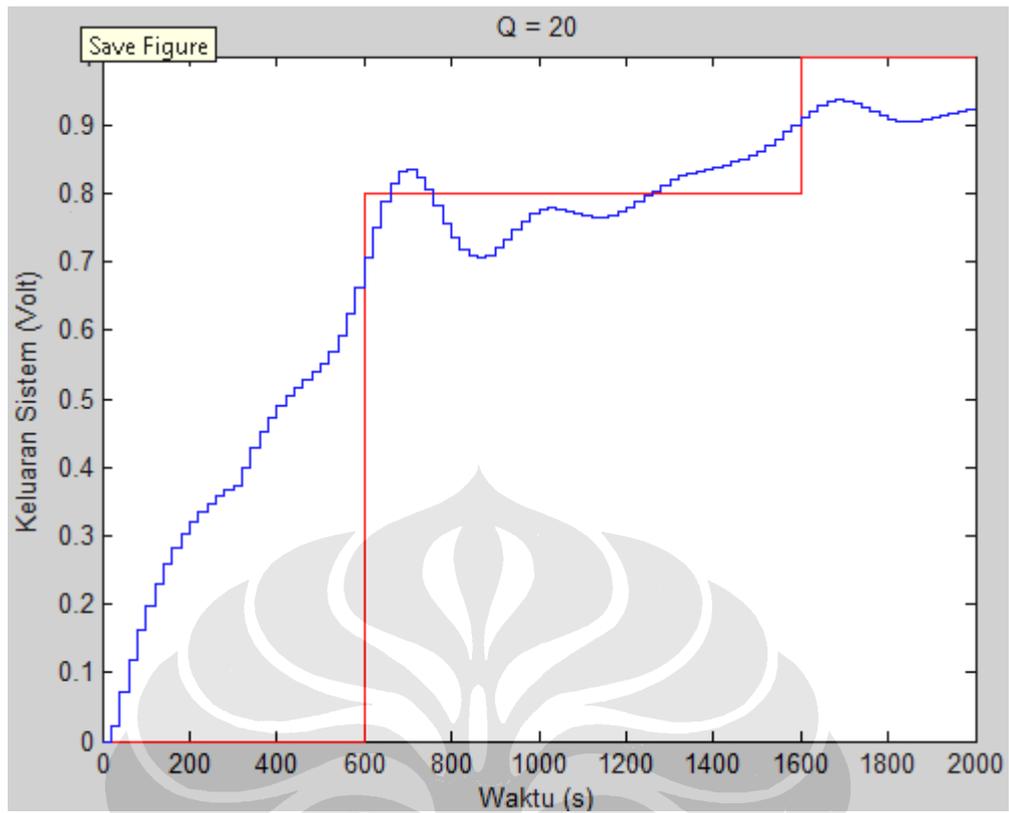
(a)



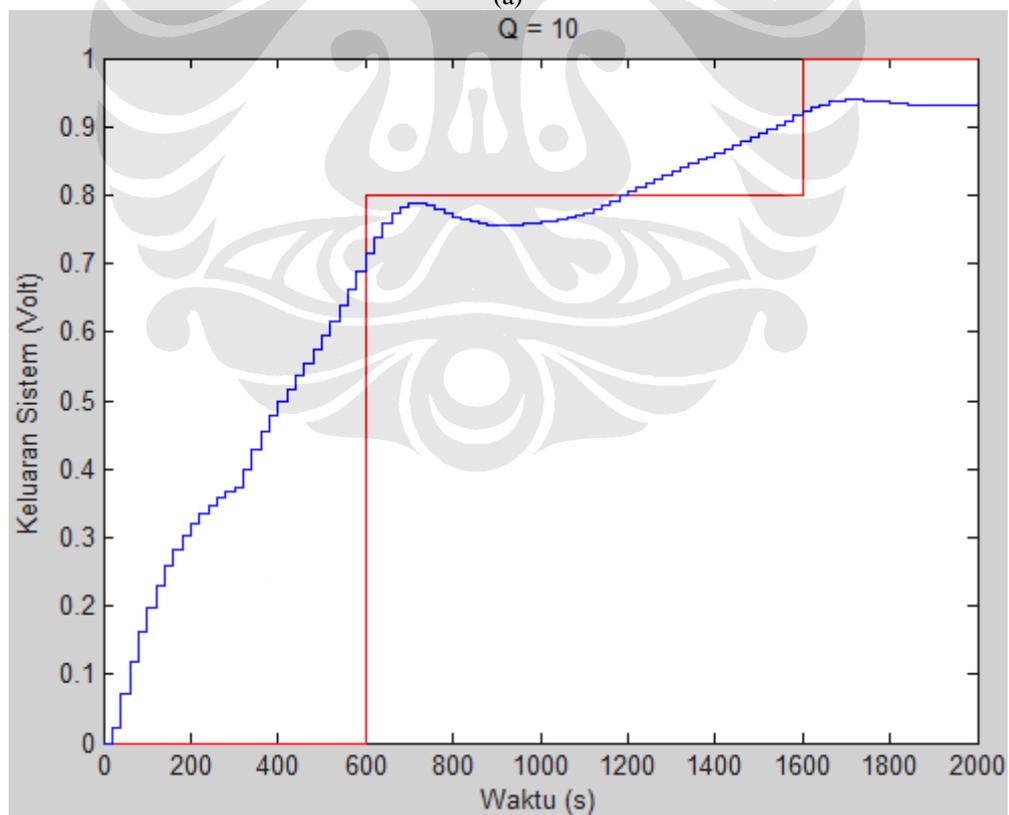
(b)  
Gambar 4.18. Estimasi masukan hasil uji eksperimen untuk nilai matriks  $\underline{R}$  yang berbeda-beda.

#### 4.4 PENGARUH NILAI FAKTOR BOBOT KESALAHAN ( $\underline{Q}$ ) PADA HASIL PENGENDALIAN MPC *WITH CONSTRAINT*

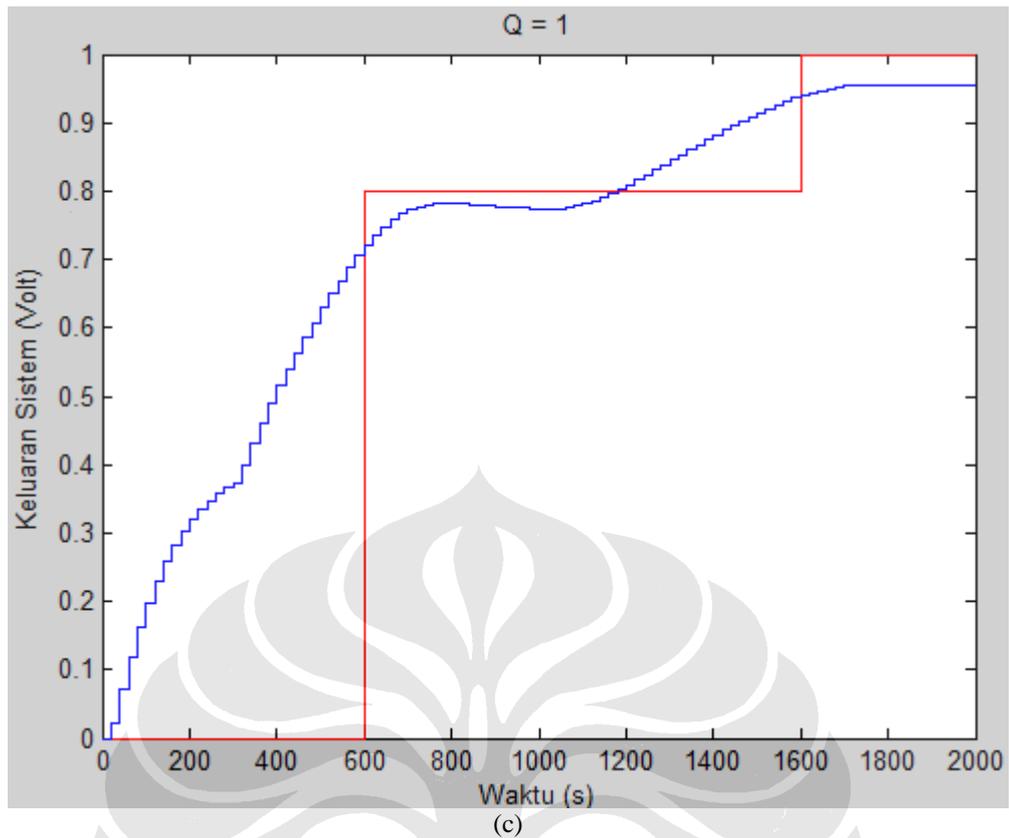
Untuk melihat pengaruh faktor bobot kesalahan  $\underline{Q}$  pada hasil pengendali, dilakukan uji eksperimen pada sistem dengan membuat nilai diagonal matriks  $\underline{Q}$  berbeda-beda, yaitu 1; 10; dan 20, sedangkan nilai parameter pengendali lainnya dibuat tetap, dengan  $Hu = 2$ ,  $Hp = 30$ , dan  $\underline{R} = I_{Hu}$



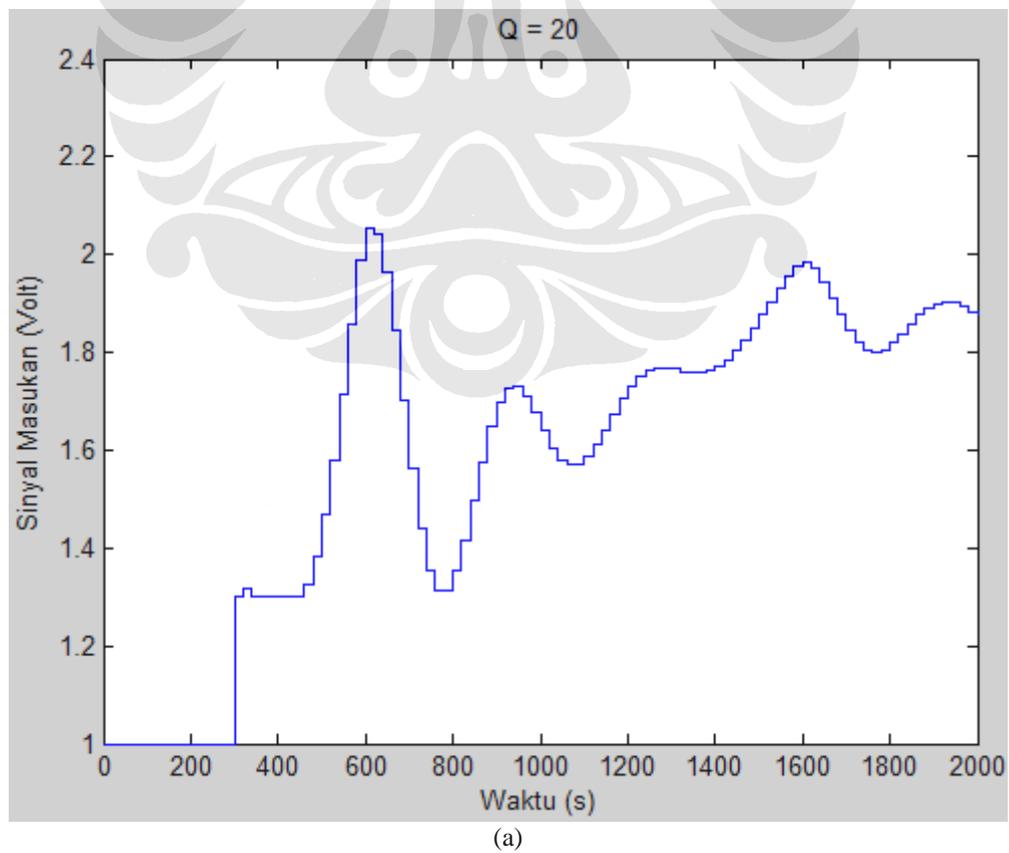
(a)

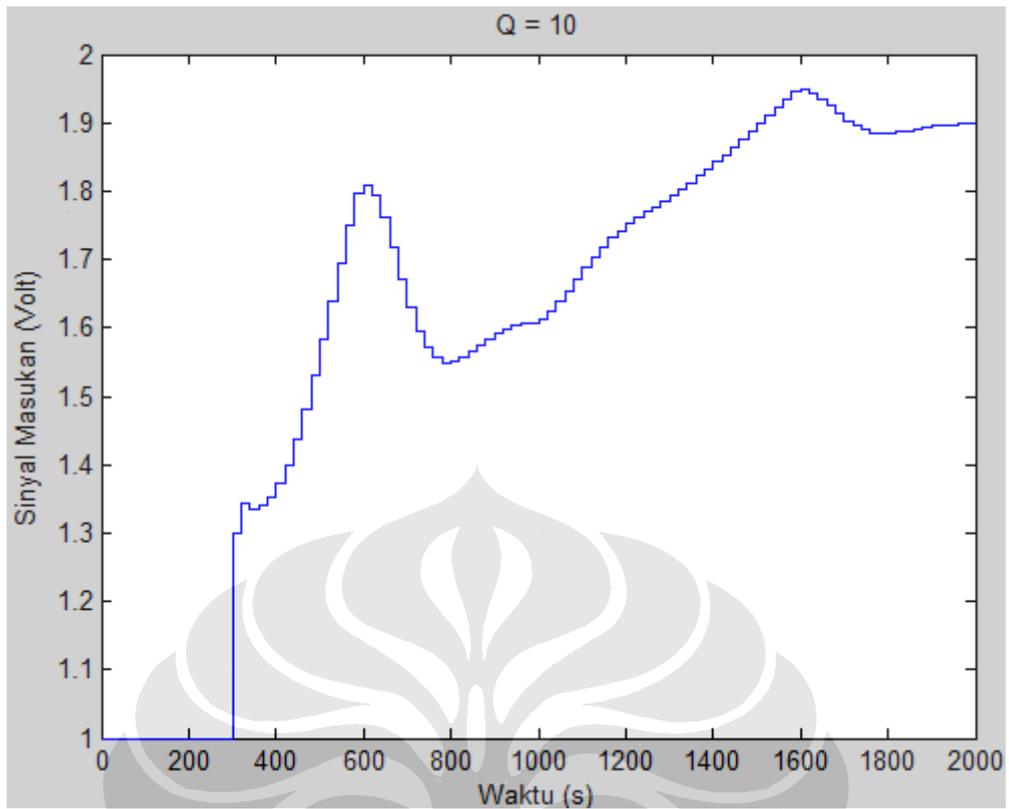


(b)

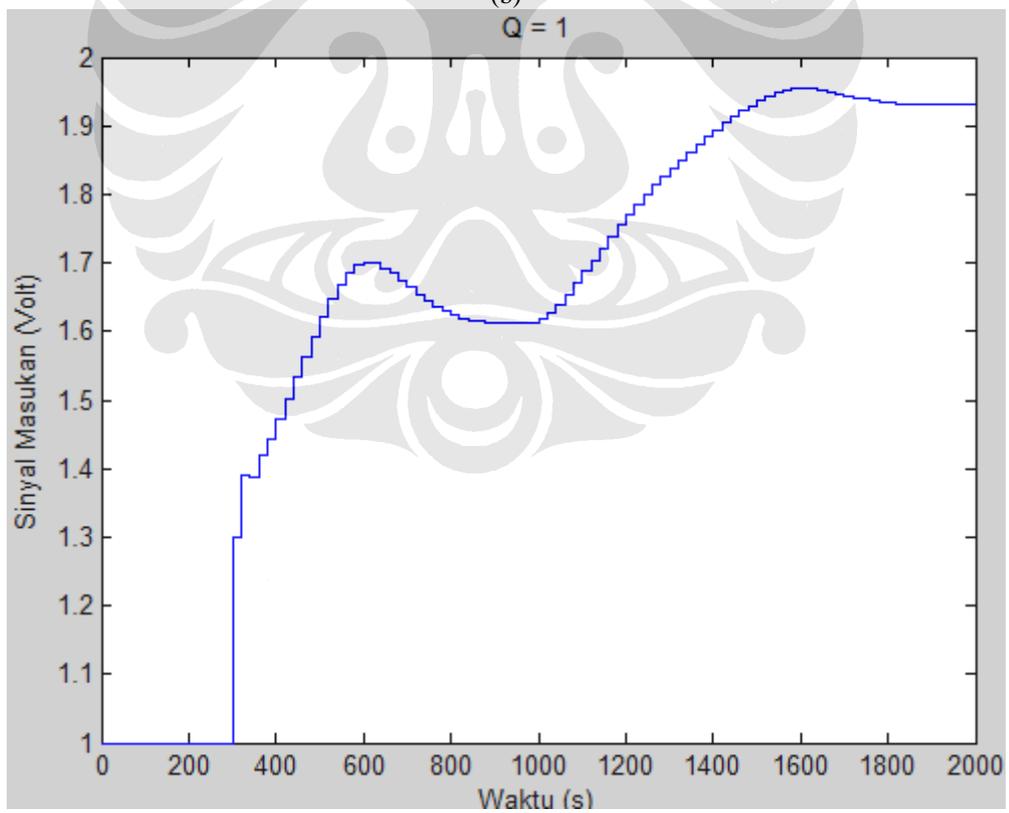


Gambar 4.19. Keluaran sistem hasil uji simulasi untuk nilai matriks  $Q$  yang berbeda-beda.



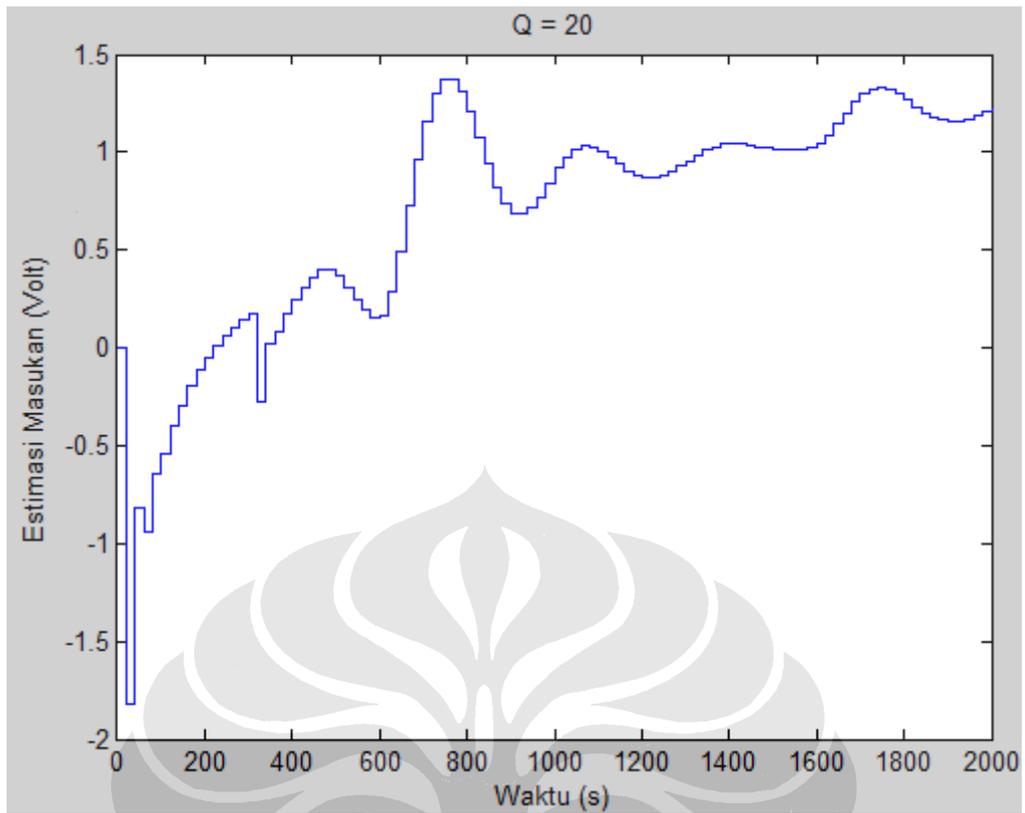


(b)

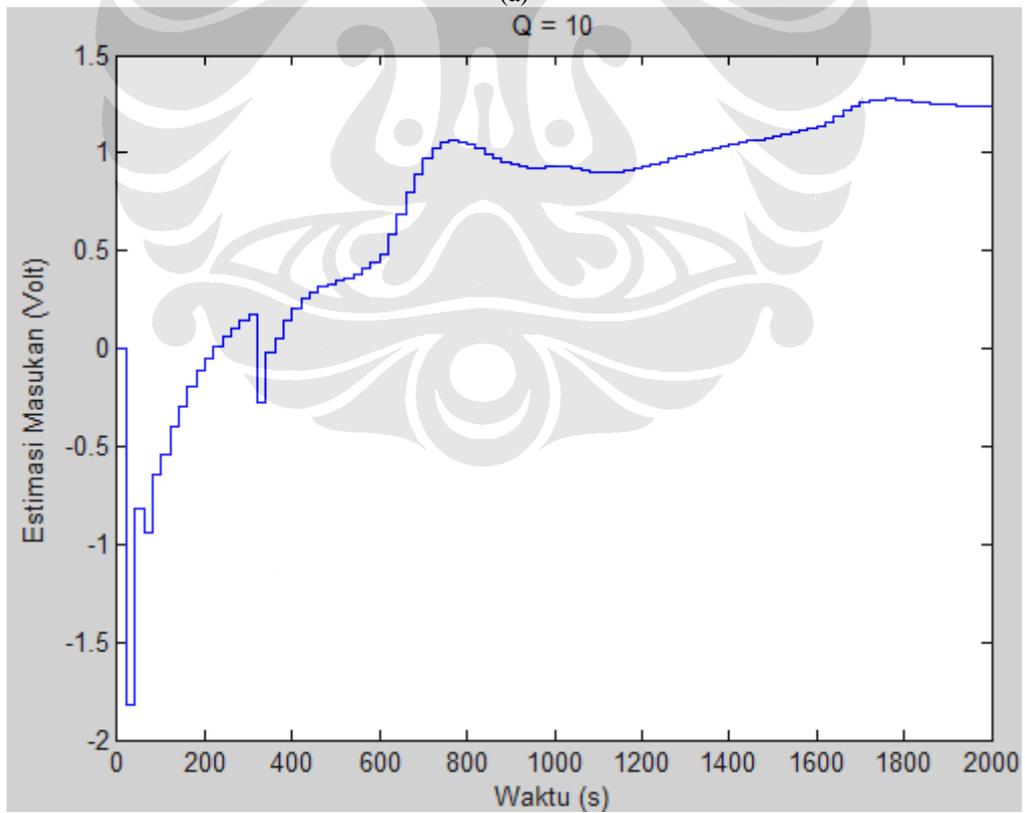


(c)

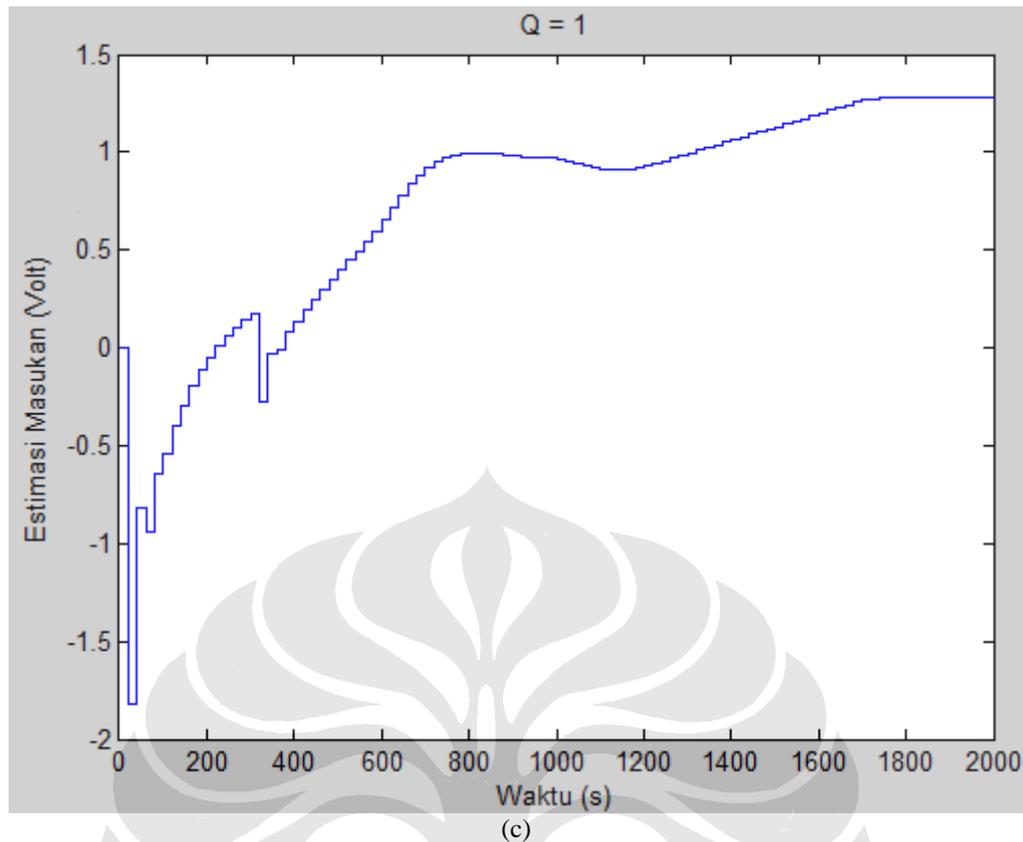
Gambar 4.20. Sinyal masukan hasil uji simulasi untuk nilai matriks  $Q$  yang berbeda-beda.



(a)



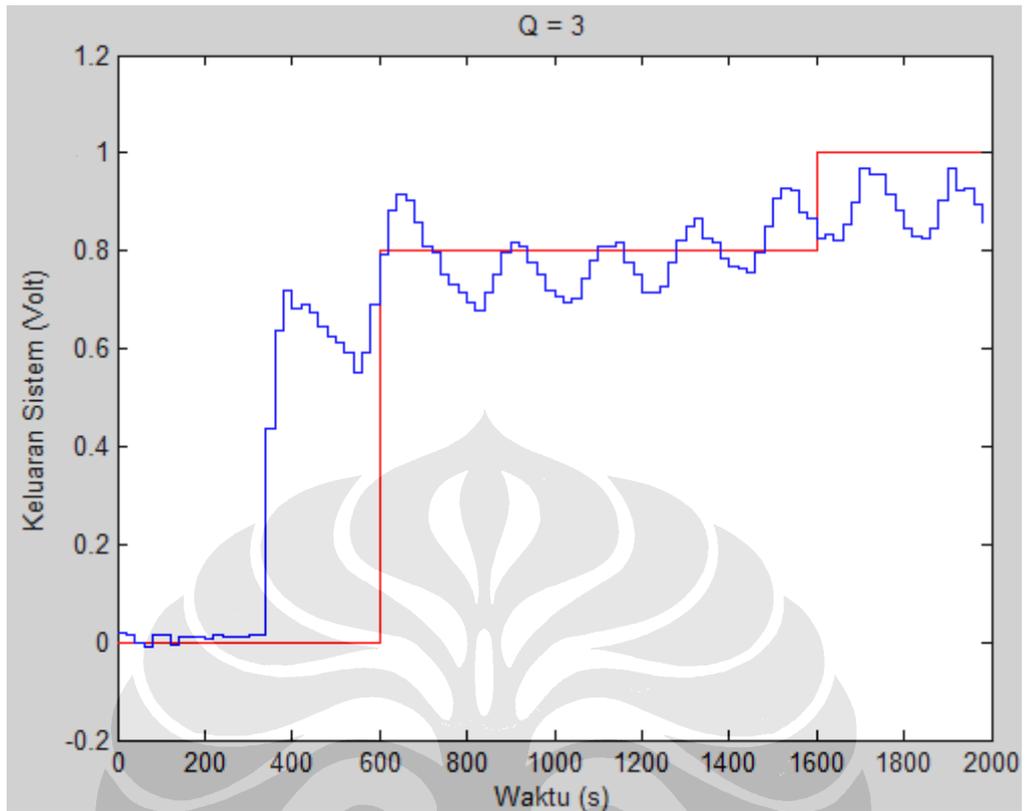
(b)



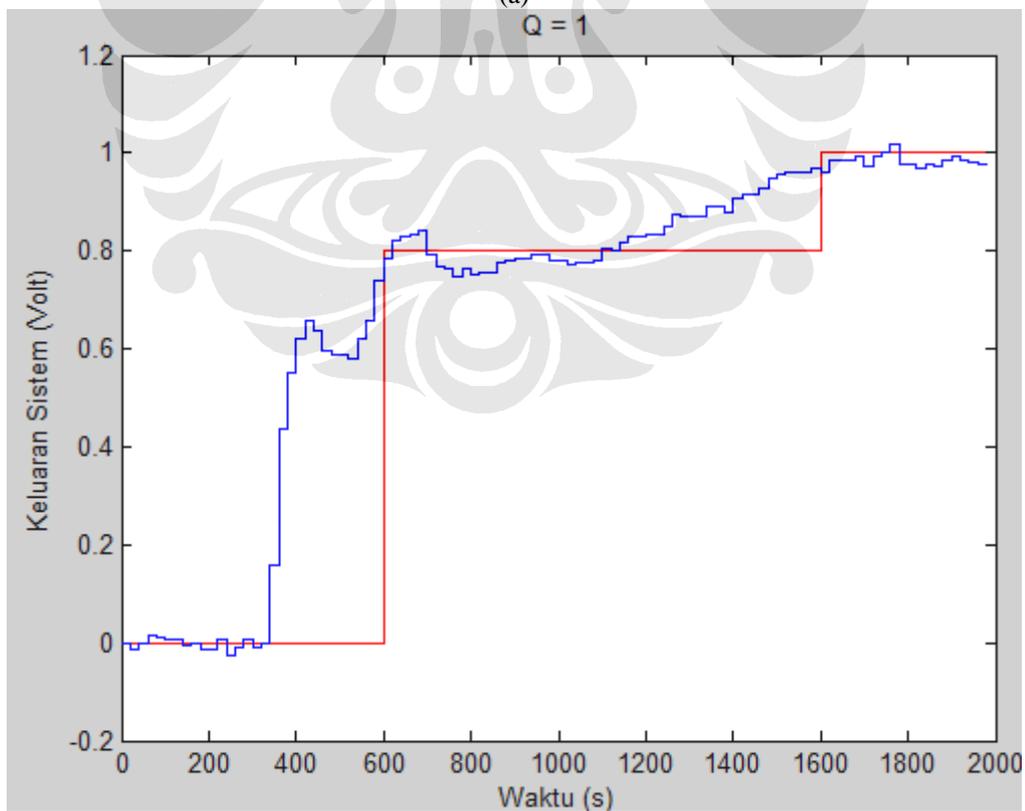
Gambar 4.21. Estimasi masukan hasil uji simulasi untuk nilai matriks  $Q$  yang berbeda-beda.

Dari hasil simulasi pada gambar 4.19, gambar 20 dan gambar 4.21 terlihat bahwa semakin kecil nilai matriks  $Q$ , perubahan sinyal kendali menjadi semakin ditekan sehingga menyebabkan keluaran sistem menjadi semakin halus. Dari simulasi terlihat bahwa dengan nilai matriks  $Q$  yang kecil ( $I_{Hp}$ ), variansi perubahan sinyal kendali sangat kecil, bahkan saat akan terjadi perubahan trayektori acuan. Dengan melihat hasil simulasi di atas dapat disimpulkan bahwa pengaruh matriks  $Q$  merupakan keterbalikan dari pengaruh matriks  $R$ .

Untuk membuktikan hasil uji simulasi, dilakukan juga uji eksperimen pada *Basic Process Rig 38-100* dengan nilai matriks  $Q$  yang berbeda, yaitu sebesar  $3I_{Hp}$  dan  $I_{Hp}$ . Keluaran sistem dari uji eksperimen dengan nilai matriks  $Q$  yang berbeda terdapat pada gambar 4.22, sedangkan grafik sinyal kendali hasil uji eksperimen terdapat pada gambar 4.23 dan grafik estimasi masukan  $x_1$  hasil uji eksperimen terdapat pada gambar 4.24. Dari hasil uji eksperimen ternyata juga terlihat bahwa hasil pengendali akan semakin bagus jika nilai matriks  $Q$  semakin kecil (mendekati  $I_{Hp}$ ).

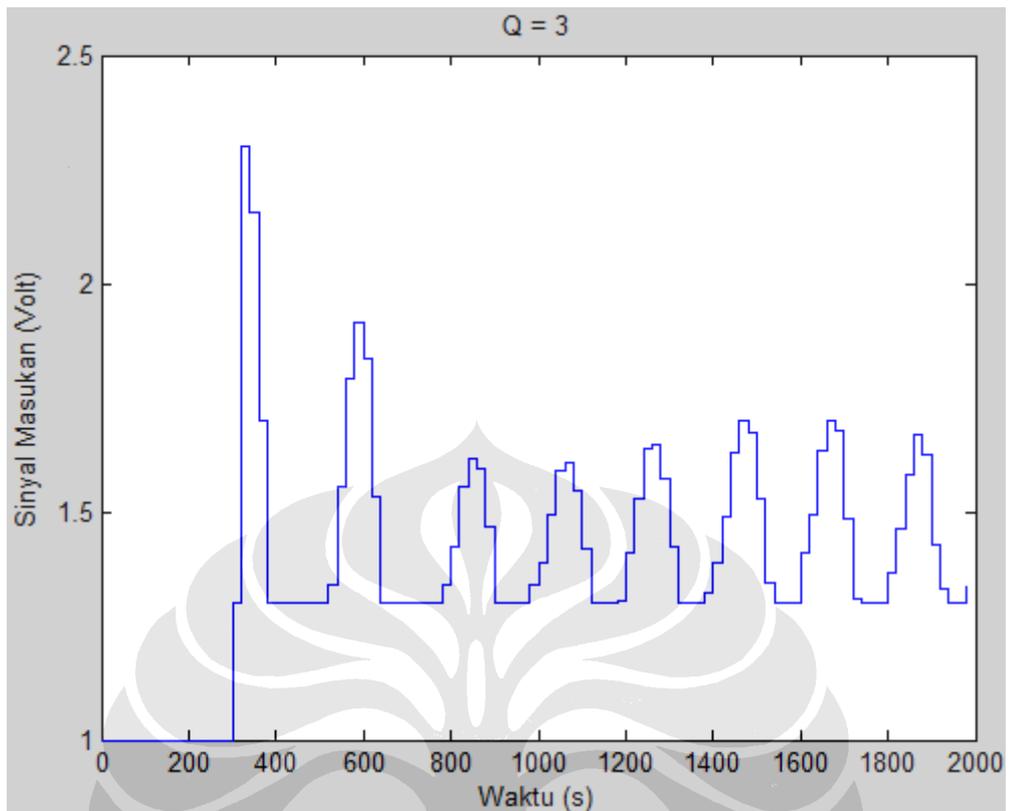


(a)

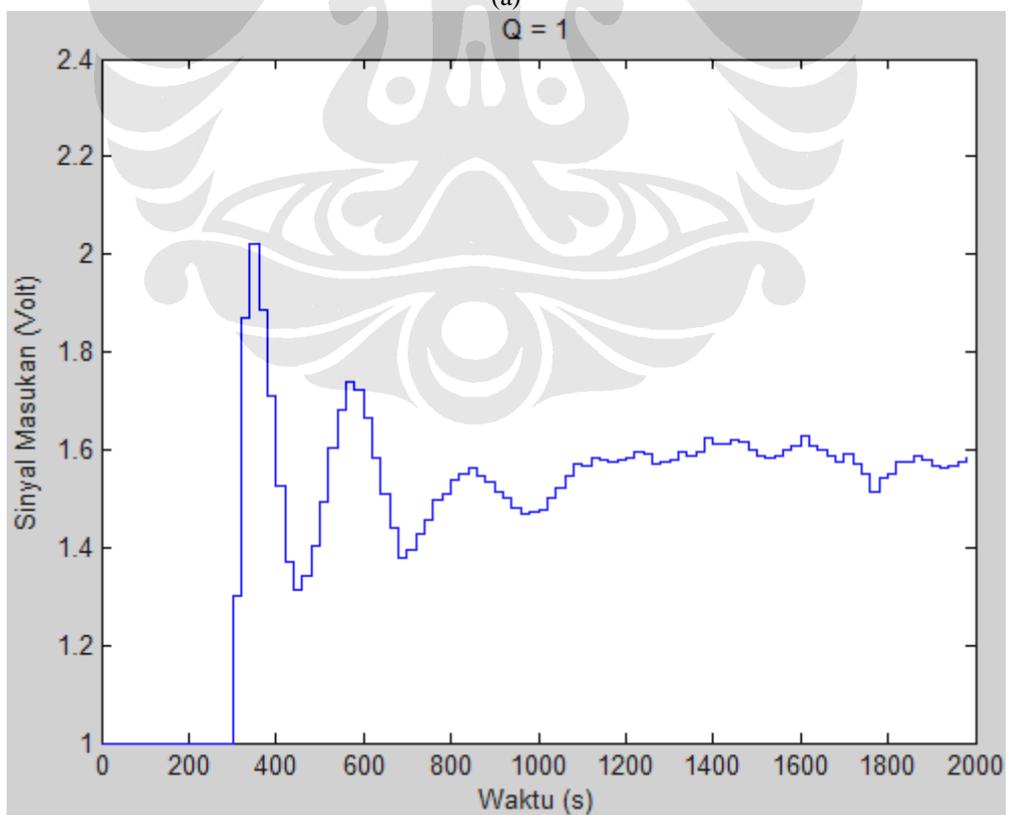


(b)

Gambar 4.22. Keluaran sistem hasil uji eksperimen untuk nilai matriks  $\underline{R}$  yang berbeda-beda.

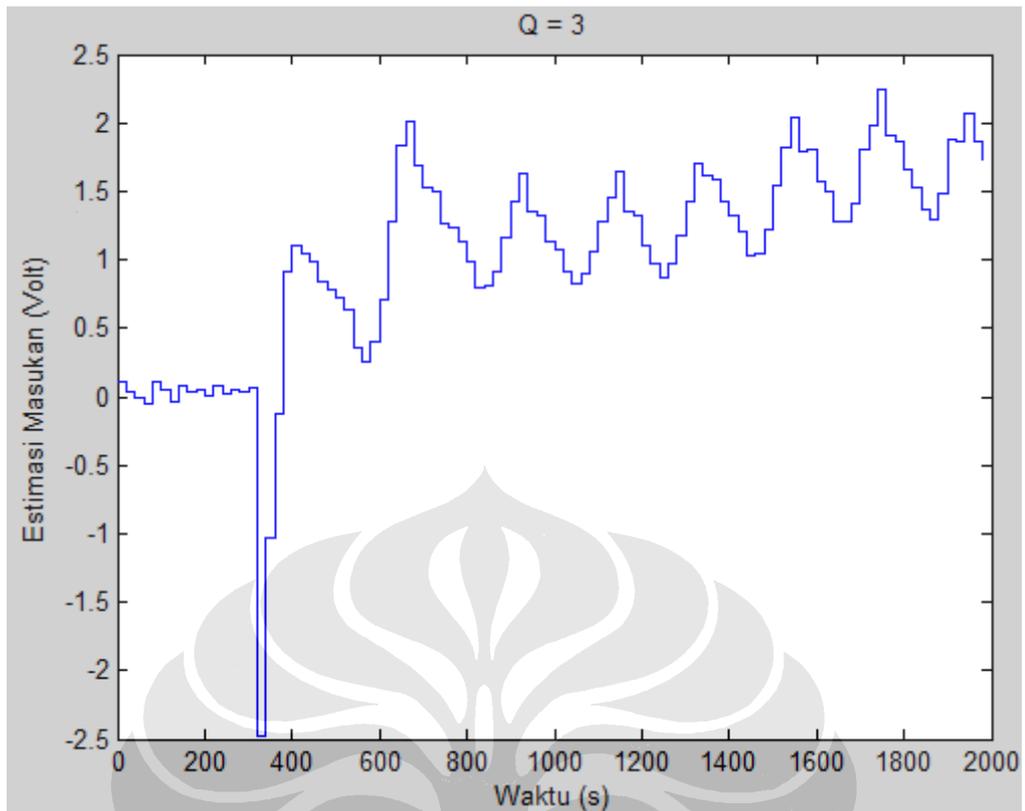


(a)

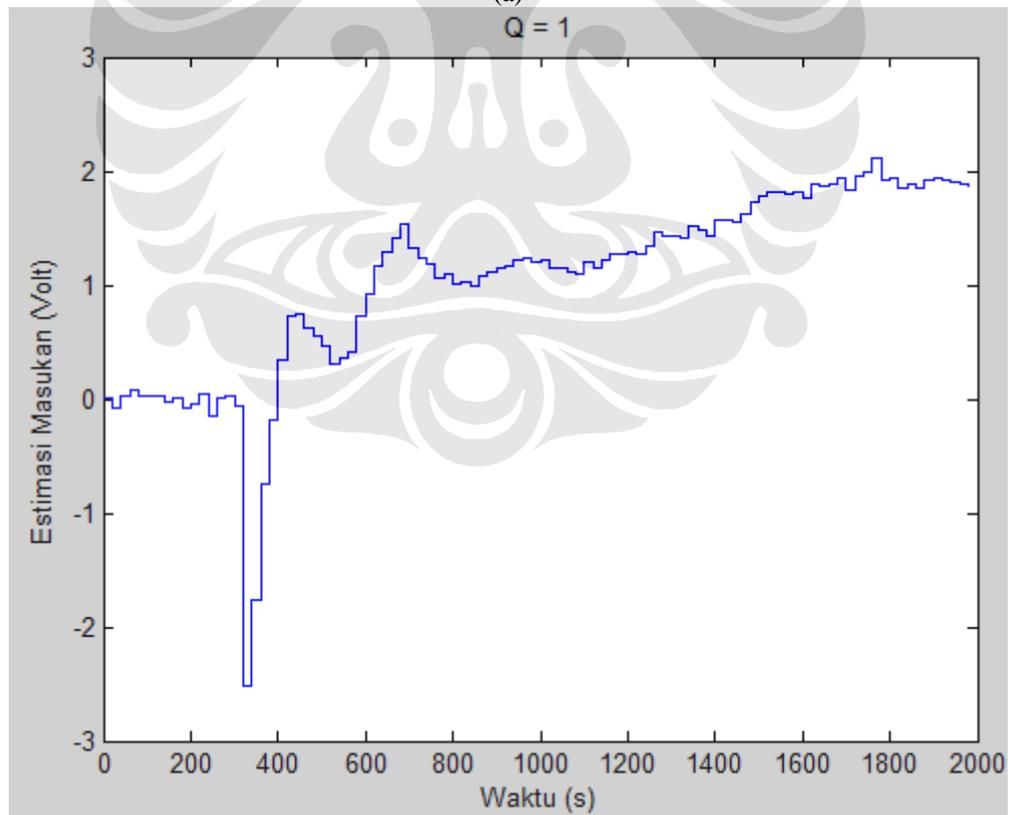


(b)

Gambar 4.23. Sinyal kendali hasil uji eksperimen untuk nilai matriks  $\underline{R}$  yang berbeda-beda.



(a)



(b)

Gambar 4.24. Estimasi masukan hasil uji eksperimen untuk nilai matriks  $R$  yang berbeda-beda.

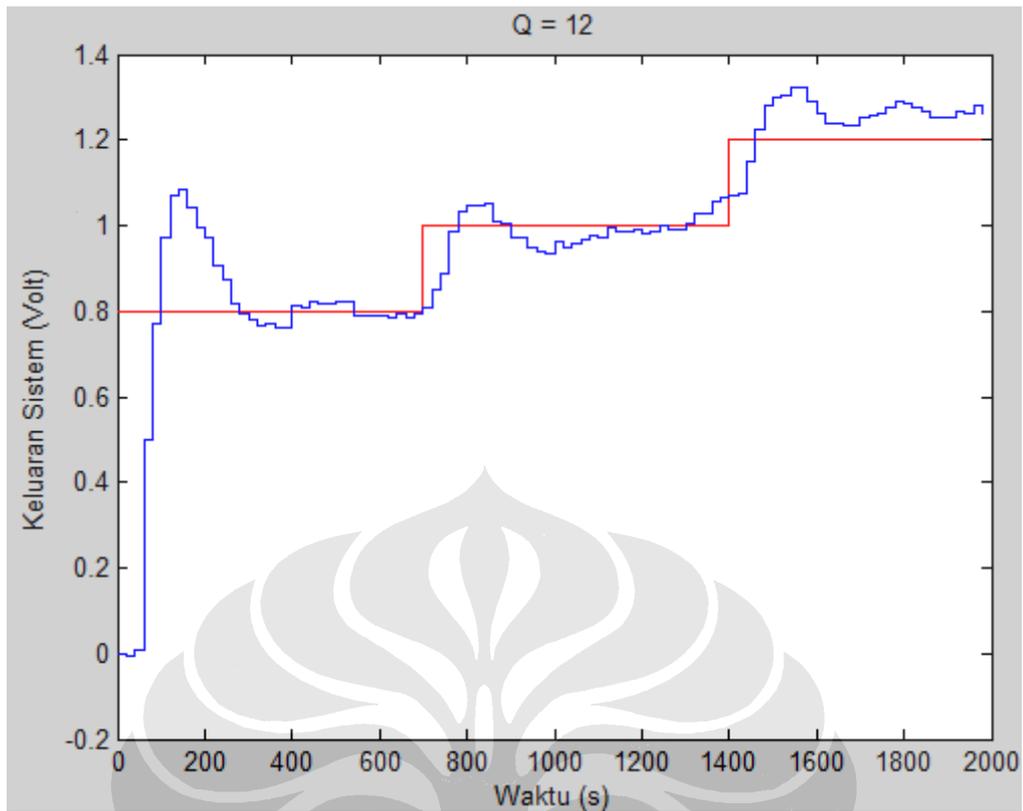
#### 4.5 UJI EKSPERIMEN PENGENDALI MPC WITH CONSTRAINT TANPA NILAI TRAYEKTORI ACUAN YANG AKAN DATANG

Pada ketiga uji eksperimen sebelumnya, nilai trayektori acuan untuk masa yang akan datang sudah diketahui. Pada ketiga uji eksperimen tersebut, fase pertama yang dilakukan sistem adalah mendeteksi trayektori acuan pada beberapa proses pencuplikan pertama, tergantung pada besarnya nilai *prediction horizon*. Pada fase ini, proses perhitungan sinyal kendali dengan menggunakan MPC *with constraint* tidak diaktifkan sehingga nilai sinyal kendali bernilai nol dan keluaran sistem menjadi minimum. Dengan diketahuinya trayektori acuan untuk masa yang akan datang, keluaran sistem dapat berubah terlebih dahulu sebelum terjadi perubahan trayektori acuan sehingga waktu yang dibutuhkan oleh keluaran sistem untuk mencapai trayektori acuan yang diinginkan menjadi cepat.

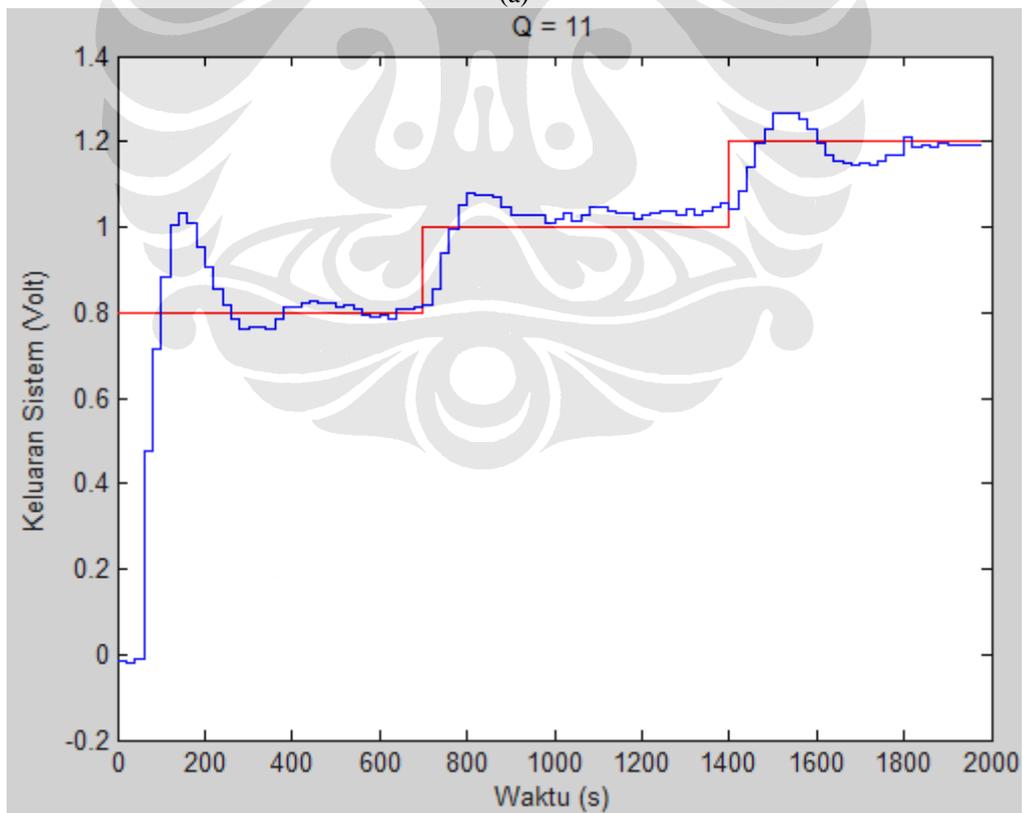
Uji eksperimen ini berbeda dengan keempat uji eksperimen sebelumnya, karena pada uji eksperimen ini sistem tidak mengetahui nilai trayektori yang akan datang dan hanya nilai trayektori acuan saat sekarang yang diketahui. Sehingga untuk mengatasi ketidaktahuan sistem pada trayektori acuannya, nilai trayektori acuan yang akan datang dianggap sama dengan nilai trayektori acuan sekarang.

Pada uji eksperimen ini, nilai matriks  $\underline{R}$  adalah  $12\underline{I}_{H_u}$ , *prediction horizon* dan *control horizon* yang digunakan masing-masing bernilai adalah 40 dan 2, sedangkan nilai matriks  $\underline{Q}$  yang digunakan adalah  $11\underline{I}_{H_p}$  dan  $12\underline{I}_{H_p}$ . Keluaran sistem hasil uji eksperimen ini dapat dilihat pada gambar 4.25 sedangkan sinyal kendali yang dihasilkan dari uji eksperimen terdapat pada gambar 4.26 dan hasil estimasi masukan  $x_1$  yang dihasilkan dari uji eksperimen terdapat pada gambar 4.27.

Dari uji eksperimen, ternyata keluaran sistem tetap bagus sedangkan untuk sinyal kendali hampir sama dengan uji eksperimen keempat, dimana semakin kecil nilai matriks  $\underline{Q}$ , perubahan sinyal kendali akan semakin ditekan dan sinyal kendali dapat berubah secara perlahan. Akibatnya, keluaran sistem hasil kendali dengan nilai matriks  $\underline{Q}$  yang lebih kecil akan menjadi lebih halus dibandingkan dengan keluaran sistem hasil kendali dengan nilai matriks  $\underline{Q}$  yang lebih besar seperti yang terlihat pada gambar 4.25.

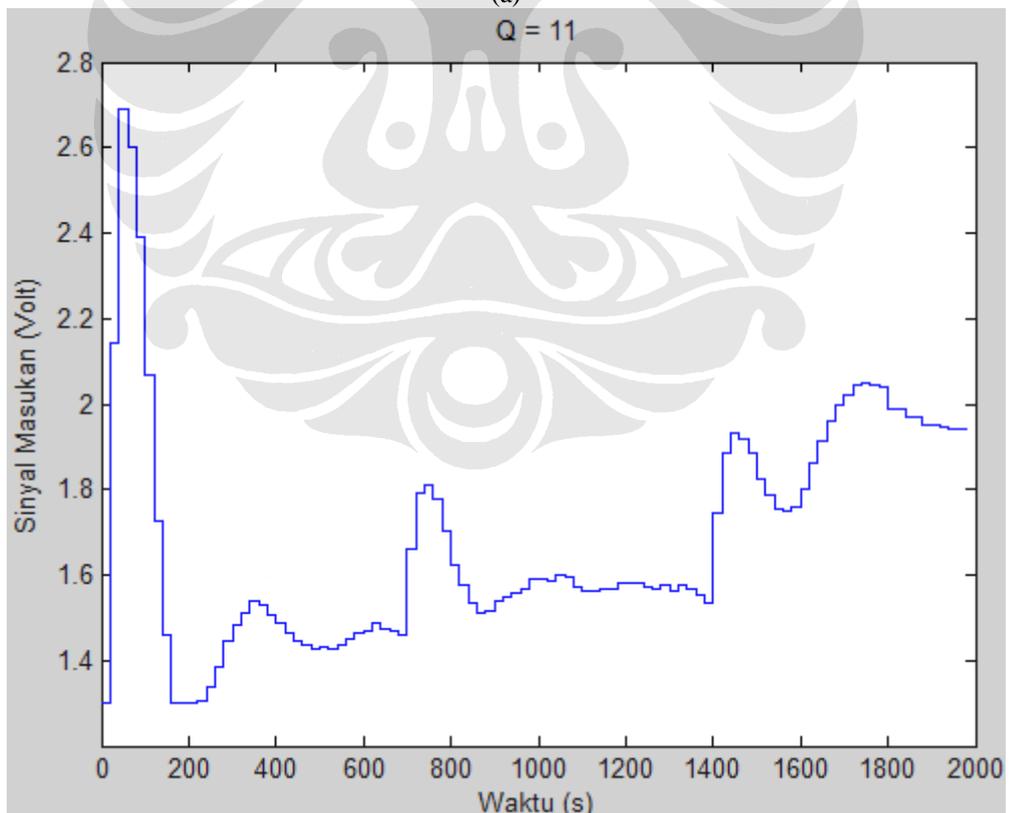
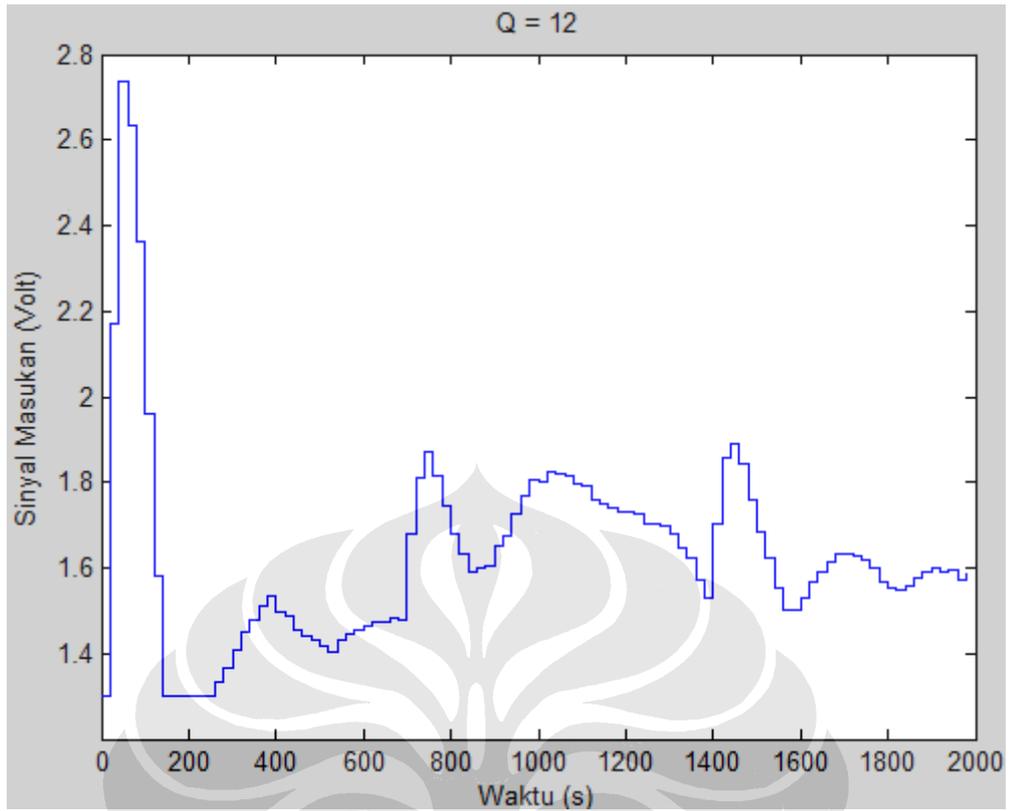


(a)

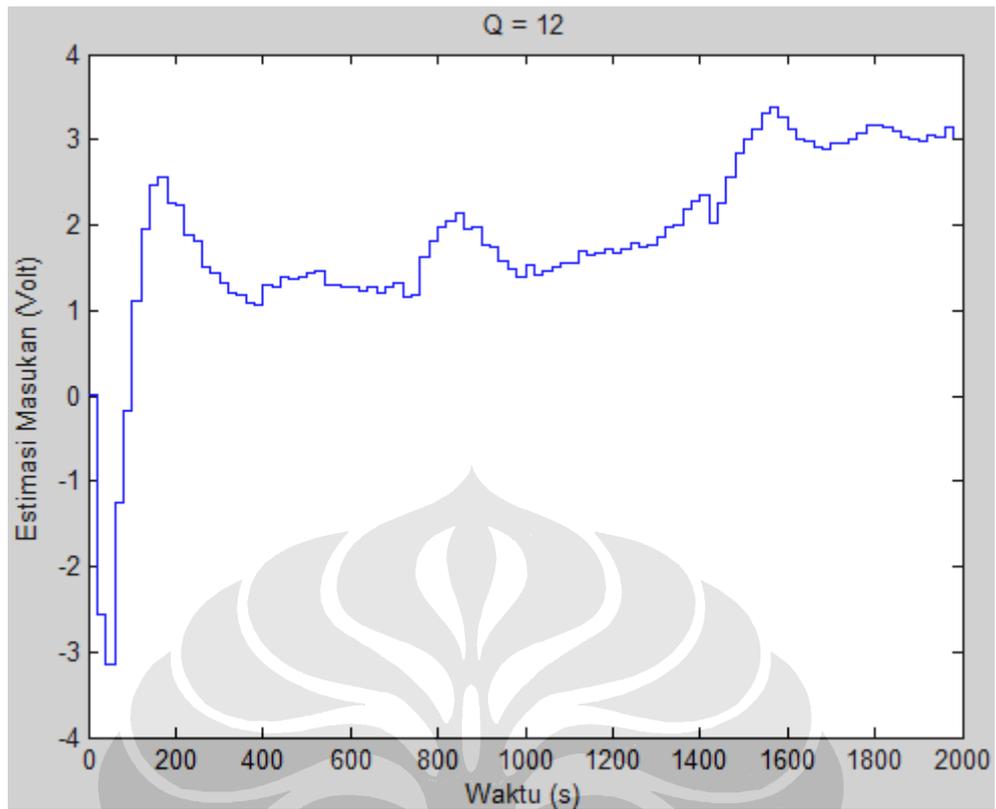


(b)

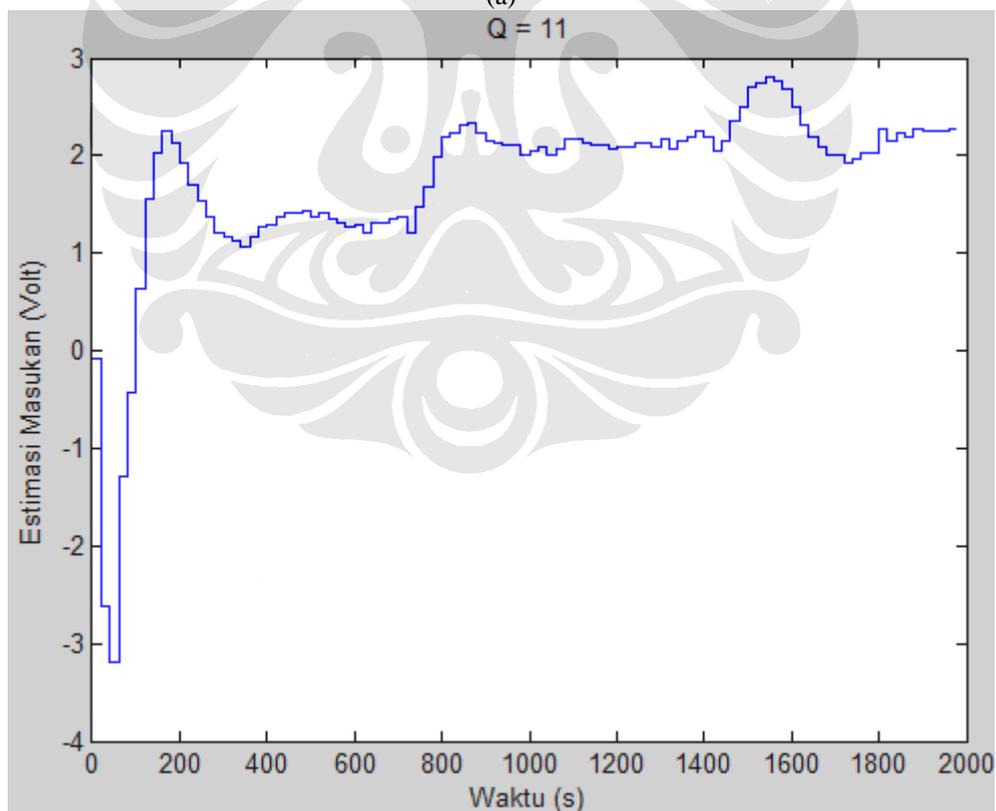
Gambar 4.25. Keluaran sistem hasil uji eksperimen dimana sistem tidak mengetahui nilai trayektori acuan untuk masa yang akan datang.



Gambar 4.26. Sinyal kendali hasil uji eksperimen dimana sistem tidak mengetahui nilai trayektori acuan untuk masa yang akan datang.



(a)



(b)

Gambar 4.27. Estimasi masukan hasil uji eksperimen dimana sistem tidak mengetahui nilai trayektori acuan untuk masa yang akan datang.

## 4.6 PERBANDINGAN KINERJA PENGENDALI METODE MPC WITH CONSTRAINT DAN METODE ATURAN KENDALI RUANG KEADAAN

### 4.6.5 Landasan Teori Aturan Kendali Ruang Keadaan

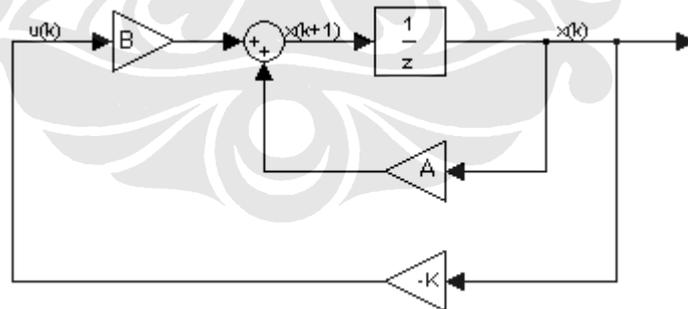
Persamaan ruang keadaan suatu sistem secara umum dapat dinyatakan dalam persamaan sebagai berikut :

$$x(k+1) = Ax(k) + Bu(k) \quad (4.1)$$

$$y(k) = Cx(k) + Du(k) \quad (4.2)$$

Pada dasarnya, metode pengendali ruang keadaan merupakan sebuah metode pengendali penempatan kutub dimana metode pengendaliannya dimulai dengan penentuan kutub-kutub sistem lingkaran tertutup yang didasarkan pada kebutuhan *transient response* dan/atau *frequency response* sistem seperti kecepatan, koefisien redaman atau *bandwidth*. Oleh karena itu, letak kutub-kutub lingkaran tertutup sistem harus ditentukan terlebih dahulu yaitu berada pada posisi  $z = \mu_1, z = \mu_2, \dots, z = \mu_n$ .

Berikut ini adalah blok diagram pengendali ruang keadaan lingkaran tertutup dengan *necessary and sufficient condition* :



Gambar 4.28. Pengendali ruang keadaan lingkaran tertutup

Sinyal kendali yang diberikan ke sistem sebesar  $u(k) = -Kx(k)$  dengan  $K$  adalah matriks penguat umpan balik keadaan dan persamaan keadaan sistem menjadi :

$$x(k+1) = (A - BK)x(k) \quad (4.3)$$

Matriks  $K$  harus dipilih sehingga membuat nilai *eigen* dari  $A-BK$  menjadi kutub-kutub lingkaran tertutup yang diinginkan,  $\mu_1, \mu_2, \dots, \mu_n$ .

Untuk menghitung besarnya matriks  $K$ , dapat digunakan formula *Ackermann* yaitu :

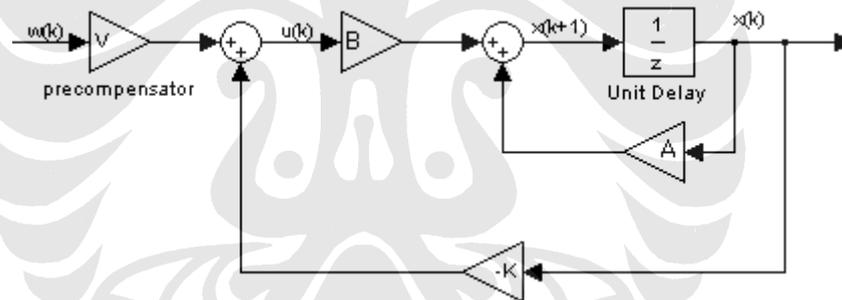
$$\begin{aligned} |zI - A + BK| &= (z - \mu_1)(z - \mu_2) \cdots (z - \mu_n) \\ &= z^n + \alpha_1 z^{n-1} + \alpha_2 z^{n-2} + \cdots + \alpha_{n-1} z + \alpha_n = 0 \end{aligned} \quad (4.4)$$

$$K = [0 \ 0 \ \cdots \ 0 \ 1] [B : AB : \cdots : A^{n-1} B]^{-1} \phi(A) \quad (4.5)$$

dimana

$$\phi(A) = A^n + \alpha_1 A^{n-1} + \cdots + \alpha_{n-1} A + \alpha_n I$$

Berikut ini adalah blok diagram pengendali ruang keadaan lingkaran tertutup dengan penguat *precompensator* :



Gambar 4.29. Pengendali ruang keadaan lingkaran tertutup dengan penguat *precompensator*

Sinyal kendali yang diberikan ke sistem sebesar :

$$u(k) = Vw(k) - Kx(k)$$

Persamaan ruang keadaan sistem menjadi :

$$x(k+1) = Ax(k) + B[Vw(k) - Kx(k)]$$

$$x(k+1) = [A - BK]x(k) + BVw(k)$$

dengan  $x(k+1) = x(k)$  saat kondisi *steady state*, maka

$$x(k)[I - A + BK] = BVw(k)$$

$$x(k) = [I - A + BK]^{-1} BVw(k) \quad (4.6)$$

Persamaan keluaran sistem adalah sebagai berikut :

$$y(k) = C[I - A + BK]^{-1} BVw(k) \quad (4.7)$$

dengan  $y(k) = w(k)$  saat kondisi *steady state*, maka

$$w(k) = C[I - A + BK]^{-1} BVw(k)$$

Sehingga persamaan penguat *precompensator* didapat sebagai berikut :

$$V = [C[I - A + BK]^{-1} B]^{-1} \quad (4.8)$$

#### 4.6.6 Uji Eksperimen dan Analisa

##### 4.6.2.1 Pengetesan *Controllability* dan Perancangan Pengendali

Persamaan ruang keadaan sistem yang dipakai adalah persamaan (3.15) Sebelum menerapkan pengendali ruang keadaan, pengetesan *controllability* sistem harus dilakukan dengan membentuk matriks *controllability* sebagai berikut :

$$\begin{bmatrix} B & AB & \dots & (A)^{n-1} B \end{bmatrix} \quad (4.9)$$

dimana :

- n adalah jumlah *state* yang dimiliki oleh sebuah sistem.
- sistem *controllable* jika matriks *controllability* memiliki *rank* sebanyak n (jumlah state)

Berdasarkan persamaan (4.9) maka matriks *controllability* dari *Coupled-Tank Basic Process Rig 38-100* adalah sebagai berikut :

$$\begin{bmatrix} 0.5399 & 0.0982 \\ 0.5337 & 0.0975 \end{bmatrix}$$

*Rank* dari matriks *controllability* diatas adalah 2 sehingga sistem *Coupled-Tank Basic Process Rig 38-100* dikatakan *fully controllable*.

Nilai eigen dari persamaan ruang keadaan sistem adalah :

$$eig = \begin{bmatrix} 0.1822 \\ -0 \end{bmatrix}$$

Letak kutub-kutub lingkaran tertutup sistem yang diinginkan adalah :

$$desired\ poles = \begin{bmatrix} 0.1 \\ 0 \end{bmatrix}$$

Matriks penguat umpan balik keadaan diperoleh dengan menggunakan persamaan 4.5 atau dengan menggunakan perintah *Acker* pada MATLAB 7.0. dan diperoleh nilai sebagai berikut :

$$K = [0.0913 \quad 0.0617]$$

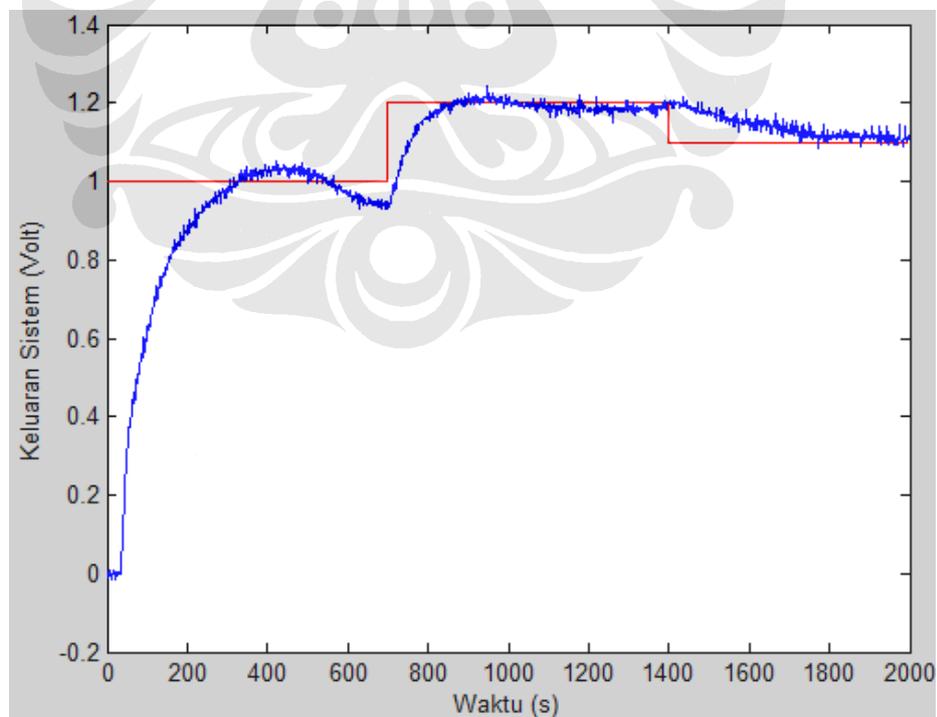
Sedangkan penguat *precompensator* diperoleh dengan menggunakan persamaan 4.8. dan diperoleh nilai sebagai berikut :

$$V = 1.6854$$

#### 4.6.2.2 Uji Eksperimen Pengendali Ruang Keadaan

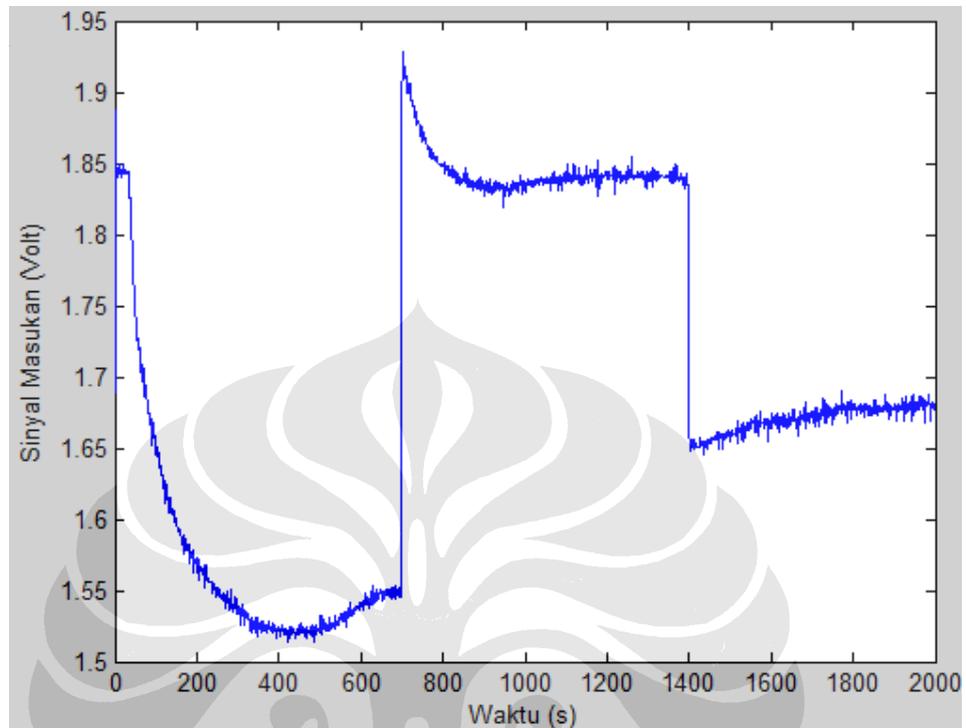
Parameter-parameter pengendali ruang keadaan yang telah didapat akan diterapkan pada sistem *Coupled-Tank Basic Process Rig 38-100*. Blok SIMULINK yang dipakai untuk uji eksperimen pengendali ruang keadaan dilampirkan pada lampiran L.3.

Berikut ini hasil keluaran uji eksperimen pengendali ruang keadaan dengan dua perubahan trayektori acuan terdapat pada gambar 4.30.



Gambar 4.30. Hasil keluaran uji eksperimen pengendali ruang keadaan dengan dua perubahan trayektori acuan

Berikut ini sinyal kendali uji eksperimen pengendali ruang keadaan dengan dua perubahan trayektori acuan terdapat pada gambar 4.31.



Gambar 4.31. Sinyal kendali uji eksperimen pengendali ruang keadaan dengan dua perubahan trayektori acuan

Setelah melakukan uji eksperimen pengendali ruang keadaan yaitu uji eksperimen dengan dua perubahan trayektori acuan, kinerja pengendali ruang keadaan dapat dikatakan cukup baik karena pengendali dapat membuat sistem dapat mengikuti nilai masukan yang diberikan. Hanya saja di bagian awal pemberian nilai masukan, keluaran sistem tampak belum sempat mengikuti trayektori acuan dengan baik. Hal ini disebabkan oleh tidak cukupnya waktu bagi sistem untuk mencapai trayektori acuan sebesar 1 yaitu selama 750 detik. Selisih ini sebenarnya sudah diatasi oleh perlakuan sinyal kendali yang diberikan pengendali ruang keadaan dimana pengendali memberikan reaksi seperti penurunan sinyal kendali ketika keluaran telah melebihi masukan dan juga sebaliknya.

Walaupun keluaran sistem sudah baik, metode pengendali ruang keadaan masih memiliki kekurangan yaitu waktu yang dibutuhkan untuk stabil lebih lama

daripada MPC dan tidak dapat diperhitungkannya besar perubahan sinyal kendali dan batasan sinyal kendali pada proses pengendalian seperti pada MPC. Akibatnya, variansi perubahan sinyal kendali menjadi cukup besar (gambar 4.31.) dan perubahan sinyal kendali terlalu besar ketika terjadi perubahan trayektori acuan. Besarnya sinyal kendali dapat dibatasi dengan menggunakan blok saturasi sehingga sinyal kendali yang masuk ke *plant* akan dipotong jika melebihi tegangan maksimum atau tegangan minimum yang diperbolehkan. Jika sinyal kendali yang masuk ke *plant* terus-menerus dipotong, maka akan membuat hasil kendali menjadi tidak bagus. Karena pada uji eksperimen ini sinyal kendali yang terpotong tidak ada, maka keluaran sistem hasil pengendalian menjadi cukup baik.



## BAB 5

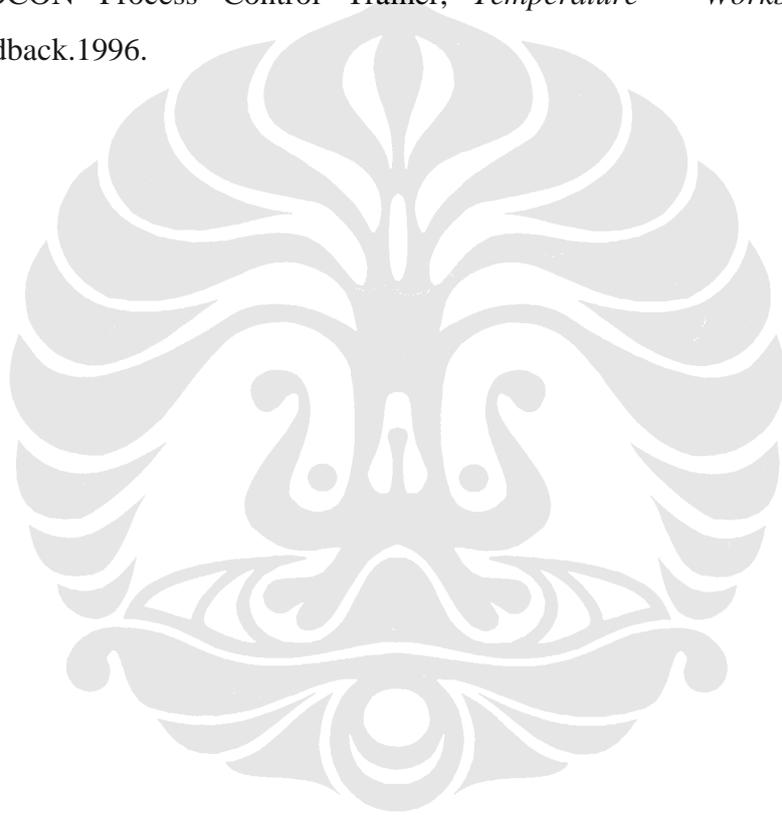
### KESIMPULAN

Dari keseluruhan pembahasan dalam skripsi ini dapat disimpulkan beberapa hal, yaitu :

1. Keluaran sistem hasil pengendalian MPC dengan *constraints* dapat bereaksi mengikuti trayektori acuan sebelum nilai trayektori acuan berubah. Hal ini terjadi jika nilai trayektori acuan untuk masa yang akan datang diketahui.
2. Pemakaian *reduced-order observer* untuk mengestimasi variabel keadaan sistem dapat menggantikan keterbatasan *hardware* sistem, dalam hal ini yaitu kekurangan sensor pada tangki pertama tetapi hasil estimasi yang dikumpulkan tidak sebaik bila menggunakan data yang dikumpulkan secara langsung oleh sensor.
3. Penggeseran letak kutub *observer* lingkaran tertutup sebaiknya tidak terlalu besar dan digeser mendekati sumbu 0. Pada sistem ini terjadi penggeseran dari nilai *eigen* sebesar 0,1822 menjadi 0,1.
4. Keluaran sistem hasil pengendalian MPC dengan *constraints* akan semakin bagus jika nilai *prediction horizon* dibuat sebesar mungkin atau dengan kata lain nilai *control horizon* menjauhi nilai *prediction horizon*.
5. Semakin besar nilai faktor bobot perubahan sinyal kendali  $R$ , maka perubahan sinyal kendali dapat semakin ditekan sehingga keluaran sistem menjadi semakin halus.
6. Sebaliknya untuk besar nilai faktor bobot kesalahan  $Q$ , jika nilainya semakin kecil maka perubahan sinyal kendali dapat semakin ditekan sehingga keluaran sistem menjadi semakin halus.
7. Untuk sistem ini didapat kombinasi terbaik dari pengaturan *prediction horizon*, *control horizon*, faktor bobot perubahan sinyal kendali dan faktor bobot kesalahan yaitu  $H_p = 30$ ,  $H_u = 2$ ,  $R = I_{H_u}$  dan  $Q = I_{H_p}$ .
8. Metode MPC dengan *constraints* dapat menghasilkan keluaran yang lebih baik dibandingkan dengan metode Aturan Kendali Ruang Keadaan karena pada MPC *with constraints* tidak akan terjadi perubahan yang drastis pada sinyal kendali dan pemotongan paksa pada sinyal kendali.

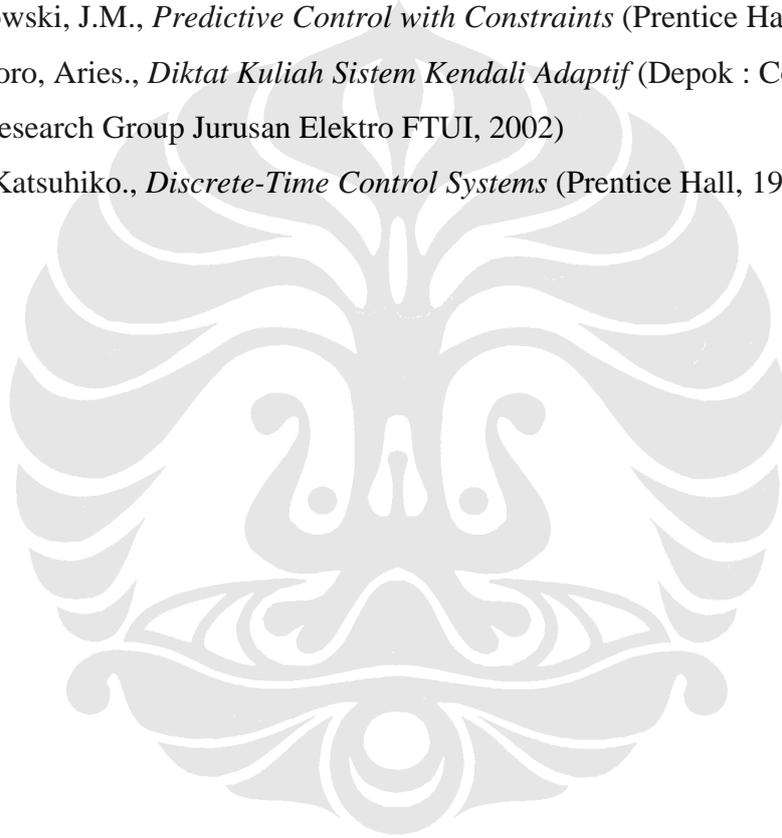
## DAFTAR ACUAN

- [1] Aries Subiantoro, *Diktat Kuliah Sistem Kendali Adaptif* (Depok : Control System Research Group Jurusan Elektro FTUI, 2002)
- [2] E.F. Camacho, C. Bordons, *Model Predictive Control* (Springer-Verlag, 1999)
- [3] J. M. Maciejowski, *Predictive Control with Constraints* (Prentice Hall, 2002)
- [4] Ogata, Katsuhiko, *Discrete-Time Control Systems* (Prentice Hall, 1995)
- [5] PROCON Process Control Trainer, *Temperature – Workbook 38-002*. Feedback.1996.

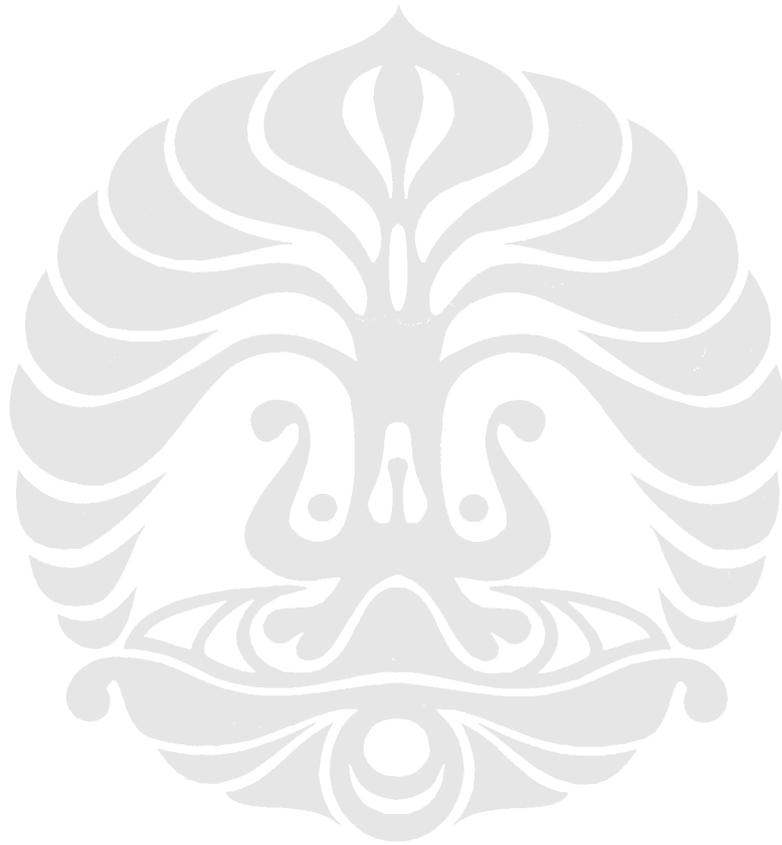


## DAFTAR PUSTAKA

- Camacho, E.F., C. Bordons, *Model Predictive Control* (Springer-Verlag, 1999)
- Kristiawan, Antonius Yuda., "Aplikasi *Model Predictive Control* dengan *Constraints* Sinyal Kendali Berbasis Algoritma *Active Set* pada Pengendalian *COUPLED-TANK CONTROL APPARATUS PP-100*" Skripsi, Program Sarjana Fakultas Teknik UI, Depok, 2007.
- Maciejowski, J.M., *Predictive Control with Constraints* (Prentice Hall, 2002)
- Subiantoro, Aries., *Diktat Kuliah Sistem Kendali Adaptif* (Depok : Control System Research Group Jurusan Elektro FTUI, 2002)
- Ogata, Katsuhiko., *Discrete-Time Control Systems* (Prentice Hall, 1995)

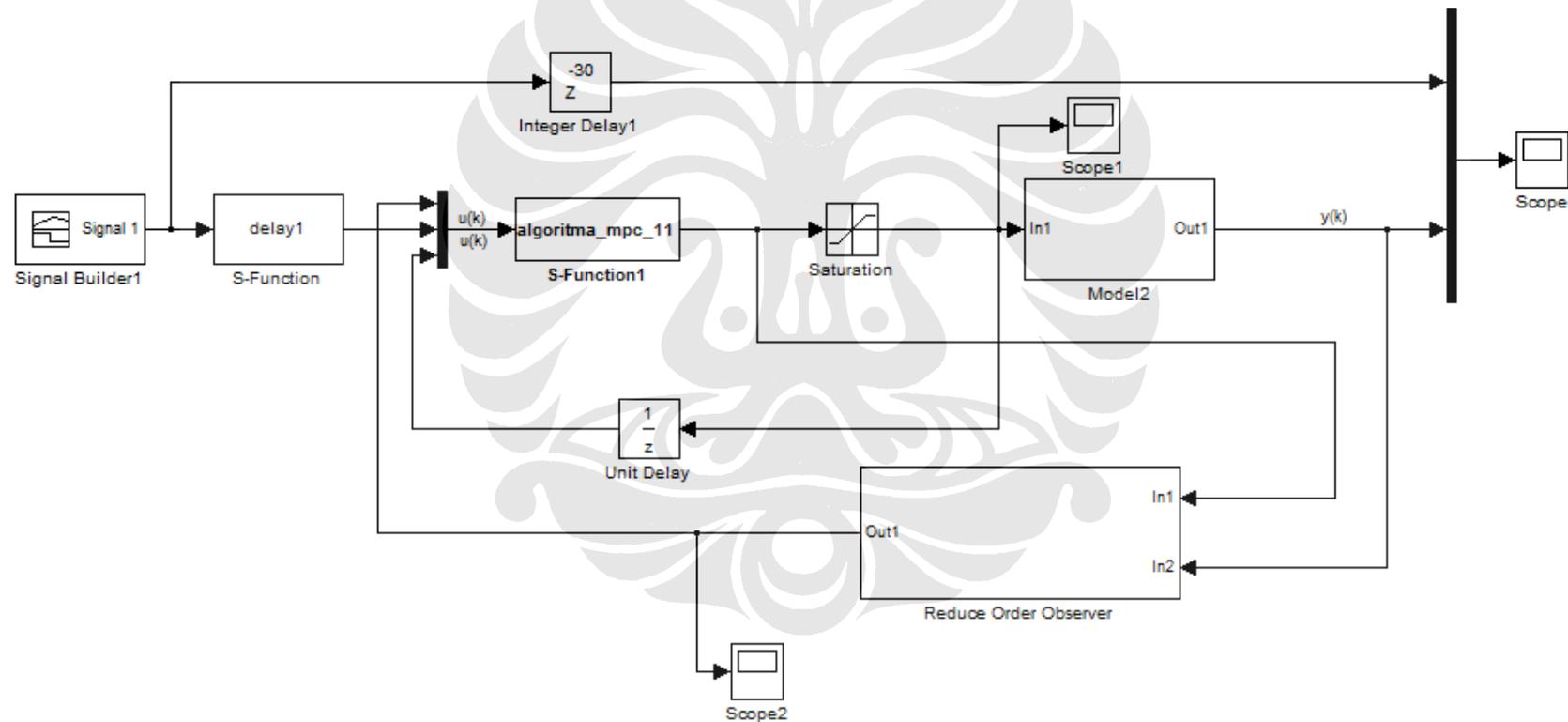


## LAMPIRAN



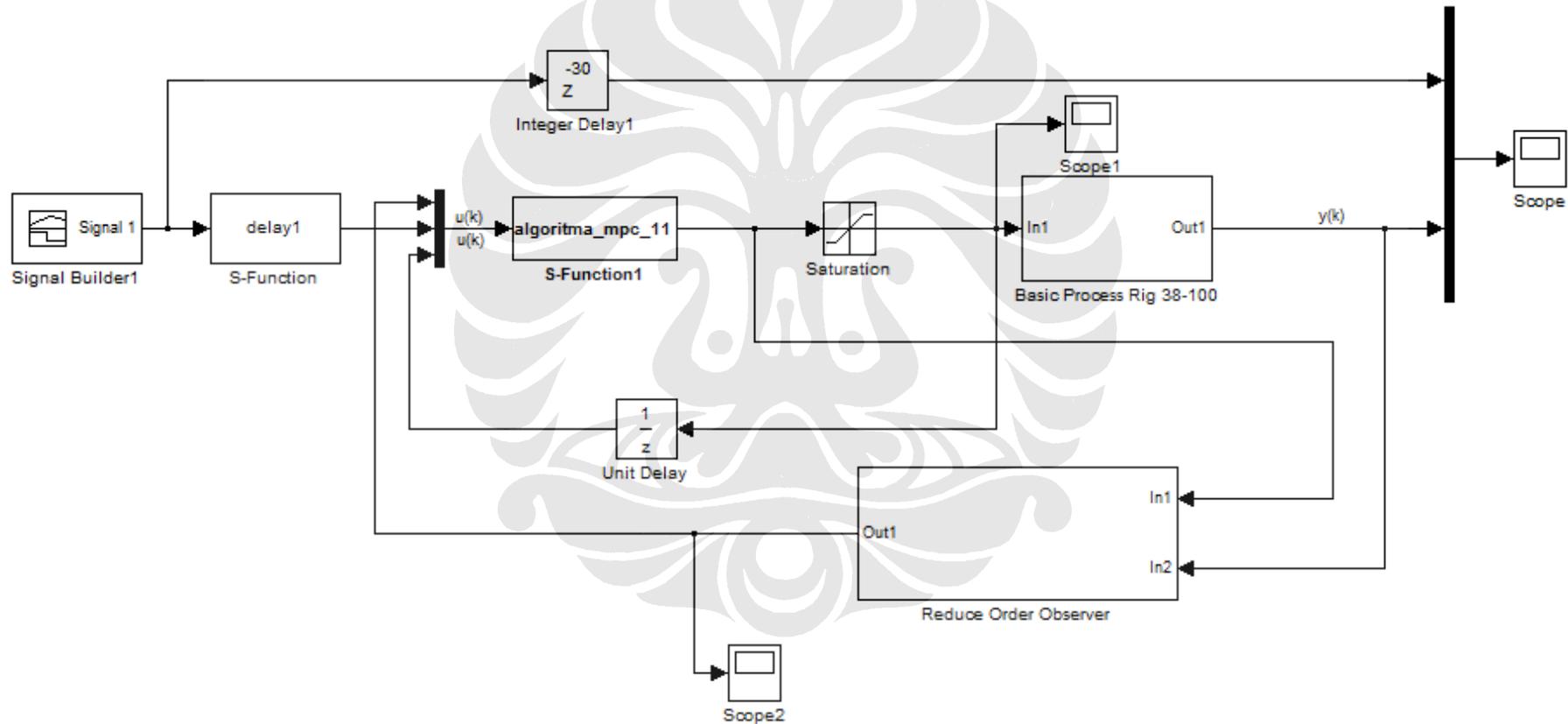
## Lampiran 1. Blok SIMULINK

### 1. Blok SIMULINK pada Simulasi Pengendalian Menggunakan Metode MPC *with Constraints*



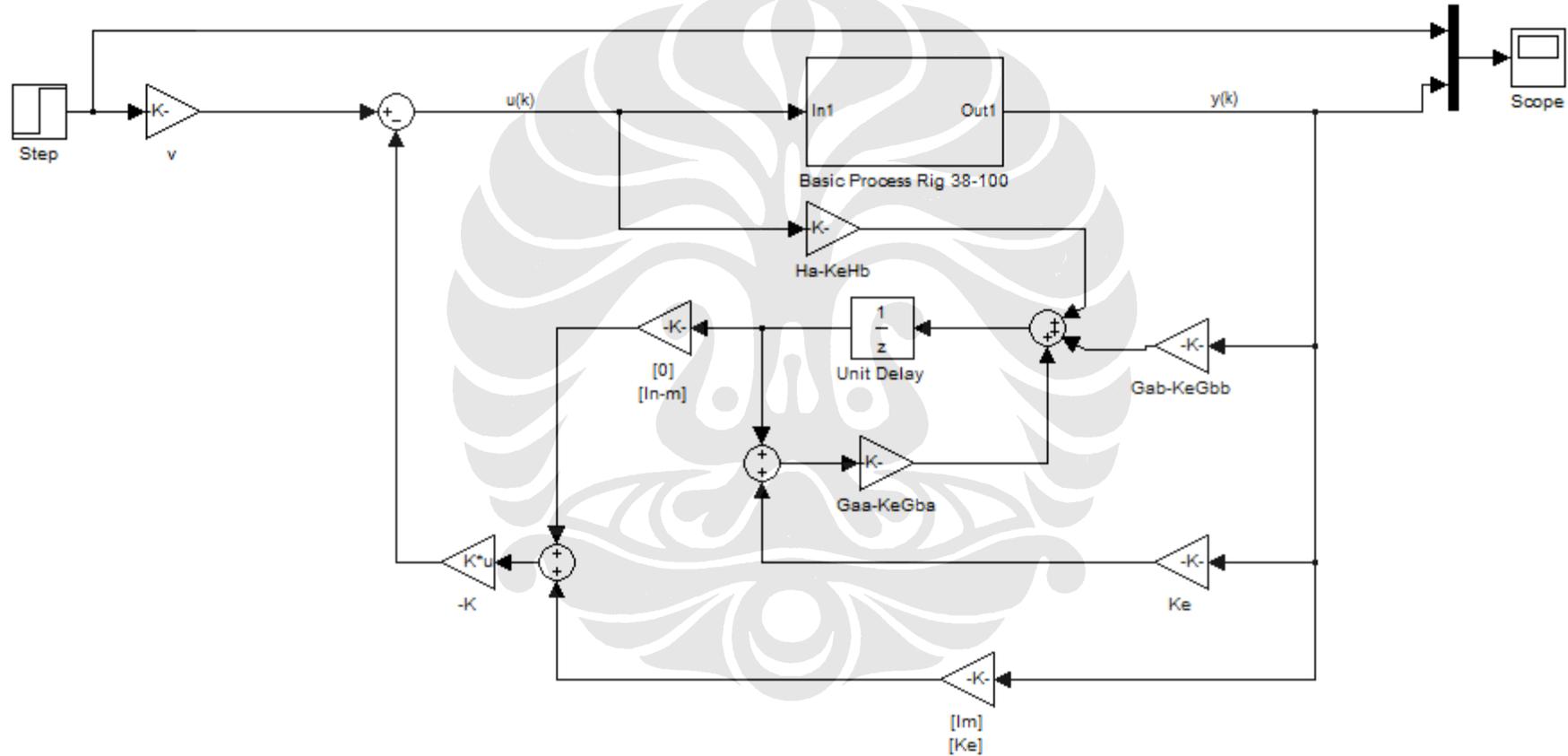
**Gambar L.1.** Gambar blok SIMULINK simulasi pengendalian pada model *Coupled-Tank Basic Process Rig 38-100* menggunakan metode MPC dengan *constraints*.

2. Blok SIMULINK pada Uji eksperimen Pengendalian Menggunakan Metode MPC *with Constraints*



**Gambar L.2.** Gambar blok SIMULINK uji eksperimen pengendalian pada *Coupled-Tank Basic Process Rig 38-100* menggunakan metode MPC dengan *constraints*.

3. Blok SIMULINK pada Uji eksperimen Pengendalian Menggunakan Metode Aturan Kendali Ruang Keadaan



**Gambar L.3.** Gambar blok SIMULINK uji eksperimen pengendalian pada *Coupled-Tank Basic Process Rig 38-100* menggunakan metode Aturan Kendali Ruang Keadaan.