



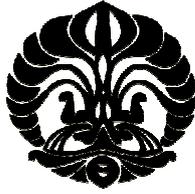
UNIVERSITAS INDONESIA

**ESTIMASI VARIABEL KEADAAN PADA SISTEM TATA
UDARA PRESISI MENGGUNAKAN ALGORITMA *MOESP*
(*MIMO OUTPUT-ERROR STATE-SPACE MODEL
IDENTIFICATION*)**

SKRIPSI

**NIKODEMUS JOKO E.M
0806331166**

**FAKULTAS TEKNIK
DEPARTEMEN TEKNIK ELEKTRO
DEPOK
JULI 2012**



UNIVERSITAS INDONESIA

**ESTIMASI VARIABEL KEADAAN PADA SISTEM TATA
UDARA PRESISI MENGGUNAKAN ALGORITMA *MOESP*
(*MIMO OUTPUT-ERROR STATE-SPACE MODEL
IDENTIFICATION*)**

SKRIPSI

Diajukan sebagai salah satu syarat memperoleh gelar sarjana.

**NIKODEMUS JOKO E.M
0806331166**

**FAKULTAS TEKNIK
DEPARTEMEN TEKNIK ELEKTRO
DEPOK
JULI 2012**

HALAMAN PERNYATAAN ORISINALITAS

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

Nama : Nikodemus Joko E.M

NPM : 0806331166

Tanda Tangan : 

Tanggal : 3 Juli 2012

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Nikodemus Joko E.M
NPM : 0806331166
Program Studi : Teknik Elektro
Judul Skripsi : Estimasi Variabel Keadaan pada Sistem Tata
Udara Presisi Menggunakan Algoritma MOESP
(*MIMO Output-Error State-Space Model
Identification*)

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

DEWAN PENGUJI

Pembimbing : Ir. Aries Subiantoro, M.SEE (.....)
Penguji : Dr. Ir. Ridwan Gunawan, M.T (.....)
Penguji : Dr. Ir. Feri Yusivar, M.Eng (.....)

Ditetapkan di : Depok
Tanggal : 3 Juli 2012

KATA PENGANTAR

Pertama-tama segala puji dan syukur patutlah dipanjatkan kepada Tuhan atas segala berkat-Nya, baik berupa kesempatan, waktu, dan kemampuan yang penulis terima untuk dapat mendukung penulisan laporan skripsi ini. Adapun penulisan laporan skripsi ini merupakan salah satu syarat untuk mencapai gelar S1 Program Studi Teknik Elektro Universitas Indonesia.

Selain itu, penulis juga ingin mengucapkan terima kasih kepada orang-orang yang telah mendukung dalam penulisan laporan skripsi ini, terutama kepada :

1. Ir. Aries Subiantoro, M.SEE sebagai dosen pembimbing yang telah dengan sabar membimbing penulis dalam penyusunan laporan skripsi ini.
2. Nur Hidayat, S.T yang telah mengajari banyak bagaimana cara mengoperasikan alat sistem tata udara presisi serta cara kerjanya.
3. Victor Fonso, S.T yang telah mengajari banyak hal tentang identifikasi sistem dan sistem tata udara presisi.
4. Orang tua dan keluarga saya yang selalu mendukung di dalam doa.
5. Teman-teman PO dan PMKJ yang selalu memberikan dukungan doa dan semangat supaya saya bisa menyelesaikan penelitian dan penulisan laporan ini dengan tepat waktu.
6. Teman-teman Teknik Elektro yang sama-sama berjuang menyelesaikan skripsi masing-masing, yang juga saling memberikan semangat.

Akhir kata, penulis sangat berharap laporan skripsi ini berguna bagi pembaca dan pengembangan ilmu pengetahuan di Universitas Indonesia.

Depok, Juli 2012

Penulis

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

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

Nama : Nikodemus Joko E.M

NPM : 0806331166

Program Studi : Elektro

Departemen : Elektro

Fakultas : Teknik

Jenis Karya : Skripsi

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

Estimasi Variabel Keadaan pada Sistem Tata Udara Presisi Menggunakan
Algoritma *MOESP* (*MIMO Output-Error State-Space Model Identification*)

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

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada Tanggal : 3 Juli 2012

Yang menyatakan



(Nikodemus Joko E.M)

ABSTRAK

Nama : Nikodemus Joko E.M
Program Studi : Teknik Elektro
Judul : Estimasi Variabel Keadaan pada Sistem Tata Udara Presisi Menggunakan Algoritma *MOESP (MIMO Output-Error State-Space Model Identification)*

Sistem tata udara presisi atau yang lebih dikenal dengan *Precision Air Conditioning (PAC)* merupakan mesin refrigerasi yang dapat mengendalikan temperatur dan kelembaban dari suatu data center sesuai dengan yang diinginkan, yaitu pada temperatur 20-25°C dan kelembaban relatif (RH) 40-55%. Pengendali yang digunakan adalah pengendali cerdas yang mengacu pada model *state space*. Pengendali jenis ini membutuhkan nilai *state* dalam desainnya. Namun pada realisasinya, sulit untuk menentukan *state* sistem tata udara presisi ini karena berbagai macam faktor, seperti keterbatasan peralatan elektronik, serta alasan ekonomi.

Dalam kegiatan penelitian ini, akan diimplementasikan suatu metode yang digunakan untuk mengestimasi *state*. Metode yang digunakan adalah metode yang menggunakan algoritma identifikasi MOESP secara rekursif. Melalui proses dekomposisi nilai tunggal dan perhitungan matematis sederhana, akan diperoleh informasi estimasi *state* untuk waktu sekarang $\hat{x}_{N|N}$ dan estimasi *state* untuk satu langkah ke depan pada waktu sekarang $\hat{x}_{N+1|N}$. Proses validasi dilakukan cara membandingkan hasil estimasi *state* dengan *state* keluaran dari model yang diperoleh dari hasil identifikasi *offline*.

Hasil estimasi *state* terbaik diperoleh melalui sistem tata udara presisi yang bersifat linier. Hasil estimasi *state* sangat bergantung dari model sistem yang diperoleh.

Kata kunci: MOESP, rekursif, variabel keadaan..

ABSTRACT

Name : Nikodemus Joko E.M
Study Program : Electrical Engineering
Title : State Estimation in Precision Air Conditioning System Using *MOESP (MIMO Output-Error State-Space Model Identification)* Algorithm

Precision Air Conditioning (PAC) is a refrigeration machine that can control temperature and humidity of a data center. The ideal temperature and Relative Humidity (RH) is 20-25°C and 40-55%. The controller used is a smart controller that refers to the state space model. Controller of this type requires the value of states and parameters needed in the designing. But in its realization, it is difficult to determine the accurate states and parameters of the Precision Air Conditioning because of variety of factors, such as the limitations of electronic equipment, as well as economic reasons. In this research, will implement a method used to estimate the state. The method used is a method that uses a recursive identification algorithm MOESP.

Through the Single Value Decomposition (SVD) process and simple mathematical calculations, it will obtain the current state estimation $\hat{x}_{M|N}$ and state estimation for the next step at the present time $\hat{x}_{N+1|N}$. The validation process carried out to compare the estimated state with state from model obtained from offline identification.

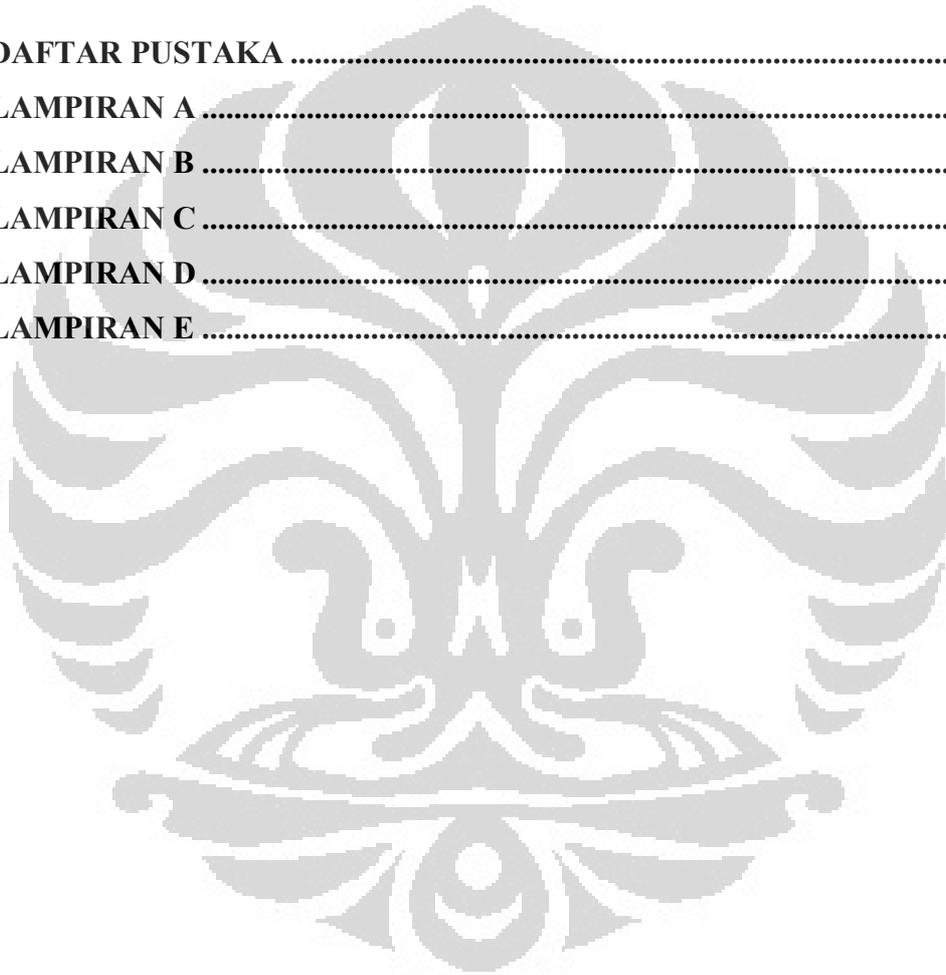
The best state estimation obtained through the linier system of Precision Air Conditioning value. State estimation results are dependent of the system model that is obtained.

Key words: MOESP, recursive, state.

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN	iii
KATA PENGANTAR	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	3
1.4 Pembatasan Masalah	3
1.5 Metodologi Penelitian	3
1.6 Sistematika Penulisan.....	4
BAB 2 DASAR TEORI	5
2.1 Algoritma MOESP Rekursif	5
2.2 MOESP rekursif dengan Gangguan Terukur	9
2.3 Algoritma untuk Mengestimasi Variabel Keadaan (<i>state</i>).....	10
2.4 Proses Estimasi Variabel Keadaan (<i>state</i>) pada Model MIMO	11
2.5 Sistem Tata Udara Presisi	15
BAB 3 ESTIMASI VARIABEL KEADAAN SISTEM TATA UDARA	
PRESISI	17
3.1 Persamaan Matematis Variabel Keadaan Sistem Tata Udara Presisi.....	17
3.2 Penentuan pasangan data (u_N, y_N)	24
3.3 Pengujian Algoritma untuk Mengestimasi Variabel Keadaan	25
3.4 Implementasi Algoritma untuk Mengestimasi Variabel Keadaan	28
BAB 4 SIMULASI DAN ANALISIS	32
4.1 Estimasi Variabel Keadaan dengan Identifikasi MOESP Rekursif.....	32

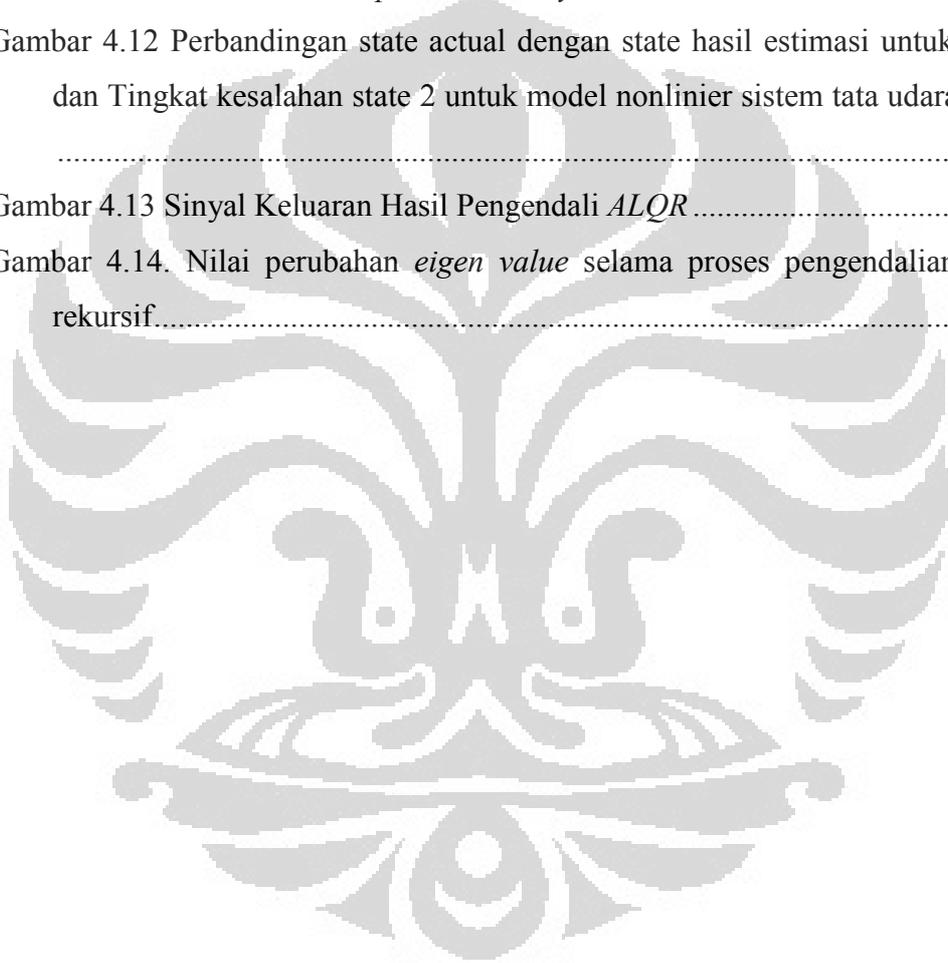
4.1.1	Model Linier Sistem Tata Udara Presisi.....	33
4.1.2	Estimasi state Model Linier Sistem Tata Udara Presisi dengan Variasi noise	40
4.1.3	Estimasi state Model nonlinier Sistem Tata Udara Presisi	43
4.2	Implementasi Algoritma dengan Desain Pengendali	44
BAB V KESIMPULAN.....		47
5.1	Kesimpulan	47
5.2	Saran.....	47
DAFTAR PUSTAKA		48
LAMPIRAN A		49
LAMPIRAN B		51
LAMPIRAN C		56
LAMPIRAN D		61
LAMPIRAN E		66



DAFTAR GAMBAR

Gambar 2.1. Algoritma MOESP Rekursif.....	8
Gambar 2.2 Bagan Sistem Tata Udara Presisi.....	16
Gambar 3.1 Skema Aliran Udara di Kondenser Kedua	19
Gambar 3.2 Skema Aliran Udara di Evaporator dan Kondenser Kedua.....	20
Gambar 3.3 Skema Aliran Udara dan Refrigeran di Evaporator.....	21
Gambar 3.4 Blok <i>Simulink</i> dari Sistem Tata Udara Presisi.....	23
Gambar 3.5 Sinyal Masukan dengan <i>Initial Seed</i> yang Berbeda	24
Gambar 3.6 Skema Pengujian dan Validasi Algoritma untuk Mengestimasi <i>State</i>	26
Gambar 3.7. Blok <i>Simulink</i> untuk Mendapatkan Data Masukan dan Keluaran Model Linier Sistem Tata Udara Presisi	28
Gambar 3.8 Diagram Blok Implementasi Algoritma untuk Mengestimasi Variabel Keadaan	28
Gambar 4.1 Perbandingan <i>state actual</i> dengan <i>state</i> hasil estimasi untuk <i>state</i> 1 dan Tingkat kesalahan <i>state</i> 1.....	33
Gambar 4.2 Perbandingan <i>state actual</i> dengan <i>state</i> hasil estimasi untuk <i>state</i> 2 dan Tingkat kesalahan <i>state</i> 2.....	34
Gambar 4.3 Perbandingan <i>state actual</i> dengan <i>state</i> hasil estimasi untuk <i>state</i> 3 dan Tingkat kesalahan <i>state</i> 3.....	35
Gambar 4.4 Perbandingan <i>state actual</i> dengan <i>state</i> hasil estimasi untuk <i>state</i> 4 dan Tingkat kesalahan <i>state</i> 4.....	36
Gambar 4.5 Perbandingan <i>state actual</i> dengan <i>state</i> hasil estimasi untuk <i>state</i> 5 dan Tingkat kesalahan <i>state</i> 5.....	37
Gambar 4.6 Perbandingan <i>state actual</i> dengan <i>state</i> hasil estimasi untuk <i>state</i> 6 dan Tingkat kesalahan <i>state</i> 6.....	38
Gambar 4.7 Perbandingan <i>state actual</i> dengan <i>state</i> hasil estimasi untuk <i>state</i> 7 dan Tingkat kesalahan <i>state</i> 7.....	39

Gambar 4.8 Perbandingan <i>state actual</i> dengan <i>state</i> hasil estimasi untuk <i>state</i> 8 dan Tingkat kesalahan <i>state</i> 8.....	40
Gambar 4.9. Perbandingan <i>state actual</i> dengan <i>state</i> hasil estimasi dan Tingkat kesalahan <i>state</i> 1 untuk <i>Spectral Density Gaussian Noise</i> sebesar 0.0001 ...	41
Gambar 4.10. Perbandingan <i>state actual</i> dengan <i>state</i> hasil estimasi dan Tingkat kesalahan <i>state</i> 1 untuk <i>Spectral Density Gaussian Noise</i> sebesar 0.01	41
Gambar 4.11. Perbandingan <i>state actual</i> dengan <i>state</i> hasil estimasi dan Tingkat kesalahan <i>state</i> 1 untuk <i>Spectral Density Gaussian Noise</i> sebesar 0.1	42
Gambar 4.12 Perbandingan <i>state actual</i> dengan <i>state</i> hasil estimasi untuk <i>state</i> 2 dan Tingkat kesalahan <i>state</i> 2 untuk model nonlinier sistem tata udara presisi	43
Gambar 4.13 Sinyal Keluaran Hasil Pengendali <i>ALQR</i>	45
Gambar 4.14. Nilai perubahan <i>eigen value</i> selama proses pengendalian secara rekursif.....	45



BAB I

PENDAHULUAN

1.1 Latar Belakang

Ruang pusat data (data center) adalah tempat untuk menyimpan peralatan IT yang digunakan sebagai peralatan telekomunikasi. Untuk menjaga kondisi peralatan IT tersebut agar dapat bekerja sebagaimana mestinya, faktor temperatur dan kelembaban udara di dalam ruangan perlu dijaga pada kondisi tertentu. Oleh karena itu, dibutuhkan sistem tata udara yang bisa mengatur temperatur dan kelembaban di dalam ruangan tersebut.

Pada kebanyakan ruang pusat data saat ini, sistem tata udara yang dipakai adalah sistem tata udara konvensional yang biasa. Hal ini disebabkan karena tidak adanya produk dalam negeri serta biaya yang cukup besar dibutuhkan untuk mendatangkan peralatan atau sistem tata udara yang didesain khusus untuk ruang pusat data dari luar negeri. Kondisi ini membuat penggunaan energi menjadi relatif lebih besar. Dengan kondisi-kondisi yang telah disebutkan di atas, maka dirancang suatu alat refrigasi yang dapat mengatur temperatur dan kelembaban kabinet yang dikenal dengan nama tata udara presisi atau *precision air conditioning (PAC)*.

Sistem tata udara presisi (*Precision Air Conditioning*) ini merupakan sistem MIMO (*Multi Input Multi Output*) yang mempunyai dua buah masukan dan dua buah keluaran. Masukan-masukan pada sistem ini adalah kecepatan kompresor dalam putaran tiap detik dan tegangan kipas dalam volt. Keluaran dari sistem tata udara presisi ini adalah temperatur udara dalam kabinet dan kelembaban relatif dalam kabinet. Sistem tata udara presisi ini juga merupakan sistem multivariabel, yaitu menggunakan 8 orde keadaan.

Sistem tata udara presisi akan digunakan untuk mengatur temperatur udara dalam kabinet dan kelembaban relatif udara dalam kabinet sehingga mencapai nilai yang diinginkan. Alat ini diharapkan dapat menjaga temperatur di kisaran 20-25⁰C dan kelembaban relatif kabinet sebesar 45-55 %. Penggunaan sistem tata

udara presisi ini dapat menghemat energi yang cukup signifikan sehingga biaya yang dikeluarkan dapat diminimalisir sekecil mungkin. Untuk mengatur kedua parameter tersebut, diperlukan perancangan pengendali cerdas. Sebelum merancang pengendali dari sistem, diperlukan model yang merepresentasikan sistem tersebut. Model yang dipilih adalah model ruang keadaan multivariabel. Model tersebut didapatkan dengan metode 4SID rekursif (Viktor, 2011).

Untuk merancang suatu pengendali pada suatu sistem, dalam hal ini pengendali dengan menggunakan Model Predictive Control (MPC), perlu diketahui terlebih dulu informasi variabel keadaan (*state*) dari sistem tersebut pada langkah pembaruan berikutnya. Namun pada realisasinya, sulit untuk mendapatkan variabel keadaan karena faktor gangguan serta keterbatasan peralatan elektronik yang dimiliki. Nilai keadaan tersebut dapat diperoleh secara rekursif dengan memanfaatkan algoritma identifikasi 4SID. Diharapkan melalui proses rekursif yang dilakukan untuk mencari variabel keadaan, diperoleh nilai keadaan yang cukup akurat dan tahan terhadap gangguan. Algoritma identifikasi yang digunakan adalah MOESP (*MIMO Output-Error State-Space Model Identification*) karena dalam penelitian sebelumnya (Viktor, 2011), metode MOESP menghasilkan model terbaik jika dibandingkan dengan algoritma yang lainnya, seperti PO-MOESP (*Ordinary MOESP scheme with Instrumental Variables Constructed from Past Input and Output Measurement*).

1.2 Rumusan Masalah

Rumusan masalah yang ada pada penelitian ini adalah :

1. Variabel keadaan (*state*) dari sistem tata udara presisi yang sulit diperoleh karena faktor keterbatasan peralatan elektronik serta alasan ekonomi.
2. Dibutuhkan algoritma estimasi untuk mengestimasi variabel keadaan pada waktu sekarang ($N|N$) dan pada langkah pembaruan berikutnya ($N+1|N$) yang diperlukan dalam perancangan pengendali.

1.3 Tujuan Penelitian

Penelitian ini bertujuan untuk mencari variabel keadaan dari sistem tata udara presisi yang bersifat multivariabel dan mempunyai beberapa masukan dan beberapa keluaran (MIMO). Metode yang digunakan untuk mencari variabel keadaan tersebut adalah dengan menggunakan algoritma identifikasi 4SID rekursif. Algoritma 4SID yang digunakan adalah MOESP (*MIMO Output-Error State-Space Model Identification*).

1.4 Pembatasan Masalah

Pada penelitian ini dilakukan dengan beberapa batasan masalah, antara lain :

1. Peneliti tidak melakukan penurunan rumus matematis untuk sistem tata udara presisi. Persamaan matematis diambil dari laporan penelitian yang lain (Viktor, 2011)
2. Laporan ini hanya membahas algoritma MOESP, beserta algoritma yang dirancang untuk mengestimasi variabel keadaan pada bab 2. Peneliti tidak melakukan analisis terhadap kinerja dari algoritma tersebut dalam mengidentifikasi model.
3. Peneliti tidak membahas algoritma pengendali yang digunakan untuk mengendalikan keluaran dari sistem tata udara presisi, yaitu *Model Predictive Control (MPC)*.

1.5 Metodologi Penelitian

Metodologi yang digunakan selama melakukan penelitian dan penulisan laporan adalah:

1. Studi Literatur
Penulis membaca buku, jurnal, disertasi yang berkaitan dengan penggunaan algoritma 4SID, baik untuk identifikasi serta untuk mengestimasi variabel keadaan secara rekursif.
2. Konsultasi dengan dosen pembimbing
Pertemuan yang rutin sekitar satu minggu sekali membuat penulis bisa melakukan penelitian dan penulisan berjalan dengan kontinu.

3. Diskusi dengan orang yang telah terlebih dahulu melakukan penelitian serupa Penulis bertanya kepada mereka yang telah melakukan penelitian serupa terlebih dahulu. Penulis melakukan surat-menyurat elektronik kepada beberapa penulis jurnal maupun disertasi.

1.6 Sistematika Penulisan

Laporan skripsi ini tersusun atas 5 bab, yaitu bab 1 yang berjudul pendahuluan. Bab ini berisi tentang latar belakang, tujuan penelitian, pembatasan masalah, metodologi penelitian, dan sistematika penulisan. Bab 2 merupakan bab yang berisi dasar teori. Pada bab ini, penulis menjelaskan metode 4SID, yaitu algoritma MOESP secara rekursif. Pada bagian ini, penulis juga menjelaskan algoritma yang digunakan dalam mengestimasi variabel keadaan. Pada bab 3, penulis menjelaskan langkah-langkah yang dilakukan dalam mengestimasi variabel keadaan. Pertama dijelaskan persamaan matematis *state* yang mewakili sistem tata udara presisi yang nilainya akan diestimasi pada penelitian ini. Penulis juga menjelaskan sinyal uji masukan yang dipakai dalam penelitian ini. Setelah itu, penulis menjelaskan implementasi algoritma ini jika dikombinasikan dengan algoritma pengendali.

Pada bab 4 penulis memaparkan hasil simulasi yang dilakukan untuk kondisi, yaitu estimasi variabel keadaan dengan menggunakan model linier PAC. Hasil estimasi *state* dengan variasi *noise* akan ditampilkan dalam ini. Selanjutnya akan dijelaskan analisis dari setiap percobaan yang dilakukan tersebut. Pada bab ini juga dijelaskan hasil estimasi *state* untuk model nonlinier PAC.

Pada bab 5, penulis menyimpulkan hasil percobaan dan analisis yang dilakukan serta menuliskan saran-saran yang berguna bagi pembaca yang ingin menggunakan, mempelajari, melanjutkan, ataupun mengembangkan percobaan yang telah dilaporkan pada laporan skripsi ini.

BAB 2

DASAR TEORI

2.1 Algoritma MOESP Rekursif

Algoritma MOESP Rekursif adalah sebuah metode memperbarui matriks data masukan keluaran (I/O) terkompresi secara rekursif menggunakan dekomposisi nilai tunggal (SVD). Pada setiap langkah pembaruan, akan didapatkan matriks koefisien dengan menggunakan dekomposisi nilai tunggal (SVD) dari matriks data I/O tersebut. Diasumsikan diketahui sistem LTI yang dideskripsikan dengan :

$$x_{k+1} = \mathbf{A}x_k + \mathbf{B}u_k \quad (2.1)$$

$$y_k = \mathbf{C}x_k + \mathbf{D}u_k \quad (2.2)$$

di mana, $x_k \in^n$, $u_k \in^r$, dan $y_k \in^m$. Matriks \mathbf{A} harus stabil (nilai *eigen value* harus berada di dalam unit *circle*) dan pasangan matriks (\mathbf{A} , \mathbf{C}) harus dapat diobservasi. Tujuan dari algoritma ini adalah mencari isi dari matriks koefisien (\mathbf{A} , \mathbf{B} , \mathbf{C} , \mathbf{D}) secara rekursif dengan mengolah data masukan keluaran (I/O) dari sistem yang nantinya digunakan pada algoritma untuk mengestimasi variabel keadaan.

Untuk urutan masukan dan keluaran yang terbatas (I/O) data $\{(u_j, y_j)\}$, $j = 1, \dots, N, u_v(i) \in^{r_v}, y_v(i) \in^{m_v}$ didefinisikan sebagai berikut:

$$u_v(i) := \begin{bmatrix} u_{i-v+1} \\ \vdots \\ u_i \end{bmatrix}, \quad y_v(i) := \begin{bmatrix} y_{i-v+1} \\ \vdots \\ y_i \end{bmatrix} \quad (2.3)$$

Lalu dari persamaan (2.1) dan (2.2) didapatkan:

$$y_v(i) = \mathbf{\Gamma}_v x_{i-v+1} + \mathbf{H}_v u_v(i) \quad (2.4)$$

di mana

$$\Gamma_v := \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{v-1} \end{bmatrix}, \quad \mathbf{H}_v := \begin{bmatrix} D & & 0 \\ CB & D & \\ \vdots & \vdots & \ddots \\ CA^{v-2} & CA^{v-3}B & \cdots & D \end{bmatrix} \quad (2.5)$$

Persamaan (2.4) menghasilkan persamaan matriks antara masukan, variabel keadaan dan keluaran seperti berikut:

$$Y_N = \Gamma_v X_{N-v+1} + \mathbf{H}_v U_N \quad (2.6)$$

dengan

$$U_N := [u_v(v) \quad u_v(v+1) \quad \cdots \quad u_v(N)] = \begin{bmatrix} u_1 & u_2 & \cdots & u_{N-v+1} \\ u_2 & u_3 & \cdots & u_{N-v+2} \\ \vdots & \vdots & & \vdots \\ u_v & u_{v+1} & \cdots & u_N \end{bmatrix} \quad (2.7)$$

$$Y_N := [y_v(v) \quad y_v(v+1) \quad \cdots \quad y_v(N)] = \begin{bmatrix} y_1 & y_2 & \cdots & y_{N-v+1} \\ y_2 & y_3 & \cdots & y_{N-v+2} \\ \vdots & \vdots & & \vdots \\ y_v & y_{v+1} & \cdots & y_N \end{bmatrix} \quad (2.8)$$

$$X_{N-v+1} := [x_1 \quad x_2 \quad \cdots \quad x_{N-v+1}]$$

U_N dan Y_N adalah matriks Hankel yang berisi N jumlah data masukan dan keluaran (I/O) yang berukuran berukuran $rv \times (N - v + 1)$ untuk masukan dan $mv \times (N - v + 1)$ untuk keluaran. $\Gamma_v \in^{mv \times n}$ disebut sebagai matriks disebut sebagai *extended observability matrix* dan $H_v \in^{mv \times rv}$ disebut matriks segitiga bawah *Toeplitz*. Jumlah baris Hankel (v) dapat ditentukan dengan syarat $v \geq n + 1$. n adalah banyaknya orde sistem. Banyaknya data yang digunakan untuk identifikasi secara *offline* harus memenuhi syarat $N \gg v$.

Didefinisikan dua matriks proyeksi:

$$\Pi_{U_N} := U_N^T (U_N U_N^T)^{-1} U_N \quad (2.9)$$

$$\Pi_{U_N}^\perp := I - \Pi_{U_N} \quad (2.10)$$

Perkalian antara persamaan (2.6) dengan Π_{U_N} menghasilkan

$$Y_N \Pi_{U_N}^\perp = \Gamma_v X_N \Pi_{U_N}^\perp \quad (2.11)$$

$Y_N \Pi_{U_N}^\perp$ pada persamaan (2.11) dapat dibentuk dari matriks Hankel masukan (2.7) dan keluaran (2.8). Jika dekomposisi nilai tunggal dari persamaan (2.11) adalah sebagai berikut

$$Y_N \Pi_{U_N}^\perp = \begin{bmatrix} E_N & E_N^\perp \end{bmatrix} \begin{bmatrix} S_n & \\ & 0 \end{bmatrix} \begin{bmatrix} F_N^\perp \\ (F_N^\perp)^T \end{bmatrix} \quad (2.12)$$

maka diketahui bahwa pada N yang cukup besar akan terdapat matriks T tidak singular $n \times n$, sehingga didapat

$$\Gamma_v = E_N T \quad (2.13)$$

Dari E_N dan E_N^\perp pada persamaan di atas, representasi ruang keadaan dari model (A_T, B_T, C_T, D_T) dapat dihitung sehingga diperoleh

$$\begin{aligned} A_T &= T A T^{-1}, & B_T &= T B \\ C_T &= C T^{-1}, & D_T &= D \end{aligned} \quad (2.14)$$

Prosedur yang lebih rinci untuk penurunan matriks-matriks di atas dideskripsikan di dalam Lampiran A.

Berikut ini akan dijelaskan algoritma MOESP rekursif untuk sistem bebas gangguan. Dari persamaan (2.12), perkalian dari $Y_N \Pi_{U_N}^\perp$ dapat diubah menjadi

$$\begin{aligned} \Xi_N &:= Y_N \Pi_{U_N}^\perp (Y_N \Pi_{U_N}^\perp)^T \\ &= Y_N \Pi_{U_N}^\perp Y_N^\perp \\ &= \begin{bmatrix} E_N & E_N^\perp \end{bmatrix} \begin{bmatrix} S_n^2 & \\ & 0 \end{bmatrix} \begin{bmatrix} E_N^T \\ (E_N^\perp)^T \end{bmatrix} \end{aligned} \quad (2.15)$$

Untuk melakukan identifikasi secara rekursif, langkah awal adalah menghitung matriks kovarian P_N

$$P_N := (U_N U_N^T)^{-1} \quad (2.16)$$

Diasumsikan bahwa Ξ_{N-1} , P_{N-1} , dan $Y_{N-1} U_{N-1}^T$ sudah didapatkan. Saat pasangan data I/O yang ke-N (u_N, y_N) diperoleh, $u_v(N)$ dan $y_v(N)$ dapat dibentuk dari persamaan (2.3). Lalu, matriks data I/O terkompresi yang ke-N (Ξ_N) dapat diperbarui secara rekursif dengan prosedur seperti berikut ini:

$$\Xi_N = \Xi_{N-1} + \alpha_N e_N e_N^T \quad (2.17)$$

di mana

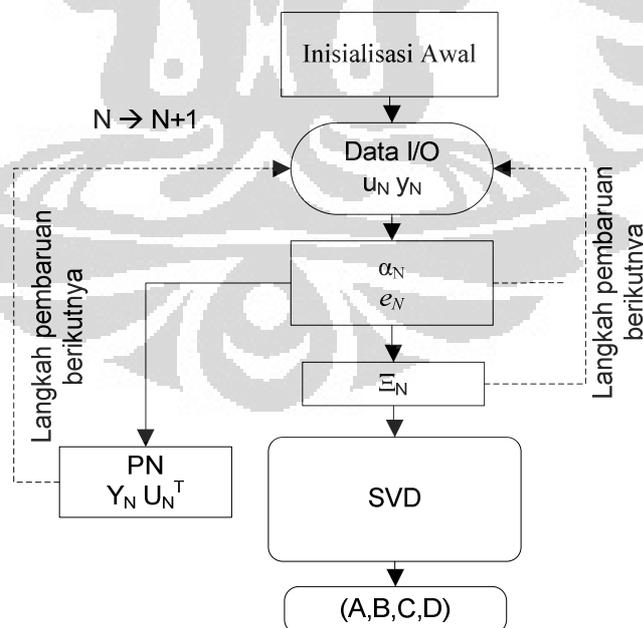
$$P_N = P_{N-1} - \alpha_N P_{N-1} u_v(N) u_v(N)^T P_{N-1} \quad (2.18)$$

$$\alpha_N := (1 + u_v(N)^T P_{N-1} u_v(N))^{-1} \quad (2.19)$$

$$e_N := y_v(N) - Y_{N-1} U_{N-1}^T P_{N-1} u_v(N) \quad (2.20)$$

$$Y_N U_N^T = Y_{N-1} U_{N-1}^T + y_v(N) u_v(N)^T \quad (2.21)$$

Berikut ini adalah skema diagram alir dari algoritma MOESP Rekursif



Gambar 2.1. Algoritma MOESP Rekursif

Untuk mendapatkan matriks koefisien ($\mathbf{A}_T, \mathbf{B}_T, \mathbf{C}_T, \mathbf{D}_T$) pada setiap langkah pembaruan, digunakan dekomposisi nilai tunggal dari matriks data I/O terkompresi ($\tilde{\Xi}_N$).

2.2 MOESP rekursif dengan Gangguan Terukur

Pada subbab ini, algoritma MOESP rekursif dengan kasus di mana terdapat nilai keluaran sistem yang terkontaminasi oleh gangguan (noise) yang dilambangkan dengan \tilde{y}_k . Diasumsikan $v_v(N)$ dan V_N adalah vektor dan matriks, yang terdiri dari gangguan \tilde{v}_k .

$$V_N := [v_v(v) \quad v_v(v+1) \quad \cdots \quad v_v(N)] = \begin{bmatrix} v_1 & v_2 & \cdots & v_{N-v+1} \\ v_2 & v_3 & \cdots & v_{N-v+2} \\ \vdots & \vdots & & \vdots \\ v_v & v_{v+1} & \cdots & v_N \end{bmatrix} \quad (2.22)$$

Definisikan

$$\tilde{y}_v(N) := y_v(N) + v_v(N) \quad (2.23)$$

$$\tilde{Y}_N := Y_N + V_N$$

sehingga

$$\tilde{Y}_N := [\tilde{y}_v(v) \quad \tilde{y}_v(v+1) \quad \cdots \quad \tilde{y}_v(N)] = \begin{bmatrix} \tilde{y}_1 & \tilde{y}_2 & \cdots & \tilde{y}_{N-v+1} \\ \tilde{y}_2 & \tilde{y}_3 & \cdots & \tilde{y}_{N-v+2} \\ \vdots & \vdots & & \vdots \\ \tilde{y}_v & \tilde{y}_{v+1} & \cdots & \tilde{y}_N \end{bmatrix} \quad (2.24)$$

Diasumsikan bahwa $\tilde{\Xi}_{N-1}, P_{N-1}$, dan $\tilde{Y}_{N-1} U_{N-1}^T$ sudah didapatkan. Saat pasangan data I/O yang ke-N (u_N, \tilde{y}_N) diperoleh, $u_v(N)$ dan $y_v(N)$ dapat dibentuk dari persamaan (2.3). Lalu, matriks data I/O terkompresi yang ke-N ($\tilde{\Xi}_N$) dapat diperbarui secara rekursif dengan prosedur seperti berikut ini:

$$\tilde{\Xi}_N = \tilde{\Xi}_{N-1} + \alpha_N \tilde{e}_N \tilde{e}_N^T \quad (2.25)$$

di mana

$$P_N = P_{N-1} - \alpha_N P_{N-1} u_v(N) u_v(N)^T P_{N-1} \quad (2.26)$$

$$\alpha_N := \left(1 + u_v(N)^T P_{N-1} u_v(N)\right)^{-1} \quad (2.27)$$

$$\tilde{e}_N := \tilde{y}_v(N) - \tilde{Y}_{N-1} U_{N-1}^T P_{N-1} u_v(N) \quad (2.28)$$

$$\tilde{Y}_N U_N^T = \tilde{Y}_{N-1} U_{N-1}^T + \tilde{y}_v(N) u_v(N)^T \quad (2.29)$$

2.3 Algoritma untuk Mengestimasi Variabel Keadaan (*state*)

Pada subbab ini, algoritma estimasi variabel keadaan (*state*) dari model multivariabel dirumuskan. Skema untuk mengestimasi variabel keadaan dari model dapat diperoleh dari algoritma rekursif 4SID, yang dalam penelitian ini menggunakan MOESP. Sistem diskrit LTI yang akan diestimasi variabel keadaannya dideskripsikan sebagai berikut:

$$x_{k+1} = \mathbf{A}x_k + \mathbf{B}u_k \quad (2.30)$$

$$y_k = \mathbf{C}x_k + \mathbf{D}u_k \quad (2.31)$$

Variabel keadaan awal dari sistem x_0 dianggap bernilai 0. Diasumsikan bahwa terdapat bilangan bulat $N_0 > 0$ sehingga untuk bilangan bulat v yang diberikan memenuhi $v \geq n+1$, urutan data masukan $\{u_i\}$ dan $\forall N \geq N_0$:

$$\sum_{j=1}^N u_v(j) u_v(j)^T \quad (2.32)$$

adalah nonsingular, dimana notasi $u_v(i)$ didefinisikan sebagai berikut

$$u_v(i) := \begin{bmatrix} u_{i-v+1} \\ \vdots \\ u_i \end{bmatrix} \quad (2.33)$$

Jika pengukuran keluaran dari sistem terkontaminasi oleh gangguan v_k , maka pengukuran dapat dideskripsikan sebagai

$$\tilde{y}_k = y_k + v_k \quad (2.34)$$

di mana gangguan v_k adalah *zero-mean Gaussian white noise* dengan kovarian

$$E v_i v_i^T = \sigma_v^2 I \quad (2.35)$$

Diasumsikan bahwa v_k tidak tergantung kepada u_i untuk setiap i dan variabel keadaan awal x_0 .

2.4 Proses Estimasi Variabel Keadaan (*state*) pada Model MIMO

Keuntungan dari skema 4SID adalah orde dari model, yang dilambangkan dengan \hat{n} , dapat ditentukan sebagai *number of dominant singular values* pada persamaan (2.12). Dengan begitu, ketika kita menggunakan algoritma 4SID secara rekursif untuk suatu sistem, kita mungkin harus menentukan lagi orde dari model pada saat langkah pembaruan yang berbeda.

Jika kita dapat menentukan orde yang sesuai dengan sistem, dari persamaan (2.14) sistem dan model yang diidentifikasi memiliki keterkaitan dengan nilai matriks transformasi yang sama. Pada saat proses identifikasi secara rekursif, persamaan matriks transformasi T_1 antara model yang diidentifikasi dengan sistem pada saat langkah ke N_1 mungkin tidak akan sama dengan T_2 pada saat langkah N_2 , atau dengan kata lain $T_1 \neq T_2$ untuk $N_1 \neq N_2$.

Pada bagian ini, akan dibahas bagaimana cara mengestimasi variabel keadaan dengan menggunakan algoritma 4SID secara rekursif dari suatu sistem multivariabel yang tidak diketahui. Masalah ini menjadi sangat penting karena pengetahuan variabel keadaan waktu sekarang dari model yang diidentifikasi digunakan untuk mendesain kontrol dari sistem tersebut. Oleh karena itu metode estimasi variabel keadaan secara konvensional, seperti filter kalman tidak dapat dipakai secara langsung karena untuk kasus ini, pada filter kalman diasumsikan bahwa orde dari sistem fix dan pada semua langkah pembaruan terjadi kesamaan matriks transformasi.

Untuk mengatasi masalah ini, akan diperkenalkan jenis *state* baru yang berpengaruh pada setiap langkah pembaruan yang berbeda. Pada persamaan

$$z_i := \Gamma_v x_i \quad z_i \in^{mv} \quad (2.36)$$

Kita dapat memperoleh z_i . Simbol z_i disebut sebagai "*the quasi-state*" pada langkah ke- i pada penelitian ini. Perlu dicatat bahwa dimensi dari z_i tidak

tergantung dari orde dari sistem n dan orde dari model \hat{n} . Dari asumsi bahwa Γ_v *rank* penuh kolom, variabel keadaan dapat dilambangkan sebagai

$$x_i = \Gamma_v^x z_i \quad (2.37)$$

dimana Γ_v^x adalah invers *Moore-Penrose* dari Γ_v

Jika hal tersebut tidak untuk gangguan pengukuran dan jika blok segitiga bawah dari matrik *Toeplitz* mengandung parameter *Markov* H_v diketahui, maka dengan mudah dapat diperoleh *quasi-state* z_i sesuai dengan persamaan (2.4),

$$z_i = y_{v(i+v-1)} - H_v u_{v(i+v-1)} \quad (2.38)$$

karena matrik H_v tidak diketahui tetapi dapat diestimasi dengan menggunakan identifikasi rekursif, maka *quasi-state* z_i juga dapat diestimasi secara rekursif.

Ketika pasangan data (u_N, \tilde{y}_N) diperoleh pada langkah ke- N , $u_v(N)$ dan $\tilde{y}_v(N)$ dapat diperoleh dengan menggunakan persamaan (2.3). Hasil estimasi Γ_v , H_v , dan (A, B, C, D) pada langkah ke- N masing-masing dilambangkan dengan

$$E_N, \hat{H}_N, (\hat{A}_N, \hat{B}_N, \hat{C}_N, \hat{D}_N) \quad (2.39)$$

Sekarang, berdasarkan persamaan (2.4), kita anggap bahwa hasil estimasi $\hat{z}_{N-v+1|N}$ dari *quasi-state* $z_{N-v+1|N}$ diperoleh melalui persamaan :

$$\hat{z}_{N-v+1|N} := \tilde{y}_v(N) - \hat{H}_N u_v(N) - E_N^\perp (E_N^\perp)^T \alpha_N \tilde{e}_N \quad (2.40)$$

Pada *term* ketiga pada ruas kanan persamaan (2.40) memiliki peran untuk mengurangi pengaruh gangguan. Pembuktiannya akan dijelaskan pada pembahasan berikutnya. Dari persamaan (2.1), hasil estimasi $\hat{z}_{N|N}$ dan $\hat{z}_{N+1|N}$ dari *quasi-state* $z_{N|N}$ dan $z_{N+1|N}$ dapat dihitung masing-masing dengan persamaan :

$$\hat{z}_{N|N} := E_N (\hat{A}_N)^{v-1} E_N^x \hat{z}_{N-v+1|N} + E_N \sum_{i=1}^{v-1} (\hat{A}_N)^{i-1} \hat{B}_N u_{N-i} \quad (2.41)$$

$$\hat{z}_{N+1|N} := E_N \hat{A}_N E_N^x \hat{z}_{N|N} + E_N \hat{B}_N u_N \quad (2.42)$$

Dari persamaan (2.37), variabel keadaan dari model yang diestimasi $(\hat{A}_N, \hat{B}_N, \hat{C}_N, \hat{D}_N)$ dapat dihitung dengan persamaan

$$\hat{x}_{N-v+1|N} := E_N^z \hat{z}_{N-v+1|N} \quad (2.43)$$

$$\hat{x}_{N|N} := E_N^z \hat{z}_{N|N} \quad (2.44)$$

$$\hat{x}_{N+1|N} := E_N^z \hat{z}_{N+1|N} \quad (2.45)$$

Pada bagian ini, akan dijelaskan bagaimana *term* ketiga ruas kanan pada persamaan (2.40) dapat menghilangkan pengaruh gangguan. Dari persamaan (2.34) dan (2.6), dapat diperoleh

$$\tilde{Y}_N = \Gamma_v X_{N-v+1} + H_v U_N + V_N \quad (2.46)$$

Diasumsikan terdapat matrik nonsingular T sehingga $\Gamma = E_N T$. *Premultiplication* pada persamaan (2.46) dengan $(E_N^\perp)^T$ dan *postmultiplication* dengan U_N^T menghasilkan

$$(E_N^\perp)^T \tilde{Y}_N U_N^T = (E_N^\perp)^T H_v U_N U_N^T + (E_N^\perp)^T V_N U_N^T \quad (2.47)$$

Kita dapat mengabaikan *term* kedua dari ruas kanan persamaan (2.47) untuk nilai N yang cukup besar,

$$(E_N^\perp)^T \tilde{Y}_N U_N^T = (E_N^\perp)^T H_v U_N U_N^T \quad (2.48)$$

Dari persamaan (2.16)

$$(E_N^\perp)^T (\tilde{y}_v(N) - H_v u_v(N)) = (E_N^\perp)^T (\tilde{y}_v(N) - \tilde{Y}_N U_N^T P_N u_v(N)) \quad (2.49)$$

perlu dicatat bahwa dari persamaan (2.19)

$$\begin{aligned} \tilde{y}_v(N) - \tilde{Y}_N U_N^T P_N u_v(N) &= \tilde{y}_v(N) - \begin{bmatrix} \tilde{Y}_{N-1} & \tilde{y}_v(N) \end{bmatrix} \begin{bmatrix} U_{N-1}^T \\ u_v(N)^T \end{bmatrix} \\ &= \tilde{y}_v(N) - \tilde{Y}_{N-1} U_{N-1}^T P_{N-1} u_v(N) - \tilde{y}_v(N) u_v(N)^T P_{N-1} u_v(N) \\ &\quad + \alpha_N \tilde{Y}_{N-1} U_{N-1}^T P_{N-1} u_v(N) u_v(N)^T P_{N-1} u_v(N) \\ &\quad + \alpha_N \tilde{y}_v(N) u_v(N)^T P_{N-1} u_v(N) u_v(N)^T P_{N-1} u_v(N) \\ &= \tilde{y}_v(N) - \tilde{y}_v(N) u_v(N)^T P_{N-1} u_v(N) (1 - \alpha_N u_v(N)^T P_{N-1} u_v(N)) \end{aligned}$$

$$\begin{aligned}
& -\tilde{Y}_{N-1}U_{N-1}^T P_{N-1}u_v(N)(1-\alpha_N u_v(N)^T P_{N-1}u_v(N)) \\
& = \tilde{y}_v(N) - \alpha_N \tilde{y}_v(N)u_v(N)^T P_{N-1}u_v(N) \\
& \quad - \alpha_N \tilde{Y}_{N-1}U_{N-1}^T P_{N-1}u_v(N) \\
& = \tilde{y}_v(N)(1-\alpha_N u_v(N)^T P_{N-1}u_v(N)) - \alpha_N \tilde{Y}_{N-1}U_{N-1}^T P_{N-1}u_v(N) \\
& = \alpha_N (\tilde{y}_v(N) - \tilde{Y}_{N-1}U_{N-1}^T P_{N-1}u_v(N)) \\
& = \alpha_N \tilde{e}_N
\end{aligned} \tag{2.50}$$

Oleh sebab itu dapat diperoleh

$$(E_N^\perp)^T (\tilde{y}_v(N) - H_v u_v(N)) = (E_N^\perp)^T \alpha_N \tilde{e}_N \tag{2.51}$$

Dengan cara lain, tanpa ada gangguan, kita dapat menurunkan

$$\begin{aligned}
(E_N^\perp)^T (\tilde{y}_v(N) - H_v u_v(N)) & = (E_N^\perp)^T (y_v(N) - Y_{N-1}U_{N-1}^T P_{N-1}u_v(N)) \\
& = (E_N^\perp)^T \alpha_N (y_v(N) - Y_{N-1}U_{N-1}^T P_{N-1}u_v(N)) \\
& = (E_N^\perp)^T \alpha_N e_N
\end{aligned} \tag{2.52}$$

Karena $\Gamma_v = E_n T$ kita dapat memperoleh

$$(E_N^\perp)^T \alpha_N e_N = (E_N^\perp)^T \alpha_N (\Gamma_v x_{N-v+1} - \Gamma_v X_{N-v} U_{N-1}^T P_{N-1} u_v(N)) = 0 \tag{2.53}$$

Oleh sebab itu, dalam membandingkan persamaan (2.51) dengan persamaan (2.53), kita dapat menganggap bentuk $E_n^\perp (E_n^\perp)^T \alpha_N \tilde{e}_N$ sebagai kesalahan estimasi dari z_{N-v+1} karena gangguan pengukuran.

Berikut adalah kesimpulan skema estimasi variabel keadaan.

1. Model dari sistem pada langkah ke- N yang dihasilkan dari algoritma 4SID secara rekursif.

$$E_N, \hat{H}_N, (\hat{A}_N, \hat{B}_N, \hat{C}_N, \hat{D}_N)$$

2. Variabel keadaan waktu lampau dari model $\hat{z}_{N-v+1|N}$ dan $\hat{x}_{N-v+1|N}$ yang diperoleh dari persamaan :

$$\hat{z}_{N-v+1|N} = \tilde{y}_v(N) - \hat{H}_N u_v(N) - E_N^\perp (E_N^\perp)^T \alpha_N \tilde{e}_N$$

$$\hat{x}_{N-v+1|N} := E_N^\perp \hat{z}_{N-v+1|N}$$

3. Variabel keadaan waktu sekarang dari model $\hat{z}_{N|N}$ dan $\hat{x}_{N|N}$ yang diperoleh dari persamaan :

$$\hat{z}_{N|N} := E_N \left(\hat{A}_N \right)^{v-1} E_N^z \hat{z}_{N-v+1|N} + E_N \sum_{i=1}^{v-1} \left(\hat{A}_N \right)^{i-1} \hat{B}_N u_{N-i}$$

$$\hat{x}_{N|N} := E_N^z \hat{z}_{N|N}$$

4. Variabel keadaan untuk satu langkah ke depan dari model $\hat{z}_{N+1|N}$ dan $\hat{x}_{N+1|N}$ yang diperoleh dari persamaan :

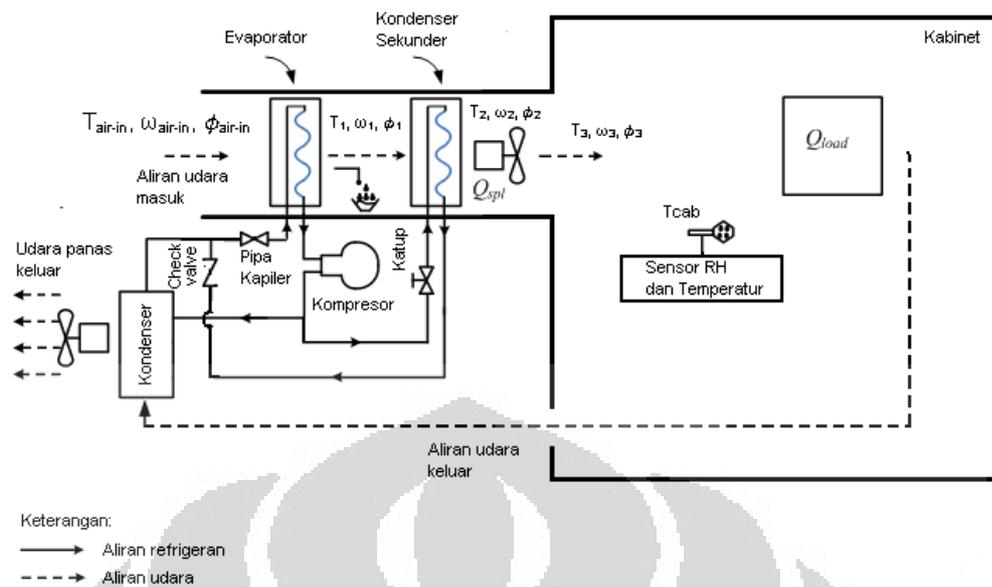
$$\hat{z}_{N+1|N} := E_N \hat{A}_N E_N^z \hat{z}_{N|N} + E_N \hat{B}_N u_N$$

$$\hat{x}_{N+1|N} := E_N^z \hat{z}_{N+1|N}$$

2.5 Sistem Tata Udara Presisi

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

PAC banyak digunakan di berbagai kebutuhan industri maupun rumah tangga. Penggunaan PAC ini dapat mengefisiensikan penggunaan energi. PAC ini digunakan untuk mengendalikan suhu dan kelembaban udara relatif pada kabinet yang ada di ruang pusat data, sehingga suhu dan kelembabannya terjaga konstan di nilai tertentu. Hal ini bertujuan untuk menjaga peralatan IT bisa beroperasi secara kontinu dengan meminimalkan kemungkinan kerusakan. Tujuan lainnya adalah untuk menjaga usia pemakaian peralatan IT tersebut agar bertahan lama. Tujuan lain yang seperti telah disebutkan di atas adalah pengefisienan energi, sehingga pengeluaran perusahaan bisa ditekan menjadi lebih murah. Suhu ideal untuk peralatan IT sekitar 20-22°C dan kelembaban relatif ideal untuk peralatan IT sekitar 45-55%.



Gambar 2.2 Bagan Sistem Tata Udara Presisi

Skripsi ini merupakan modifikasi dari skripsi maupun jurnal tentang PAC yang telah dilakukan sebelumnya oleh Viktor (Viktor, 2011), Rise Hapshary Surayuda (Surayuda, 2010), Sutarna (Sutarna, 2008), dan Deng dan (Deng & Qi, 2009). Prinsip kerja dari sistem tata udara presisi ini dapat dilihat pada laporan penulisan skripsi sebelumnya, yaitu pada Viktor (Viktor, 2011).

BAB 3

ESTIMASI VARIABEL KEADAAN SISTEM TATA UDARA PRESISI

3.1 Persamaan Matematis Variabel Keadaan Sistem Tata Udara Presisi

Dalam membahas persamaan matematis dan selama proses estimasi, ada beberapa asumsi yang perlu diperhatikan, antara lain:

- Campuran udara terjadi di dalam evaporator, kondenser dan lingkungan
- Suhu evaporasi di evaporator dianggap konstan
- Sisi udara di evaporator meliputi daerah kering (*dry region*) dan daerah basah (*wet region*)
- Perbandingan volume udara *dry region* terhadap volume udara sisi *wet region* adalah 1:4
- Suhu kondensasi di kondenser dianggap konstan
- Sisi udara di kondenser hanya meliputi daerah kering saja
- Aliran refrigerant yang mengalir ke kondenser kedua diasumsikan sebanyak 10% dari *mass flow refrigerant total*, yaitu aliran refrigerant yang keluar dari kompresor atau aliran refrigerant yang masuk ke evaporator ataupun keluar dari evaporator
- Tekanan di kompresor dianggap konstan
- Beban dianggap konstan
- Kecepatan aliran udara volumetric (*air volumetric flow*) dalam sistem dianggap konstan
- Rugi-rugi panas pada daerah aliran udara diabaikan

Sistem tata udara presisi ini, keluaran yang dikendalikan adalah suhu kabinet (T_{cab}) dan kelembaban relatif kabinet (ω_{cab}). Persamaan matematis yang dipakai adalah model kompresor, model evaporator, model kondenser kedua, model udara masuk kabinaet (*supply air*) dan model kabinet. Sedangkan model kondenser pertama diabaikan karena kondenser pertama tidak berpengaruh

terhadap suhu dan kelembaban relatif di kabinet. T_{cab} dan ω_{cab} didapat tanpa memerlukan informasi mengenai udara yang dibuang keluar oleh kondenser pertama. Dari kelima model yang dipakai, terdapat 8 variabel keadaan yang nilainya akan diestimasi dalam penelitian ini, secara berurutan antara lain :

1. T_{cab} , suhu udara kabinet ($^{\circ}\text{C}$).
2. ω_{cab} , kelembaban spesifik udara kabinet (kg/kg).
3. T_1 , suhu udara keluaran evaporator ($^{\circ}\text{C}$).
4. T_1' , suhu udara di antara *dry region* dan *wet region* evaporator ($^{\circ}\text{C}$).
5. T_2 , suhu udara keluaran kondenser kedua ($^{\circ}\text{C}$).
6. T_{we} , suhu dinding evaporator ($^{\circ}\text{C}$).
7. T_{wc2} , suhu dinding kondenser kedua ($^{\circ}\text{C}$).
8. ω_1 , kelembaban spesifik udara keluaran evaporator (kg/kg).

- **Model Kompresor**

Persamaan matematis dari model kompresor yang digunakan untuk sistem tata udara presisi ini adalah sebagai berikut:

$$M_{ref} = \frac{sV_{com}}{v_s} \left[1 + 0,015 \left[\left(\frac{P_c}{P_e} \right)^{\frac{1}{\beta}} - 1 \right] \right] \quad (3.1)$$

di mana

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

s : kecepatan kompresor (rps)

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

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

P_c : tekanan kondensasi (kPa)

P_e : tekanan evaporasi (kPa)

β : indeks kompresi

Swept volume pada persamaan (3.1) dicari dengan:

$$V_{com} = \frac{V_d}{n_c} \quad (3.2)$$

di mana

- V_d : displacement volume compressor (m^3)
 n_c : jumlah silinder pada kompresor

- **Model Kondenser Kedua**



Gambar 3.1 Skema Aliran Udara di Kondenser Kedua

Gambar 3.1 menunjukkan skema diagram aliran udara di kondenser kedua, yang berbeda dengan skema aliran udara evaporator karena kondenser hanya memiliki daerah *dry region*.

Persamaan matematis untuk kondenser kedua pada sistem tata udara presisi adalah:

$$C_{pu}\rho_u V_{wc2} \frac{dT_2}{dt} = C_{pu}\rho_u f(T_1 - T_2) + UA_3 \left(T_{wc2} - \frac{T_1 + T_2}{2} \right) \quad (3.3)$$

Persamaan matematis dinding kondenser kedua pada sistem tata udara presisi adalah:

$$C_{pw}\rho_w V_{wc2} \frac{dT_{wc2}}{dt} = UA_3 \left(\frac{T_1 + T_2}{2} - T_{wc2} \right) - M_{ref2} (h_{oc2} - h_{ic2}) \quad (3.4)$$

di mana

- V_{wc2} : volume sisi udara kondenser kedua (m^3)
 UA_3 : perpindahan kalor keseluruhan di kondenser kedua ($kW/^\circ C$)
 T_2 : suhu udara keluaran kondenser kedua ($^\circ C$)
 T_{wc2} : suhu dinding kondenser kedua ($^\circ C$)
 M_{ref2} : aliran massa refrigerant di kondenser kedua (kg/s)

$$(M_{ref2} = 10\% M_{ref})$$

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

h_{oe2} : entalpi di output kondenser kedua (kJ/kg)

Pada Gambar 2.2 terlihat bahwa udara yang keluar dari kondenser kedua melewati *fan* sebelum masuk ke kabinet. *Fan* memiliki panas yang dapat menyebabkan suhu udara naik sedikit walaupun tidak signifikan dan dapat diabaikan. Akan tetapi, karena diasumsikan tidak terdapat beban kelembaban yang dihasilkan oleh *fan*, maka kelembaban spesifik udara setelah melewati *fan* (ω_3) dianggap sama dengan kelembaban spesifik keluaran kondenser sekunder (ω_2), sehingga persamaannya menjadi:

$$\omega_3 = \omega_2 = \omega_1 \quad (3.5)$$

Persamaan matematis model udara masuk (*supply air*)

$$T_3 = \frac{C_{pu} \rho_u f T_2 + Q_{spl}}{C_{pu} \rho_u f} \quad (3.6)$$

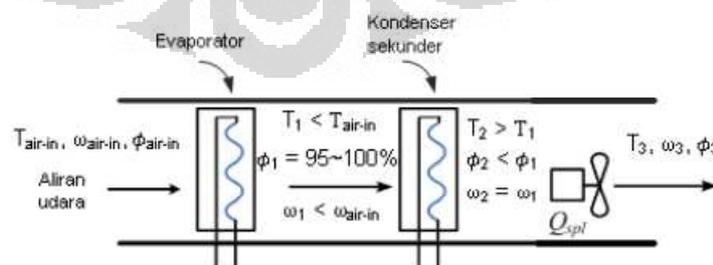
di mana

T_3 : Suhu udara setelah melewati *fan* (°C)

Q_{spl} : *heat* dari *fan* (kW)

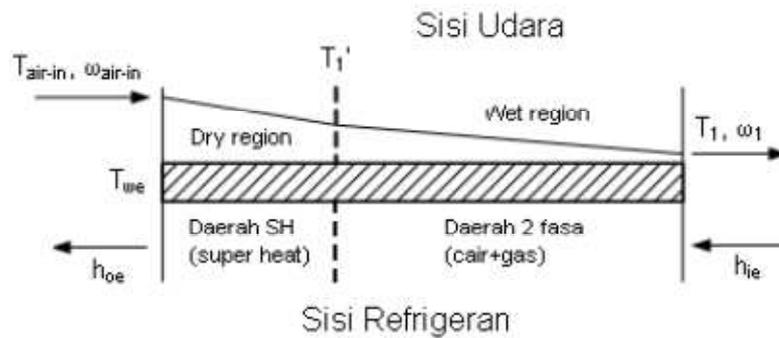
• Model Evaporator

Gambar 3.2 menunjukkan skema aliran udara yang melewati evaporator dan kondenser kedua beserta keterangan perubahan suhu dan kelembaban selama melewati kedua komponen tersebut



Gambar 3.2 Skema Aliran Udara di Evaporator dan Kondenser Kedua

Evaporator sendiri memiliki skema aliran udara yang terbagi menjadi *dry region* dan *wet region*.



Gambar 3.3 Skema Aliran Udara dan Refrigeran di Evaporator

Persamaan matematis di *dry region* pada evaporator adalah:

$$C_{pu}\rho_u V_1 \frac{dT_1'}{dt} = C_{pu}\rho_u f(T_{air-in} - T_1') + UA_1 \left(T_{we} - \frac{T_{air-in} + T_1'}{2} \right) \quad (3.7)$$

Persamaan matematis di *wet region* pada evaporator adalah:

$$C_{pu}\rho_u V_2 \frac{dT_1}{dt} + \rho_u V_2 h_{fg} \frac{d\omega_1}{dt} = C_{pu}\rho_u f(T_1' - T_1) + \rho_u f h_{fg} (\omega_{air-in} - \omega_1) + UA_2 \left(T_{we} - \frac{T_1' + T_1}{2} \right) \quad (3.8)$$

$$\omega_1 = \frac{(0,0198T_1'^2 + 0,085T_1' + 4,4984)}{1000} \quad (3.9)$$

$$\frac{d\omega_1}{dt} = \frac{(2 \times 0,0198T_1' + 0,085) \frac{dT_1'}{dt}}{1000} \quad (3.10)$$

Persamaan matematis di dinding evaporator sistem tata udara presisi yang digunakan adalah:

$$C_{pw}\rho_w V_{we} \frac{dT_{we}}{dt} = UA_1 \left(\frac{T_{air-in} + T_1'}{2} - T_{we} \right) + UA_2 \left(\frac{T_1' + T_1}{2} - T_{we} \right) - M_{refl} (h_{oe} - h_{ie}) \quad (3.11)$$

di mana

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

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

ρ_u : kerapatan udara (kg/m)

ρ_w : kerapatan dinding evaporator/kondenser (kg/m³)

- V_1 : volume sisi udara evaporator di *dry region* (m^3)
 V_2 : volume sisi udara evaporator di *wet region* (m^3)
 V_{we} : volume sisi udara evaporator total (m^3)
 f : kecepatan aliran udara volumetris (m^3/s)
 h_{fg} : kalor laten dari vaporasi udara (kJ/kg)
 UA_1 : perpindahan kalor keseluruhan di *dry region* evaporator (kW/°C)
 UA_2 : perpindahan kalor keseluruhan di *wet region* evaporator (kW/°C)
 ω_{air-in} : kelembaban spesifik udara di ruang pusat data (kg/kg)
 ω_1 : kelembaban spesifik udara keluaran evaporator (kg/kg)
 T_{air-in} : suhu udara di ruang pusat data (°C)
 T_1' : suhu udara di antara *dry region* dan *wet region* evaporator (°C)
 T_1 : suhu udara keluaran evaporator (°C)
 T_{we} : suhu dinding evaporator (°C)
 M_{refl} : aliran massa refrigeran di evaporator (kg/s) ($M_{refl} = M_{ref}$)
 h_{ie} : entalpi di input evaporator (kJ/kg)
 h_{oe} : entalpi di output evaporator (kJ/kg)

- **Model Kabinet**

Persamaan matematis suhu kabinet:

$$C_{pu} \rho_u V_{cab} \frac{dT_{cab}}{dt} = C_{pu} \rho_u f (T_3 - T_{cab}) + Q_{load} \quad (3.12)$$

Persamaan matematis kelembaban kabinet:

$$\rho_u V_{cab} \frac{d\omega_{cab}}{dt} = \rho_u f (\omega_3 - \omega_{cab}) + M \quad (3.13)$$

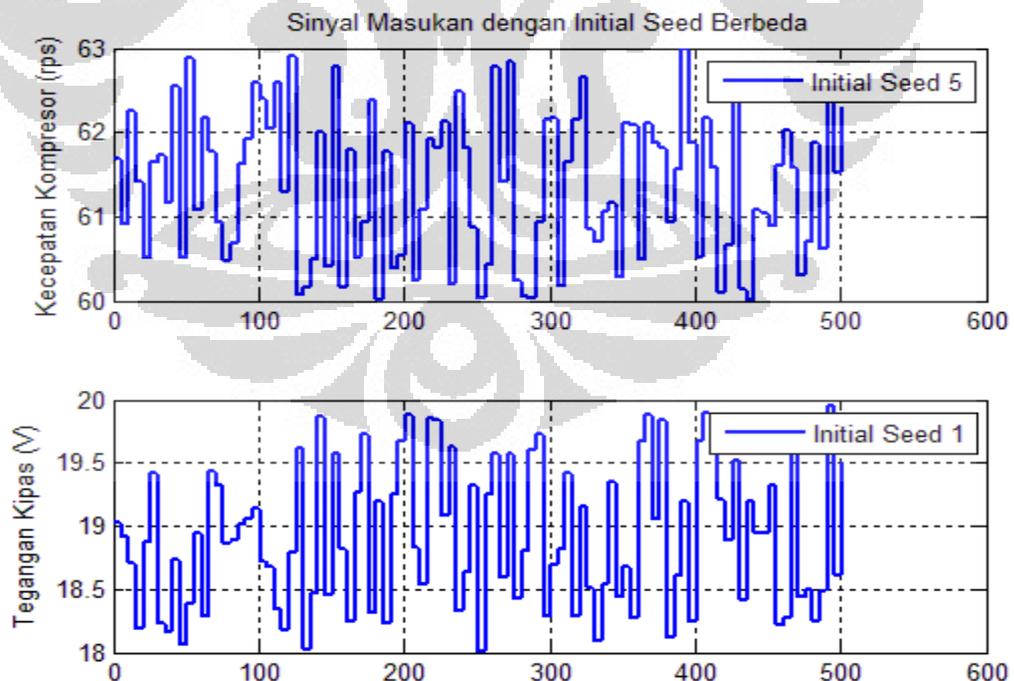
di mana

- V_{cab} : volume kabinet (m^3)
 T_{cab} : suhu udara kabinet (°C)
 ω_{cab} : kelembaban spesifik udara kabinet (kg/kg)
 Q_{load} : beban *heat sensible* dari peralatan
 M : beban kelembaban di kabinet (kg/s)

3.2 Penentuan pasangan data (u_N, y_N)

Untuk menguji algoritma pengestimasi variabel keadaan sistem tata udara presisi ini, diperlukan terlebih dahulu pasangan data (u_N, y_N) yang diperoleh dari sistem tata udara presisi. Masukan yang diberikan untuk kecepatan kompresor dan tegangan kipas adalah masukan jenis PRS (*Pseudo Random Signal*). Ciri dari masukan PRS adalah sinyal acak yang semu, di mana sinyal tersebut akan mempunyai nilai yang sama pada setiap *initial seed* yang sama. Initial seed merupakan semacam golongan sinyal, di mana untuk nilai *initial seed* yang sama, maka sinyal yang dikeluarkan mempunyai bentuk dan nilai yang sama.

Sinyal ini akan dimasukkan ke dalam sistem dan akan direkam hasil keluarannya. Proses identifikasi dan estimasi variabel keadaan yang dilakukan pada percobaan ini menggunakan dua buah masukan yang menggunakan *initial seed* yang berbeda. Jika menggunakan dua buah masukan yang mempunyai *initial seed* yang sama, maka matriks P_N pada persamaan (2.14) akan menghasilkan nilai yang salah, karena $(U_N U_N^T)$ tidak *invertible*.



Gambar 3.5 Sinyal Masukan dengan *Initial Seed* yang Berbeda

Untuk kecepatan kompresor, sinyal yang divariasikan berkisar antara 60 hingga 63 rps. Sedangkan untuk tegangan kipas, sinyal yang divariasikan berkisar antara 18 hingga 20. Dapat dilihat dari gambar 3.5 bahwa bentuk sinyal kecepatan kompresor (bagian atas) dan sinyal tegangan kipas (bagian bawah) adalah berbeda. Tidak ada korelasi antara sinyal yang satu dengan sinyal yang lainnya. Kedua sinyal ini akan dijadikan sinyal masukan sistem tata udara presisi. Dari sinyal uji masukan yang ditunjukkan pada gambar 3.5.

3.3 Pengujian Algoritma untuk Mengestimasi Variabel Keadaan

Untuk menguji algoritma yang dipakai, data masukan dan keluaran dari sistem PAC dikumpulkan terlebih dahulu. Untuk mengambil data masukan dan keluaran, digunakan blok *simulink* seperti pada gambar 3.7. Dengan menggunakan data masukan acak dengan *initial seed* yang berbeda untuk kecepatan kompresor dan tegangan kipas, diperoleh data keluaran untuk suhu kabinet ($^{\circ}\text{C}$) dan kelembaban relatif kabinet (%). Data masukan dan keluaran ini kemudian direkam pada *file masukan_keluaran_linier* yang bertipe *.mat*. Pasangan data I/O ini merupakan pasangan data untuk model linier sistem tata udara presisi yang digunakan untuk menguji algoritma untuk mengestimasi variabel keadaan. Proses berikutnya adalah proses inisialisasi awal. Untuk mendapatkan inisialisasi awal dari $E_N, \hat{H}_N, (\hat{A}_N, \hat{B}_N, \hat{C}_N, \hat{D}_N)$ digunakan algoritma MOESP secara *offline*.

Dalam menguji algoritma untuk mengestimasi variabel keadaan, jumlah pasangan data (u_N, y_N) yang digunakan dalam algoritma MOESP rekursif sama dengan jumlah keseluruhan pasangan data dikurangi dengan jumlah data *offline* (N). Ilustrasinya adalah jika terdapat data sebanyak 1000 data, dengan data *offline* (N) sebanyak 300 data mulai dari data ke-1 hingga ke- N , maka data yang dipakai untuk proses identifikasi secara rekursif serta untuk validasi adalah data ke- $N+1$ hingga data ke-1000.

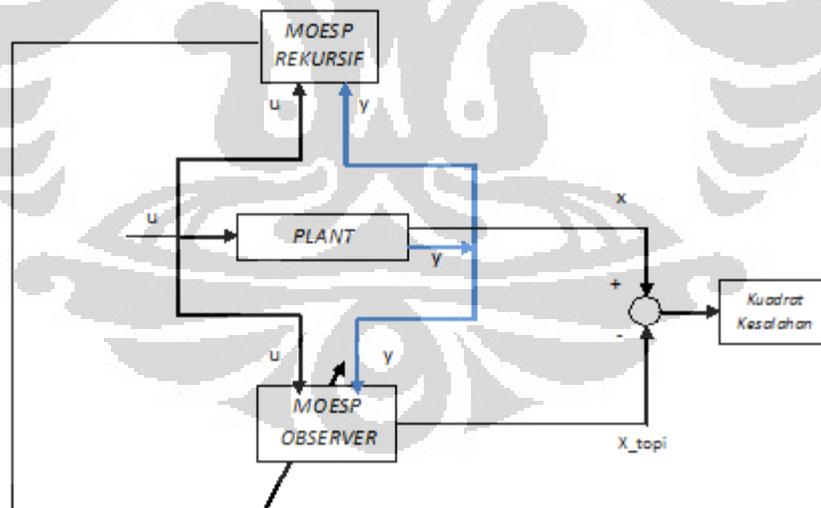
Proses validasi dilakukan dengan menggunakan persamaan :

$$x_{k+1} = \mathbf{A}x_k + \mathbf{B}u_k \quad (3.16)$$

$$y_k = \mathbf{C}x_k + \mathbf{D}u_k \quad (3.17)$$

Nilai A, B, C, dan D pada persamaan (3.16) dan (3.17) diperoleh melalui proses identifikasi secara *offline* menggunakan algoritma MOESP, sedangkan variabel keadaan waktu sekarang $\hat{x}_{N|N}$ dan prediksi $\hat{x}_{N+1|N}$ diperoleh dengan menggunakan algoritma yang akan diuji pada penelitian ini. Dari persamaan (3.16) dan (3.17), akan diperoleh nilai variabel keadaan pada waktu sekarang (x_k) dan pada waktu langkah ke depan (x_{k+1}) dari hasil perkalian antara matrik A dan B. Dari data variabel keadaan ini, nilainya akan dibandingkan dengan hasil estimasi variabel keadaan untuk waktu sekarang ($\hat{x}_{N|N}$) untuk melihat seberapa besar tingkat kesalahan yang terjadi. Faktor gangguan akan diperhitungkan untuk mengetahui pengaruhnya terhadap hasil estimasi variabel keadaan yang akan ditampilkan pada bab 4.

Secara umum, proses pengujian algoritma dan validasi dapat dilihat dari diagram di bawah ini :



Gambar 3.6 Skema Pengujian dan Validasi Algoritma untuk Mengestimasi State

Perlu dicatat bahwa blok *PLANT* pada gambar 3.5 diwakili oleh model hasil identifikasi *offline*, sedangkan blok *MOESP OBSERVER* merupakan blok yang digunakan untuk menghasilkan variabel keadaan hasil estimasi.

Untuk menguji algoritma untuk mengestimasi variabel keadaan ini, digunakan model linier sistem tata udara presisi untuk mewakili blok *PLANT* pada gambar 3.6 yang diperoleh melalui proses identifikasi menggunakan metode MOESP *offline*. Hasil identifikasi yang diperoleh adalah sebagai berikut :

$$A = \begin{bmatrix} 0.974525 & -0.01366 & -0.00441 & -0.00099 & -0.01133 & 0.018842 & -0.00418 & -0.01468 \\ -0.07666 & 0.832216 & 0.012794 & 0.17223 & 0.1086 & -0.03933 & 0.092703 & 0.020134 \\ 0.068861 & 0.075798 & 0.963319 & -0.0198 & 0.023203 & -0.01762 & 0.000708 & -0.00013 \\ 0.079615 & 0.070479 & 0.012201 & 0.516287 & -0.75091 & -0.17064 & -0.50661 & 0.317302 \\ 0.060414 & 0.008054 & -0.06385 & 0.552407 & 0.528967 & 0.143632 & -0.04947 & -0.10849 \\ 0.098599 & 0.11378 & 0.019149 & 0.097361 & 0.088844 & -0.67208 & -0.12845 & -0.32955 \\ 0.067682 & 0.054275 & 0.000742 & -0.04147 & -0.10308 & 0.393316 & 0.026605 & -0.02373 \\ -0.02239 & -0.0486 & -0.00423 & -0.08236 & -0.01574 & -0.17013 & -0.37892 & -0.45597 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.0033 & 0.008158 \\ -0.00057 & -0.00557 \\ 0.006841 & -0.01526 \\ 0.014192 & -0.03765 \\ 0.010281 & -0.02706 \\ 0.017131 & -0.04519 \\ 0.013584 & -0.03685 \\ -0.00208 & 0.004475 \end{bmatrix}$$

$$C = \begin{bmatrix} -12.1395 & -0.69088 & 0.114385 & 0.038945 & 0.328561 & -0.06891 & -0.21067 & 0.096712 \\ -0.20531 & 0.024634 & 0.395873 & 0.011623 & 0.022705 & -0.01259 & -0.01089 & 0.007233 \end{bmatrix}$$

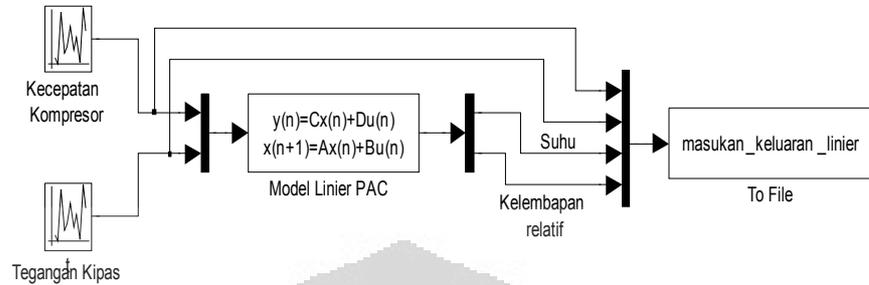
$$D = \begin{bmatrix} 0.03306 & -0.08744 \\ 0.0019 & -0.00504 \end{bmatrix}$$

Nilai eigen dari matrik A adalah sebagai berikut :

$$\text{eigen } A = \begin{bmatrix} -0.6116 + 0.1608i \\ -0.6116 - 0.1608i \\ 0.5226 + 0.6412i \\ 0.5226 - 0.6412i \\ 0.1295 \\ 0.8102 \\ 0.9761 + 0.0109i \\ 0.9761 - 0.0109i \end{bmatrix}$$

Hasil Pengujian *Observability* dari model di atas adalah 8, yang artinya adalah variabel keadaan (*state*) yang dapat diobservasi pada model linear sistem tata udara presisi ini adalah sebanyak 8 *state* dari totalnya 8 *state*. Hasil pengujian *Controllability* dari model di atas adalah 8 dari 8, yang artinya adalah variabel

kedaaan (*state*) yang dapat dikendalikan pada model linear sistem tata udara presisi ini adalah sebanyak 8 *state* dari total 8 *state*. Hasil pengujian akan ditampilkan pada bab 4.

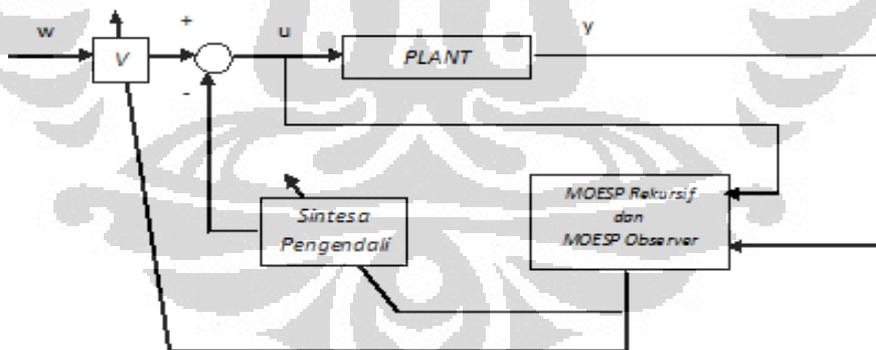


Gambar 3.7. Blok Simulink untuk Mendapatkan Data Masukan dan Keluaran Model Linier Sistem Tata Udara Presisi

Pada penelitian ini juga akan ditampilkan hasil estimasi variabel keadaan dengan menggunakan model nonlinier sistem tata udara presisi yang diwakilkan oleh gambar 3.4.

3.4 Implementasi Algoritma untuk Mengestimasi Variabel Keadaan

Untuk mengimplementasikan algoritma untuk mengestimasi *state* ini, dapat dilihat dari gambar 3.7 di bawah ini :



Gambar 3.8 Diagram Blok Implementasi Algoritma untuk Mengestimasi Variabel Keadaan

Pada penelitian ini, akan didesain suatu pengendali yang memanfaatkan hasil estimasi variabel keadaan yang nantinya akan menghasilkan sinyal pengendali yang digunakan untuk mengendalikan sistem agar sinyal keluaran (y) dari sistem mengikuti sinyal referensi (w) yang diinginkan. Diasumsikan telah diperoleh N pasangan data I/O $(u_i, y_i), i = 1, \dots, N$. Untuk menyelesaikan masalah

ini, sinyal pengendali untuk satu langkah ke depan u_{N+1} ditentukan berdasarkan langkah berikut.

Langkah pertama yang dilakukan adalah mengestimasi model $\hat{A}_N, \hat{B}_N, \hat{C}_N, \hat{D}_N$ untuk mengestimasi variabel keadaan. Model diperoleh dengan menggunakan algoritma identifikasi MOESP secara rekursif. Dalam penelitian ini, akan digunakan desain *Adaptive Linear-Quadratic Regulator* untuk model di atas. *Adaptive Linear-Quadratic Regulator (ALQR)* adalah algoritma yang digunakan untuk menghasilkan penguat optimal (K) sebagai pengganti blok pengendali sehingga persamaan (3.18) membuat fungsi kriteria pada persamaan (3.19) sekecil mungkin.

$$u_k = -K_N x_k \quad (3.18)$$

$$J(u) = \int_0^{\infty} (x^T Q x + u^T R u + 2x^T N u) dt \quad (3.19)$$

Nilai K pada persamaan (3.18) diperoleh melalui persamaan

$$K = R^{-1} (B^T S + N^T) \quad (3.20)$$

dimana nilai S merupakan solusi dari persamaan *Riccati*

$$A^T S + S A - (S B + N) R^{-1} (B^T S + N^T) + Q = 0 \quad (3.21)$$

Untuk menghasilkan sinyal pengendali pada satu langkah ke depan u_{N+1} , maka digunakan variabel keadaan untuk satu langkah ke depan $\hat{x}_{N+1|N}$ yang diperoleh melalui algoritma yang diuji pada penelitian ini. Dengan menggunakan persamaan (3.18), maka diperoleh

$$u_{N+1|N} = -K_N x_{N+1|N} \quad (3.22)$$

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

$$u(k) = Vw(k) - Kx(k) \quad (3.23)$$

Pada saat kondisi tunak, yaitu saat nilai $y_s(k) = w(k)$, nilai keadaan saat pencuplikan setelahnya sama dengan nilai keadaan sekarang, yaitu $x(k+1) = x(k) = x_s(k)$. Dengan mensubstitusi persamaan (3.23) ke dalam persamaan model proses (2.1) dan (2.2), maka diperoleh

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ &= Ax(k) + BVw(k) - BKx(k) \\ x_s(k) &= (A - BK)x_s(k) + BVw(k) \\ (I - A + BK)x_s(k) &= BVw(k) \\ x_s(k) &= (I - A + BK)^{-1} BVw(k) \end{aligned} \quad (3.24)$$

dengan $x_s(k)$ merupakan nilai keadaan tunak. Substitusi persamaan (3.23) dan (3.24) ke persamaan model proses dalam keadaan tunak menghasilkan :

$$\begin{aligned} y_s(k) &= Cx_s(k) + Du(k) \\ &= Cx_s(k) + DVw(k) - DKx_s(k) \\ &= (C - DK)x_s(k) + DVw(k) \\ &= (C - DK)(I - A + BK)^{-1} BVw(k) + DVw(k) \\ y_s(k) &= \left((C - DK)(I - A + BK)^{-1} B + D \right) Vw(k) \end{aligned} \quad (3.25)$$

Dalam keadaan tunak, keluaran sistem diharapkan sama dengan referensi, yaitu $y_s(k) = w(k)$ sehingga :

$$\begin{aligned} V &= \left((C - DK)(I - A + BK)^{-1} B + D \right)^{-1} \\ V_N &= \left((\hat{C}_N - \hat{D}_N K_N) (I - \hat{A}_N + \hat{B}_N K_N)^{-1} \hat{B}_N + \hat{D}_N \right)^{-1} \end{aligned} \quad (3.26)$$

Semua langkah di atas disimulasikan secara rekursif sampai sinyal keluaran sistem mengikuti sinyal referensi yang diinginkan. Hasil simulasi dengan desain pengendali ditampilkan pada bab 4.

Untuk perancangan ke depan, khususnya untuk sistem tata udara presisi, digunakan metode pengendali *Model Predictive Control (MPC)*. Metode pengendali ini termasuk dalam metode pengendali prediktif yang berbasiskan kepada model proses. Desain pengendali *MPC* tidak dibahas pada penulisan skripsi ini. Informasi yang dibutuhkan dalam desain pengendali *MPC* ini adalah matriks A,B,C,D dan variabel keadaan waktu sekarang $\hat{x}_{N|N}$. Oleh karena itu, desain *MPC* yang digunakan juga harus bersifat rekursif, yang berarti nilai A,B,C,D yang digunakan berubah-ubah setiap waktunya.

Secara umum ide perancangan skema estimasi *state* yang dikombinasikan dengan pengendali adalah sebagai berikut :

1. Mengumpulkan pasangan data I/O $(u_i, y_i), i = 1, \dots, n_{data}$. Sebanyak N data akan digunakan untuk mengidentifikasi sistem secara *offline*. Dari identifikasi sistem secara *offline* ini, akan diperoleh nilai A,B,C,D awal yang digunakan untuk melakukan proses identifikasi secara rekursif.
2. Melakukan proses identifikasi secara rekursif dengan metode *MOESP*. Dari proses identifikasi ini akan dihasilkan $\hat{A}_N, \hat{B}_N, \hat{C}_N, \hat{D}_N$, nilai yang dihasilkan pada langkah ke- N .
3. Melakukan proses estimasi variabel keadaan dengan menggunakan algoritma yang diuji dalam penelitian ini. Proses estimasi ini akan menghasilkan variabel keadaan waktu sekarang $\hat{x}_{N|N}$ dan variabel keadaan untuk satu langkah ke depan pada waktu sekarang $\hat{x}_{N+1|N}$.
4. Proses pengendali dengan menggunakan algoritma *MPC/LQR*. Keluaran dari pengendali ini akan menghasilkan sinyal pengendali pada satu langkah ke depan u_{N+1} . Diasumsikan bahwa sinyal kendali waktu sekarang u_N bernilai 0, sehingga untuk menghasilkan u_{N+1} , *MPC/LQR* membutuhkan informasi variabel keadaan untuk satu langkah ke depan pada waktu sekarang $\hat{x}_{N+1|N}$.

BAB 4

SIMULASI DAN ANALISIS

Bab ini menjelaskan hasil simulasi yang didapat dari percobaan seperti yang telah dijelaskan pada perancangan percobaan pada Bab 3. Percobaan yang dilakukan menggunakan waktu pencuplikan selama 5 detik. Data yang dipakai dalam proses identifikasi dan estimasi variabel keadaan sebanyak 1000 data sehingga jumlah keseluruhan waktu yang dibutuhkan adalah 5000 data.

Bab ini dibagi menjadi beberapa subbab. Subbab yang pertama adalah validasi estimasi variabel keadaan yang dilakukan dengan metode MOESP secara rekursif yang ditambahkan dengan algoritma yang diuji pada penelitian ini. Tujuan dari subbab ini adalah untuk membandingkan hasil estimasi variabel keadaan dengan nilai variabel keadaan yang diperoleh melalui persamaan (3.16) dan (3.17) yang merupakan representasi dari sistem yang sebenarnya. Dalam subbab ini juga dijelaskan hasil validasi dengan menggunakan data I/O dari model linier dan model nonlinier. Hasil validasi dengan variasi *noise* akan ditampilkan pada subbab ini.

Pada subbab berikutnya, ditunjukkan hasil implementasi algoritma untuk mengestimasi variabel keadaan yang dikombinasikan dalam desain pengendali. Sinyal keluaran sistem hasil pengendali akan ditampilkan dalam grafik. Kriteria untuk menentukan tingkat validasi hasil estimasi *state* menggunakan kriteria kuadrat kesalahan yang ditunjukkan persamaan di berikut :

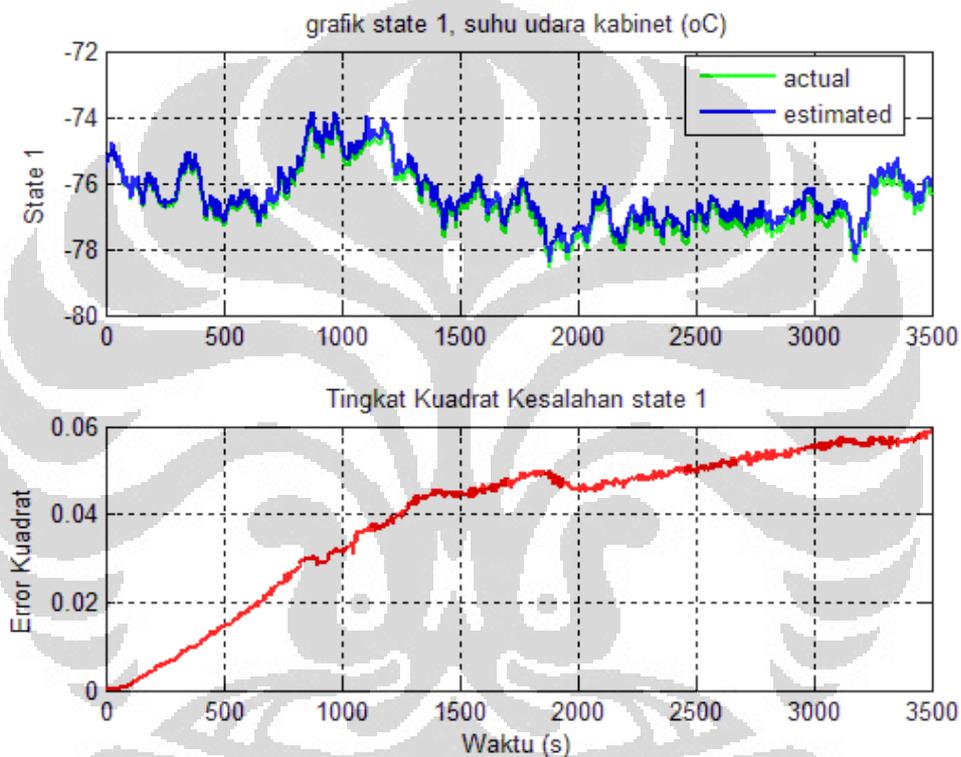
$$\text{Kriteria}_{\text{ Kesalahan}} = (x_i - \hat{x}_i)^2 \quad (4.1)$$

4.1 Estimasi Variabel Keadaan dengan Identifikasi MOESP Rekursif

Simulasi pertama yang dilakukan adalah estimasi variabel keadaan dengan menggunakan identifikasi MOESP rekursif. Seperti yang telah dijelaskan sebelumnya, subbab ini menampilkan hasil estimasi variabel keadaan dan hasil validasinya untuk model linier dan model nonlinier. Untuk kondisi awal, jumlah data *offline* (N) yang digunakan berjumlah 300 data, sedangkan jumlah data yang

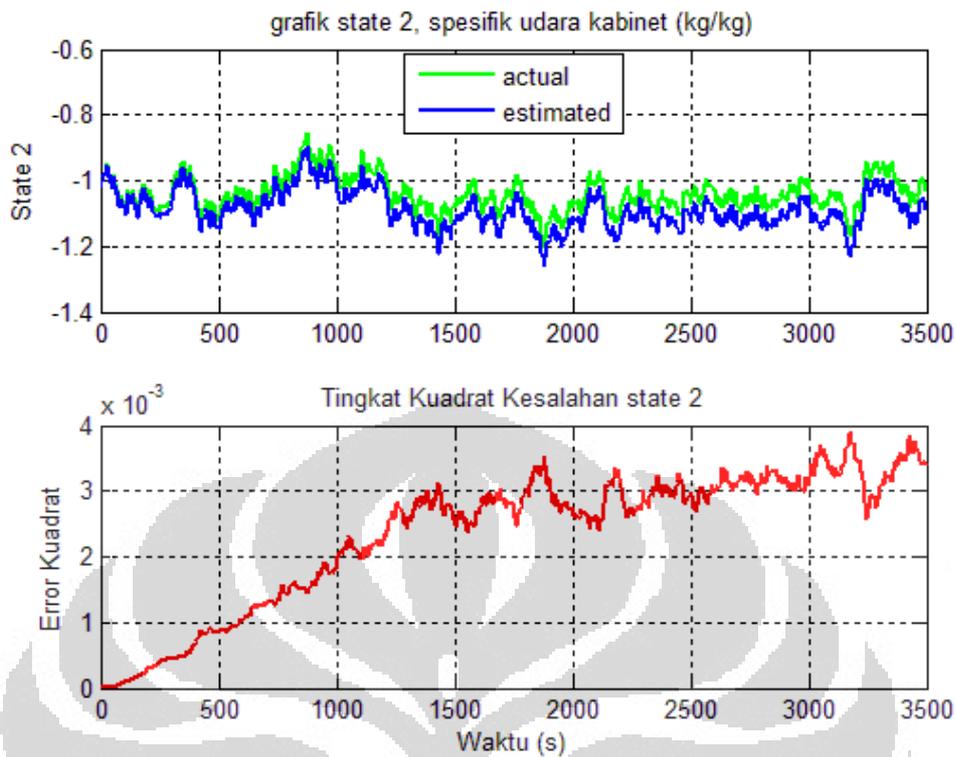
digunakan dalam perhitungan rekursif sebanyak 701 data. Jumlah baris hankel (v) yang digunakan berjumlah 15 baris dan faktor pelupa sama dengan 1. Grafik berwarna merah menunjukkan tingkat kesalahan hasil estimasi $(x_i - \hat{x}_i)^2$, sedangkan grafik berwarna hijau dan biru masing-masing menunjukkan variabel keadaan dari model sistem tata udara presisi dan variabel keadaan hasil estimasi. Berikut adalah hasil estimasi variabel keadaan yang dilakukan.

4.1.1 Model Linier Sistem Tata Udara Presisi



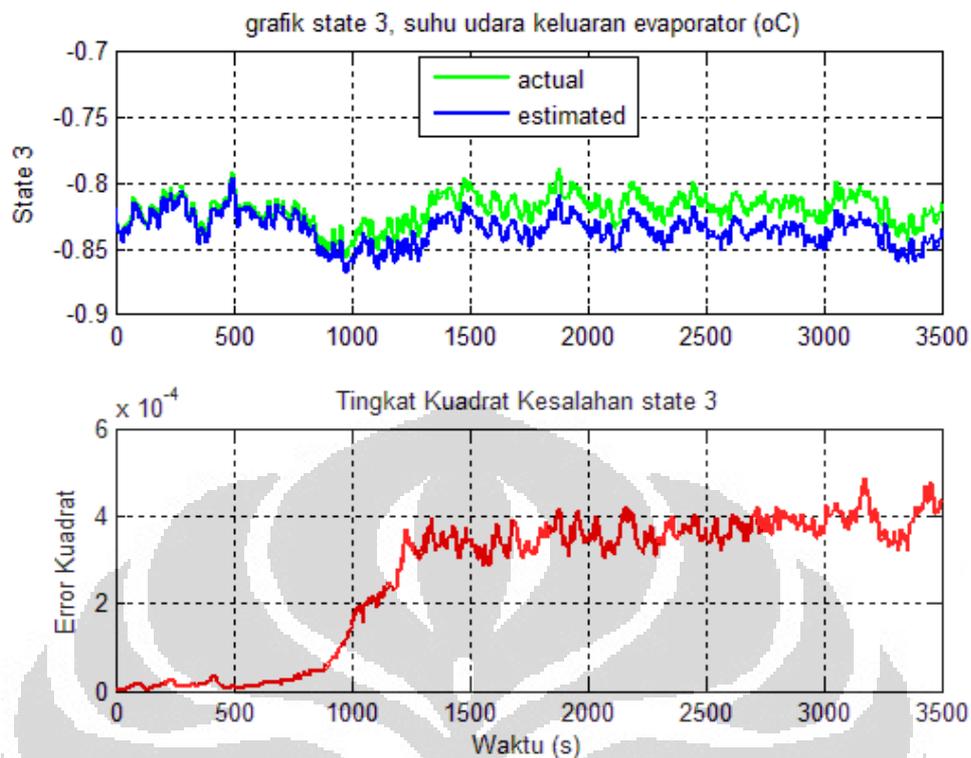
Gambar 4.1 Perbandingan *state actual* dengan *state* hasil estimasi untuk *state 1* dan Tingkat kesalahan *state 1*

Gambar 4.1 menunjukkan perbandingan antara grafik *state actual* dengan *state* hasil estimasi dan tingkat kuadrat kesalahan untuk *state 1*. Terlihat bahwa untuk grafik *state* hasil estimasi yang ditunjukkan grafik berwarna hijau dapat mengikuti *state actual* yang ditunjukkan grafik berwarna biru. Pada grafik kuadrat kesalahan yang ditunjukkan grafik berwarna merah, nilai $(x_i - \hat{x}_i)^2$ terlihat cenderung mengalami kenaikan hingga nilai 0.06 pada detik ke 3500.



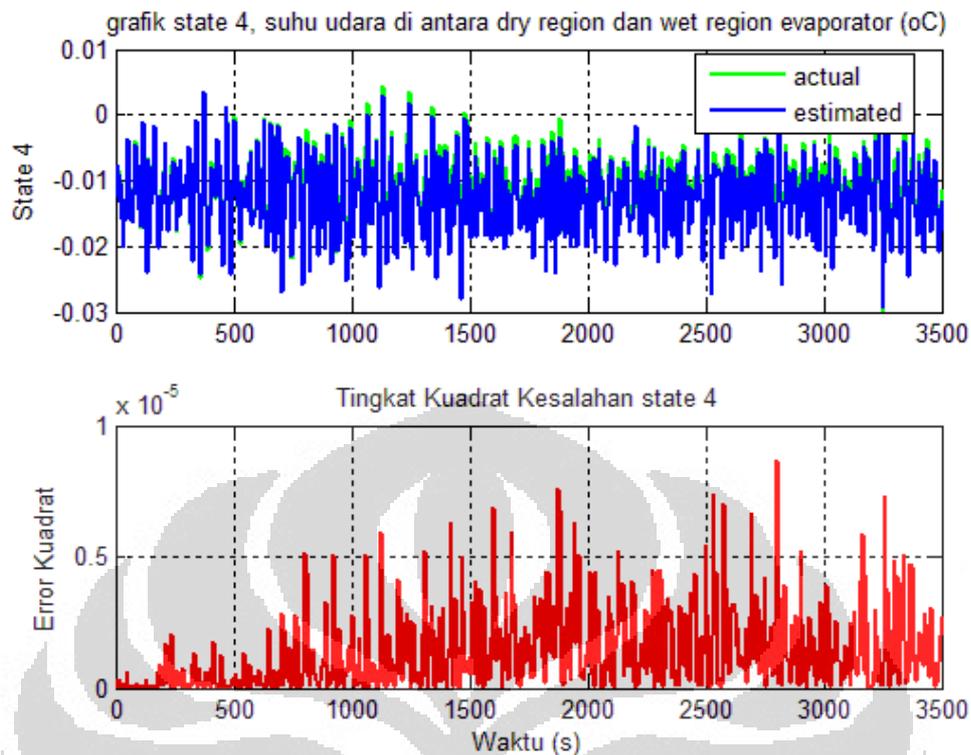
Gambar 4.2 Perbandingan *state actual* dengan *state* hasil estimasi untuk *state 2* dan Tingkat kesalahan *state 2*

Gambar 4.2 merupakan grafik perbandingan *state actual* dengan *state* hasil estimasi dan tingkat kuadrat kesalahan untuk *state 2*. Dari hasil grafik yang telah diperoleh, terlihat bahwa hasil estimasi *state* untuk *state 2* ini relatif cukup baik. Kondisi yang sama seperti pada *state 1* terjadi grafik tingkat kesalahan yang ditunjukkan grafik berwarna merah, yaitu terjadi kenaikan nilai $(x_i - \hat{x}_i)^2$.



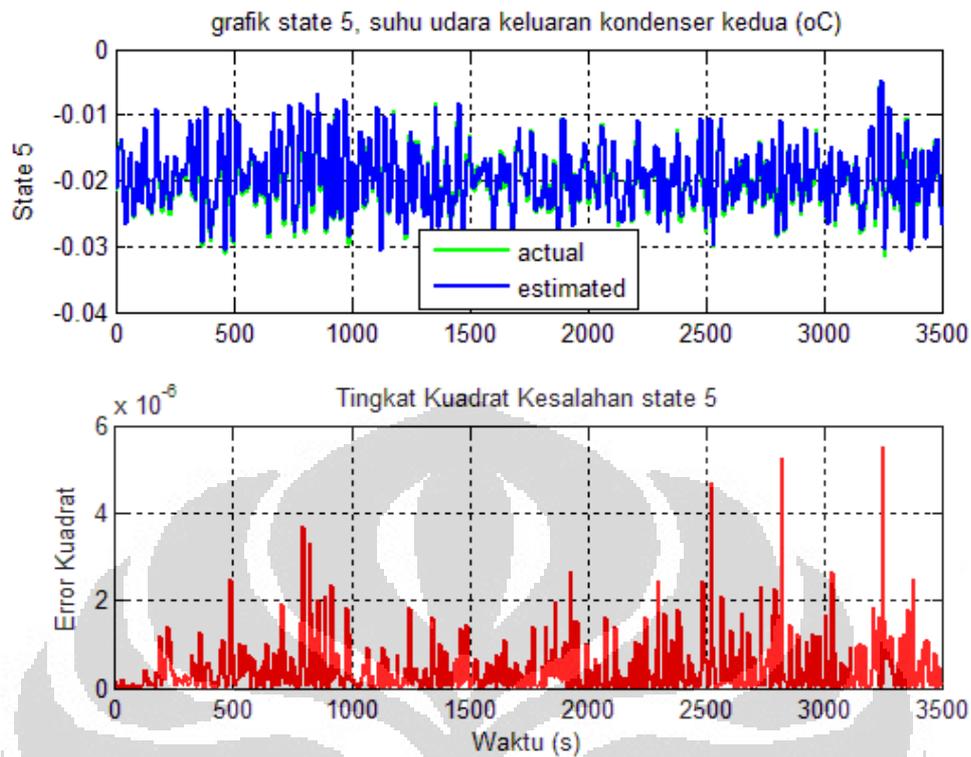
Gambar 4.3 Perbandingan *state actual* dengan *state* hasil estimasi untuk *state 3* dan Tingkat kesalahan *state 3*

Dari gambar 4.3, terlihat bahwa hasil estimasi *state* untuk *state 3* relatif cukup baik. Hal ini terlihat dari tingkat kuadrat kesalahan bernilai mendekati hampir nol saat memasuki detik ke 0 hingga detik ke 800. Saat memasuki detik ke 801 hingga detik ke 3500, terjadi kenaikan nilai kuadrat kesalahan yang konstan di nilai 4×10^{-4} . Pada gambar 4.3, kecenderungan yang terjadi adalah nilai *state* hasil estimasi yang ditunjukkan grafik berwarna biru berada di bawah *state actual* yang ditunjukkan grafik berwarna hijau.



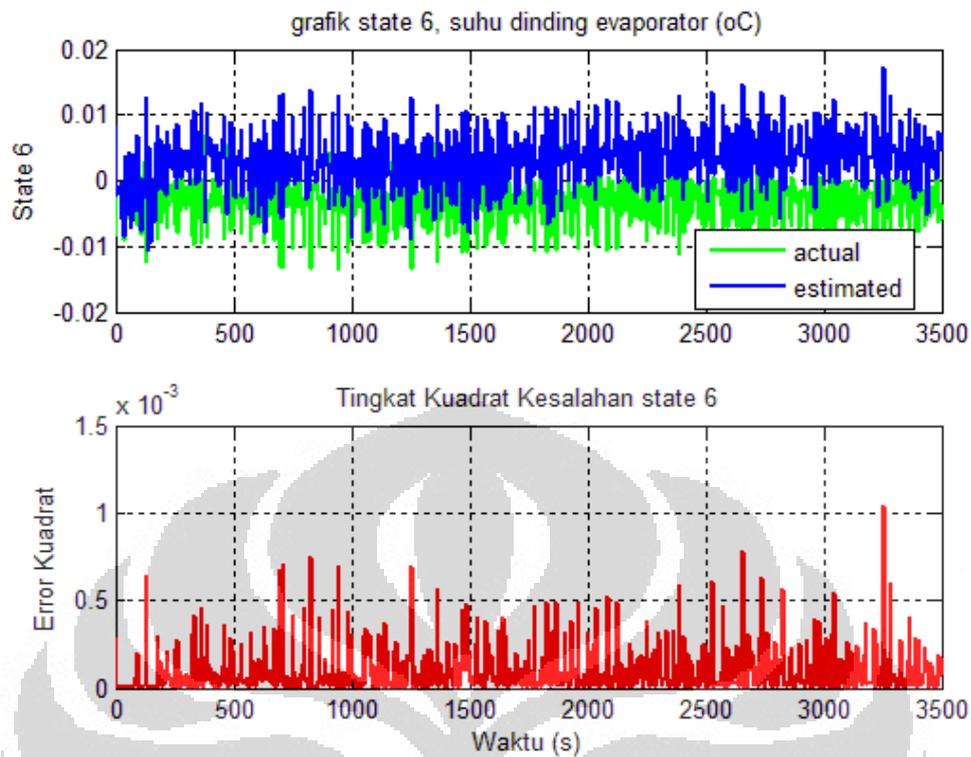
Gambar 4.4 Perbandingan *state actual* dengan *state* hasil estimasi untuk *state 4* dan Tingkat kesalahan *state 4*

Pada grafik perbandingan *state actual* dengan *state* hasil estimasi untuk *state 4* yang ditunjukkan gambar 4.4, terlihat bahwa hasil estimasi *state* dapat mengikuti nilai *state actual* yang ditunjukkan grafik berwarna hijau dengan baik. Tingkat kuadrat kesalahan yang terjadi sangat kecil, walaupun terjadi sedikit kenaikan nilai error kuadrat saat memasuki detik ke 800. Nilai kesalahan yang terjadi berkisar pada nilai 0 hingga 1×10^{-5} .



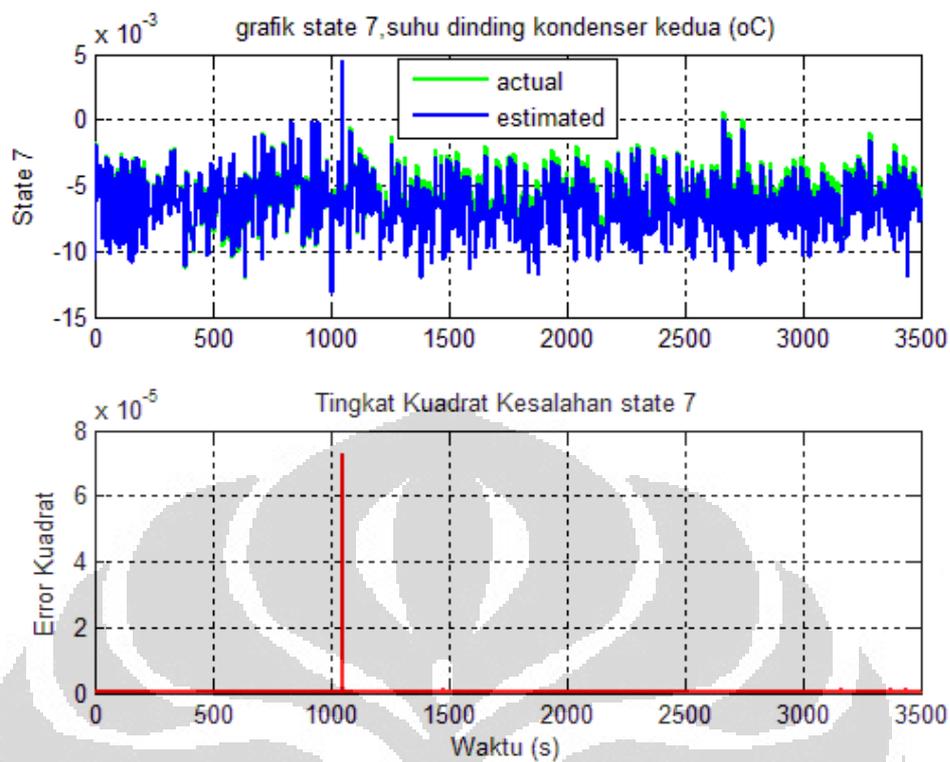
Gambar 4.5 Perbandingan *state actual* dengan *state* hasil estimasi untuk *state 5* dan Tingkat kesalahan *state 5*

Hasil estimasi *state* untuk *state 5* sudah sangat baik. Hasil estimasi *state* yang ditunjukkan grafik berwarna biru dapat mengikuti nilai *actualnya* dengan baik. Tingkat kuadrat kesalahan yang terjadi relatif kecil dan konstan, yaitu pada kisaran nilai 2×10^{-6} .



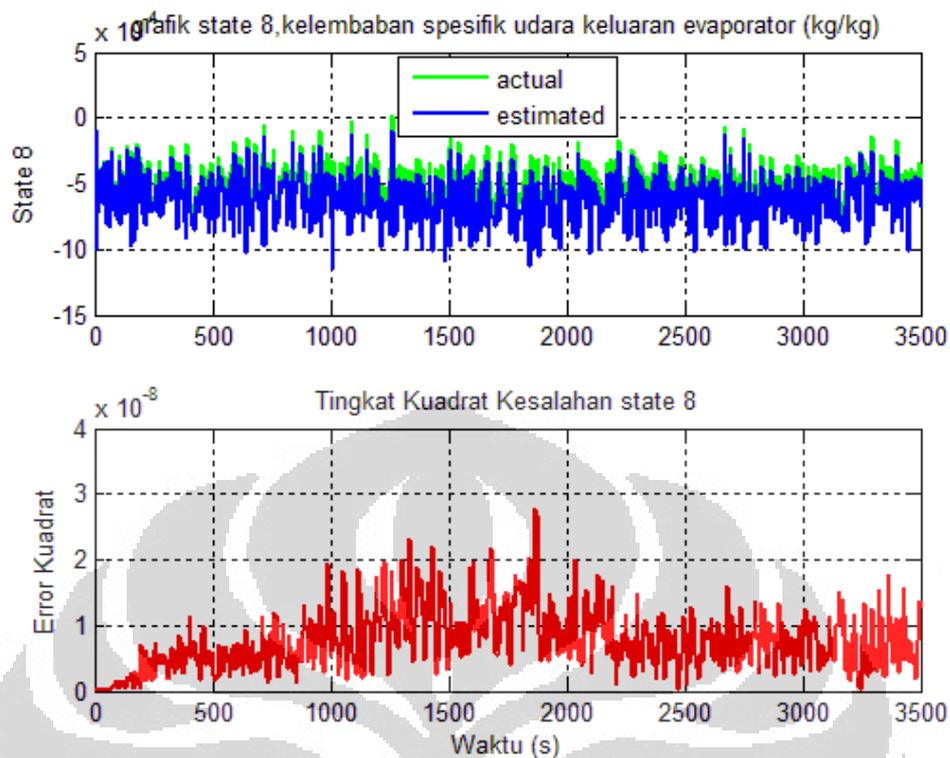
Gambar 4.6 Perbandingan *state actual* dengan *state* hasil estimasi untuk *state 6* dan Tingkat kesalahan *state 6*

Gambar 4.6 menunjukkan bahwa hasil estimasi *state* untuk *state 6* cukup baik, walaupun kecenderungan yang terjadi adalah hasil estimasi *state* yang ditunjukkan grafik berwarna biru berada pada di atas grafik berwarna hijau yang merupakan grafik *state actual*. Tingkat kuadrat kesalahan yang terjadi relatif konstan, yaitu pada kisaran nilai 0 hingga 0.5×10^{-3}



Gambar 4.7 Perbandingan *state actual* dengan *state* hasil estimasi untuk *state 7* dan Tingkat kesalahan *state 7*

Pada gambar 4.7, tingkat kuadrat kesalahan untuk *state 7* yang ditunjukkan grafik berwarna merah memiliki kecenderungan nilai konstan pada nilai 0. Jika dibandingkan dengan grafik kuadrat kesalahan *state* yang lain, tingkat kuadrat kesalahan untuk *state 7* ini memiliki nilai yang paling kecil. Hal ini membuktikan bahwa hasil estimasi *state 7* memiliki nilai yang sangat akurat.



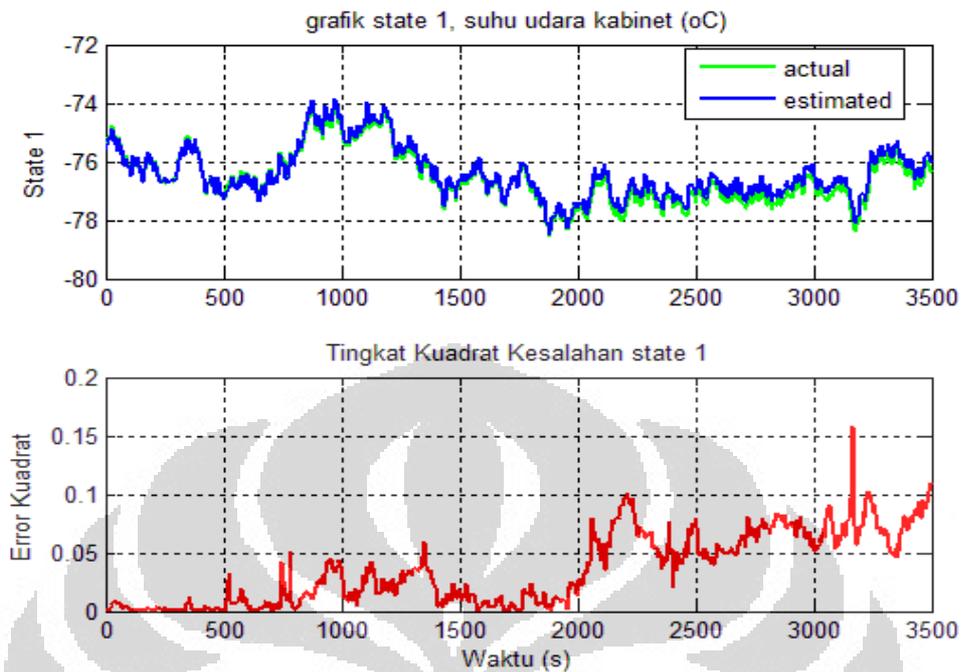
Gambar 4.8 Perbandingan *state actual* dengan *state* hasil estimasi untuk *state 8* dan Tingkat kesalahan *state 8*

Pada estimasi *state 8* yang ditunjukkan oleh gambar 4.8, terlihat bahwa hasil estimasi untuk *state* juga sudah relatif baik, walaupun terjadi sedikit perbedaan antara grafik berwarna hijau dengan grafik berwarna biru. Grafik kuadrat kesalahan juga relatif kecil, yaitu berkisar antara 0 hingga 3×10^{-8} .

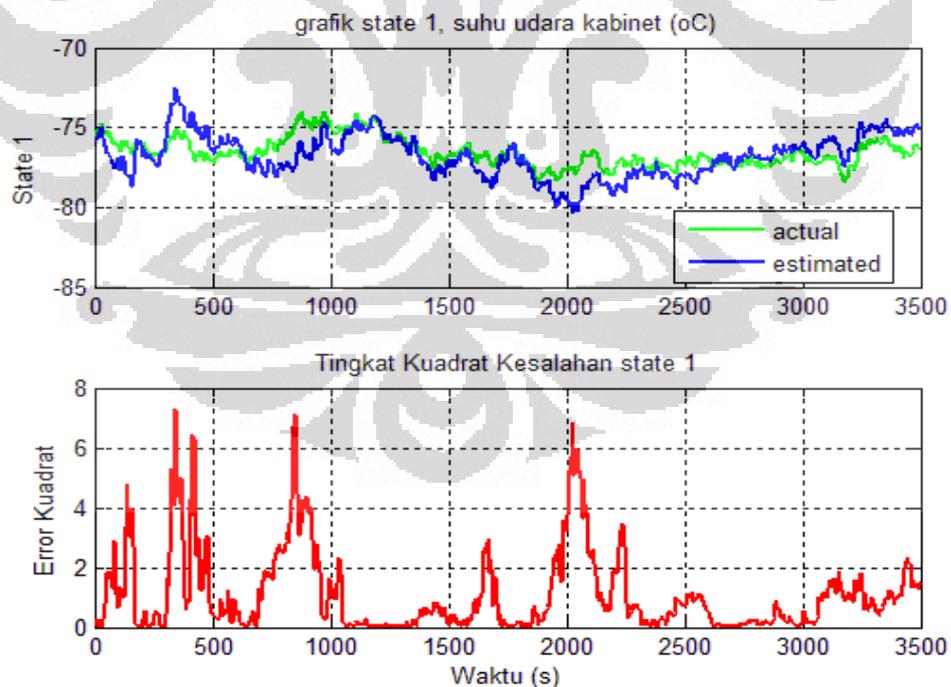
Pada grafik *state 1, 2 dan 3*, kecenderungan yang terjadi adalah grafik tingkat kuadrat kesalahan berwarna merah memiliki tren naik seiring dengan bertambahnya waktu komputasi. Hal ini disebabkan karena pengaruh model sistem yang diperoleh dan nilai *eigen value* dari model sistem.

4.1.2 Estimasi state Model Linier Sistem Tata Udara Presisi dengan Variasi noise

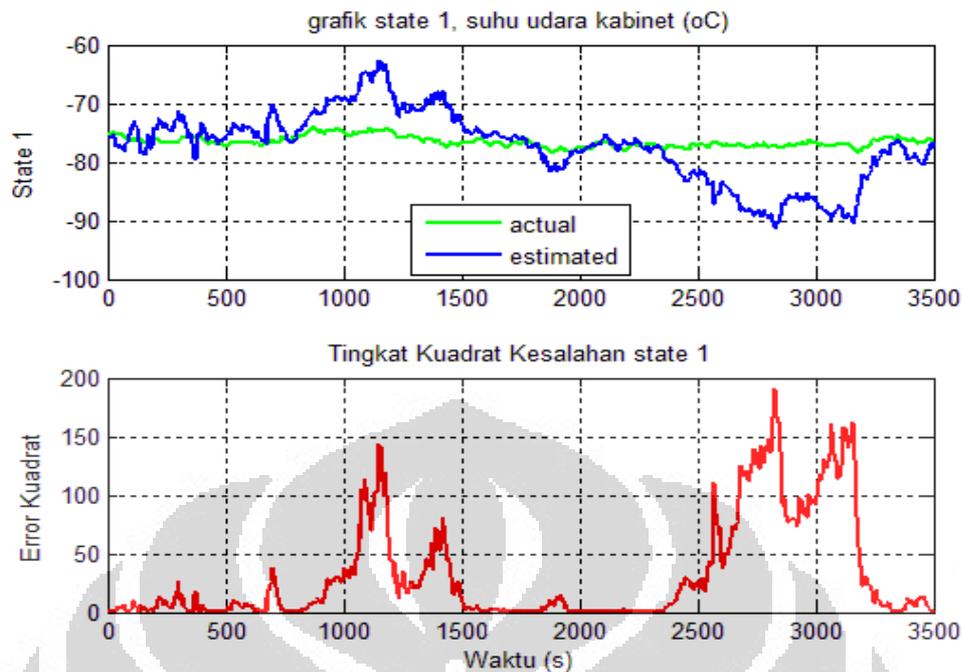
Pada subbab ini, ditampilkan grafik perbandingan hasil estimasi *state* dengan *state actual* dengan variasi *Spectral Density Gaussian Noise* sebesar 0.0001, 0.01 dan 0.1.



Gambar 4.9. Perbandingan *state actual* dengan *state* hasil estimasi dan Tingkat kesalahan *state 1* untuk *Spectral Density Gaussian Noise* sebesar 0.0001



Gambar 4.10. Perbandingan *state actual* dengan *state* hasil estimasi dan Tingkat kesalahan *state 1* untuk *Spectral Density Gaussian Noise* sebesar 0.01



Gambar 4.11. Perbandingan *state actual* dengan *state* hasil estimasi dan Tingkat kesalahan *state 1* untuk *Spectral Density Gaussian Noise* sebesar 0.1

Gambar 4.9, 4.10 dan 4.11 merupakan grafik perbandingan antara *state actual* dengan *state* hasil estimasi untuk *state 1* dengan variasi *Spectral Density Gaussian Noise*. Untuk nilai *Spectral Density Gaussian Noise* sebesar 0.0001 ditunjukkan oleh gambar 4.9. Untuk nilai *Spectral Density Gaussian Noise* sebesar 0.01 ditunjukkan oleh gambar 4.10, sedangkan untuk nilai *Spectral Density Gaussian Noise* sebesar 0.1 ditunjukkan oleh gambar 4.11. Gambar 4.9 menunjukkan hasil estimasi *state* yang ditunjukkan grafik berwarna biru dapat mengikuti dengan baik grafik berwarna hijau yang merupakan nilai *state actual*. Jika dilihat dari grafik tingkat kuadrat kesalahan, terlihat bahwa kesalahan estimasi yang terjadi relatif kecil, yaitu berkisar pada nilai 0 hingga 0.15. Dari gambar 4.9, terlihat bahwa tingkat kuadrat kesalahan mengalami kenaikan saat memasuki detik ke 2000.

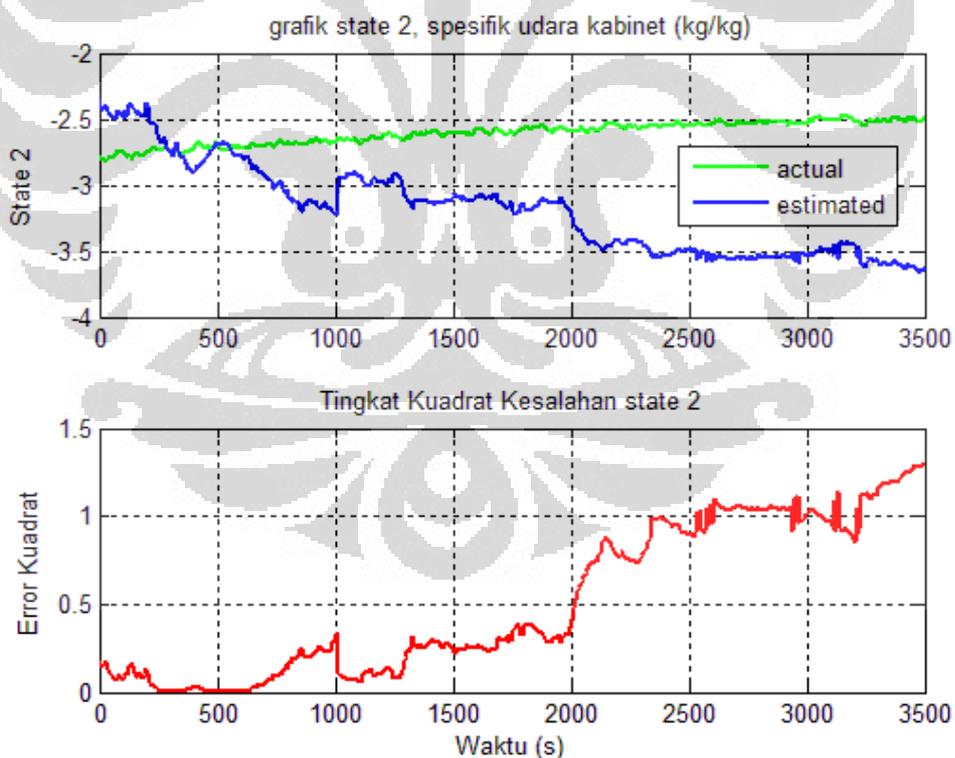
Gambar 4.10 menunjukkan hasil estimasi *state* masih relatif cukup baik walaupun tingkat kuadrat kesalahannya lebih jelek daripada grafik tingkat kuadrat kesalahan pada gambar 4.9. Dari grafik tingkat kuadrat kesalahan yang

ditunjukkan oleh grafik berwarna merah, tingkat kuadrat kesalahan mendekati nilai 0 terjadi pada kisaran detik 1000-2000 dan detik 2500-3500.

Pada gambar 4.11, terlihat bahwa hasil estimasi cukup baik, walaupun pada kisaran detik 1000 hingga 1500 dan 2500 hingga 3200 tingkat kuadrat kesalahannya tidak baik, yaitu mencapai nilai 150. Hal ini disebabkan karena faktor *noise* yang cukup besar yang mempengaruhi hasil estimasi *state*. Pada kisaran detik yang lainnya, tingkat kuadrat kesalahan yang terjadi mencapai nilai mendekati nol. Grafik perbandingan hasil estimasi *state* dengan *state actual* serta grafik tingkat kuadrat kesalahan *state* yang lainnya dengan variasi *noise* dapat dilihat pada bagian lampiran B, C, dan D.

4.1.3 Estimasi state Model nonlinier Sistem Tata Udara Presisi

Pada bagian ini akan ditampilkan hasil estimasi variabel keadaan untuk *state 2*, yaitu spesifik udara kabinet. Untuk hasil estimasi variabel keadaan yang lainnya, dapat dilihat pada bagian lampiran E.



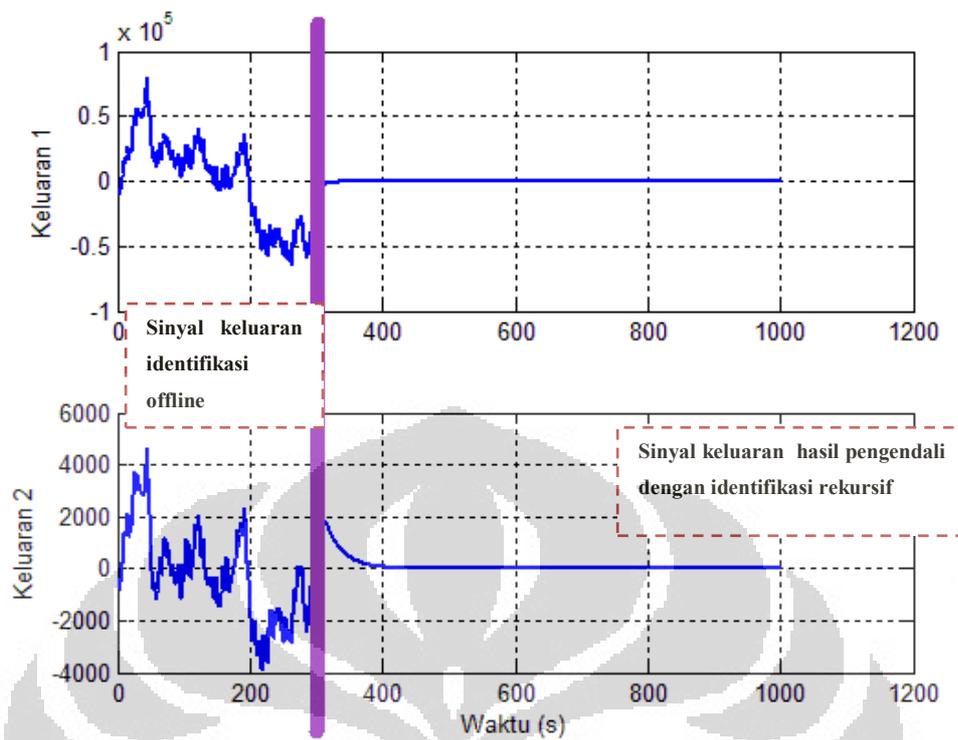
Gambar 4.12 Perbandingan state actual dengan state hasil estimasi untuk state 2 dan Tingkat kesalahan state 2 untuk model nonlinier sistem tata udara presisi

Gambar 4.12 merupakan perbandingan antara variabel keadaan *actual* yang ditunjukkan grafik berwarna hijau dengan variabel keadaan hasil estimasi yang ditunjukkan grafik berwarna biru untuk model nonlinier sistem tata udara presisi. Pada 1000 detik pertama, hasil estimasi *state* cukup bisa mengikuti nilai *actual*-nya, namun setelah melewati detik ke 1000, kecenderungan yang terjadi adalah nilai kuadrat kesalahan mengalami kenaikan. Perlu diketahui bahwa faktor gangguan diabaikan pada percobaan dengan menggunakan model nonlinier sistem tata udara presisi ini. Hasil estimasi *state* yang lainnya juga menunjukkan bahwa hasil estimasi *state* belum bisa mengikuti dengan baik nilai *state actual*-nya.

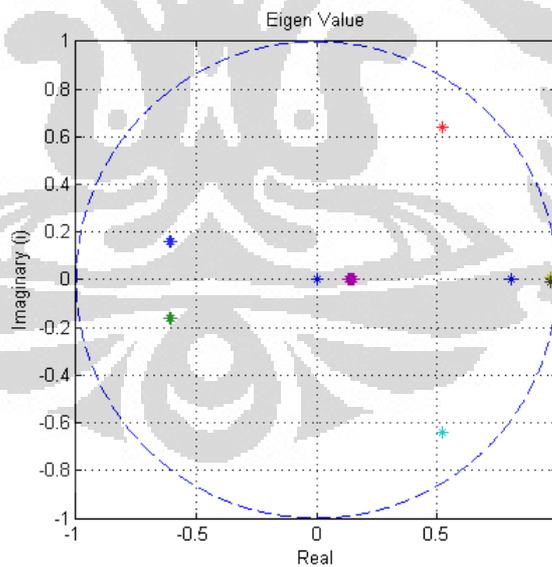
4.2 Implementasi Algoritma dengan Desain Pengendali

Untuk mengimplementasikan algoritma untuk mengestimasi *state*, maka didesain pengendali yang memanfaatkan informasi *state* hasil estimasi. Diasumsikan sistem yang tidak diketahui dideskripsikan sebagai persamaan (2.1) dan (2.2) dengan koefisien matriks.

Tujuan dari pengendali ini adalah untuk mengendalikan sinyal keluaran (y) mengikuti sinyal referensi yang diinginkan (w). Pada percobaan ini, sinyal referensi yang diinginkan adalah 24 dan 5, masing-masing untuk keluaran 1 dan 2. Karena koefisien matriks A,B,C,D di atas diasumsikan tidak diketahui, maka terlebih dahulu dilakukan proses identifikasi untuk mendapatkan $\hat{A}_N, \hat{B}_N, \hat{C}_N, \hat{D}_N$ yang digunakan untuk mengestimasi variabel keadaan (*state*). Untuk mendapatkan pasangan data masukan dan keluaran, menggunakan blok *simulink* pada gambar 3.7. Desain pengendali yang digunakan adalah *Adaptive Linear-quadratic regulator (ALQR)*. Desain pengendali ini membutuhkan informasi *state* dari sistem. Desain pengendali ini secara lebih rinci telah dijelaskan pada bab sebelumnya.



Gambar 4.13 Sinyal Keluaran Hasil Pengendali *ALQR*



Gambar 4.14. Nilai perubahan *eigen value* selama proses pengendalian secara rekursif

Sinyal masukan yang digunakan untuk proses identifikasi *offline* diatur pada dengan *range* nilai -10000 sampai dengan 10000. Nilai *range* dibuat sebesar

mungkin untuk mencegah sinyal masukan hasil pengendali yang dihasilkan LQR secara rekursif pada langkah pembaruan berikutnya berada di luar *range* data masukan yang telah ditentukan sebelumnya untuk proses identifikasi model secara *offline* yang menyebabkan model yang dihasilkan tidak stabil dan sistem tidak dapat dikendalikan.

Gambar 4.13 merupakan grafik sinyal keluaran. Sebelah kiri garis ungu merupakan sinyal keluaran pada saat proses identifikasi *offline*, sedangkan sebelah kanan garis berwarna ungu merupakan sinyal keluaran hasil pengendali selama proses identifikasi rekursif. Proses identifikasi *offline* berlangsung hingga detik ke 300, sementara proses identifikasi rekursif berlangsung mulai detik ke 301 hingga ke 1000. Dari gambar 4.13 terlihat bahwa untuk sinyal keluaran 1 hasil pengendali sudah mengikuti sinyal referensi yang diinginkan, sedangkan untuk sinyal keluaran 2 hasil pengendali, sudah mengikuti sinyal referensi yang diinginkan saat memasuki detik ke 400.

Gambar 4.14 merupakan gambar perubahan nilai *eigen* selama proses rekursif. Terlihat bahwa model yang diperoleh masih stabil. Hal ini ditunjukkan dengan nilai *eigen* yang masih berada di dalam *unit circle*.

BAB V

KESIMPULAN

5.1 Kesimpulan

Kesimpulan dari hasil penelitian yang dilakukan adalah sebagai berikut :

1. Hasil Estimasi *state* dengan menggunakan algoritma MOESP rekursif pada model linier sistem tata udara presisi menghasilkan hasil estimasi *state* yang akurat dengan tingkat kuadrat kesalahan terbesar 0.06 untuk *state* 1.
2. Hasil estimasi *state* dengan menggunakan algoritma MOESP rekursif pada model linier sistem tata udara presisi cukup akurat untuk beberapa variasi *spectral density gaussian noise* yang diberikan.
3. Hasil estimasi *state* terbaik pada model linier sistem tata udara presisi diperoleh dengan nilai *spectral density gaussian noise* sebesar 0.0001 jika dilihat dari kriteria kesalahan kuadrat.
4. Hasil estimasi *state* dengan menggunakan algoritma MOESP rekursif untuk model linier sistem tata udara presisi dapat digunakan untuk proses desain pengendali yang dalam penelitian ini menggunakan desain pengendali *LQR (Adaptive Linear-Quadratic Regulator)*.
5. Estimasi *state* dengan menggunakan algoritma MOESP rekursif belum dapat mengestimasi *state* untuk model nonlinier sistem tata udara presisi.

5.2 Saran

Untuk penelitian selanjutnya, untuk menguji algoritma estimasi *state* ini, dapat menggunakan desain pengendali *MPC* yang bersifat rekursif.

DAFTAR PUSTAKA

- [1] Katayama, T. (2005). *Subspace Methods for System Identification*. Kyoto: Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University,
- [2] Oku, H. (2000). *Sequential subspace state-space system identification and state estimation of unknown multivariable systems*. Tokyo: Department of Mathematical Engineering and Information Physics, University of Tokyo.
- [3] Surayuda, R. H. (2010). *Identifikasi Sistem Tata Udara Presisi Menggunakan Metode Subspace N4SID*. Depok: Departemen Elektro, Fakultas Teknik, Universitas Indonesia.
- [4] Victor. (2010). *Identifikasi Model Ruang Keadaan dengan Vektor Pengkoreksi pada Sistem Tata Udara Presisi Menggunakan Algoritma N4SID*. Depok: Departemen Elektro, Fakultas Teknik, Universitas Indonesia.
- [5] Victor. (2011). *Identifikasi Model Ruang Keadaan Multivariabel pada Sistem Tata Udara Presisi Menggunakan Algoritma Subspace State-Space System Identification (4SID)*. Depok: Departemen Elektro, Fakultas Teknik, Universitas Indonesia.

LAMPIRAN A

Penurunan persamaan untuk mencari matriks koefisien (A_T, B_T, C_T, D)

Pertama, menghitung C_T dan A_T . Dari $\Gamma_v = E_N T \cdot E_N$ berisi:

$$E_N = \begin{bmatrix} C_T \\ C_T A_T \\ \vdots \\ C_T A_T^{v-1} \end{bmatrix}$$

Lalu didapatkan C_T yang merupakan r baris paling atas dari E_N .

Kemudian, mencari $E_N^{(1)}$ dan $E_N^{(2)}$ dengan cara:

$$E_N^{(1)} = \begin{bmatrix} C_T \\ \vdots \\ C_T A_T^{v-2} \end{bmatrix}, \quad E_N^{(2)} = \begin{bmatrix} C_T A_T \\ \vdots \\ C_T A_T^{v-1} \end{bmatrix}$$

Hubungan antara $E_N^{(1)}$ dan $E_N^{(2)}$ yang disebut sebagai *shift invariance* yang adalah $E_N^{(1)} A_T = E_N^{(2)}$

Sehingga, A_T dengan:

$$A_T = (E_N^{(1)})^\dagger E_N^{(2)}$$

di mana $(E_N^{(1)})^\dagger$ melambangkan invers matriks Moore-Penrose dari $E_N^{(1)}$.

Kedua, menghitung B_T dan D_T . $(E_N^\perp)^T E_N = 0$ dan $E_N T = \Gamma_v$ menghasilkan $(E_N^\perp)^T \Gamma_v = 0$. Perkalian dari (2.6) dengan $(E_N^\perp)^T$ menghasilkan

$$(E_N^\perp)^T Y_N = (E_N^\perp)^T H_v U_N$$

Kemudian

$$(E_N^\perp)^T H_v = (E_N^\perp)^T Y_N U_N^T (U_N U_N^T)^{-1} \tag{A.1}$$

$(E_N^\perp)^T$ dan $(E_N^\perp)^T Y_N U_N^T (U_N U_N^T)^{-1}$ dapat dibagi menjadi matriks:

$$(E_N^\perp)^T = [a_1 \quad a_2 \quad \dots \quad a_v]$$

$$(E_N^\perp)^T Y_N U_N^T (U_N U_N^T)^{-1} = [b_1 \quad b_2 \quad \dots \quad b_v]$$

Di mana a_i dan b_i merupakan matriks $(mv - n) \times m$ dan $(mv - n) \times r$.

Dengan lambang yang lebih sederhana,

$$H_v := \begin{bmatrix} h_0 & & & 0 \\ h_1 & h_0 & & \\ \vdots & \vdots & \ddots & \\ h_{v-1} & h_{v-2} & \cdots & h_0 \end{bmatrix}, \quad h_i := \begin{cases} D, & i = 0, \\ CA^{i-1}B, & i > 0. \end{cases}$$

Setelah itu, dari (A.1),

$$[a_1 \ a_2 \ \cdots \ a_v] H_v = [b_1 \ b_2 \ \cdots \ b_v] \quad (\text{A.2})$$

Kemudian menghasilkan

$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_v \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & \cdots & a_v \\ a_2 & & & a_v \\ \vdots & a_v & & \\ a_v & & & 0 \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{v-1} \end{bmatrix} \quad (\text{A.3})$$

catatan,

$$h_i = CA^{i-1}B = C_T A_T^{i-1} B_T, \quad \text{untuk } i > 0$$

Kita mendapatkan

$$\begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{v-1} \end{bmatrix} = \begin{bmatrix} D \\ C_T \\ \vdots \\ C_T A_T^{v-2} \end{bmatrix} B_T = \begin{bmatrix} I & 0 \\ 0 & E_N^{(1)} \end{bmatrix} \begin{bmatrix} D \\ B_T \end{bmatrix} \quad (\text{A.4})$$

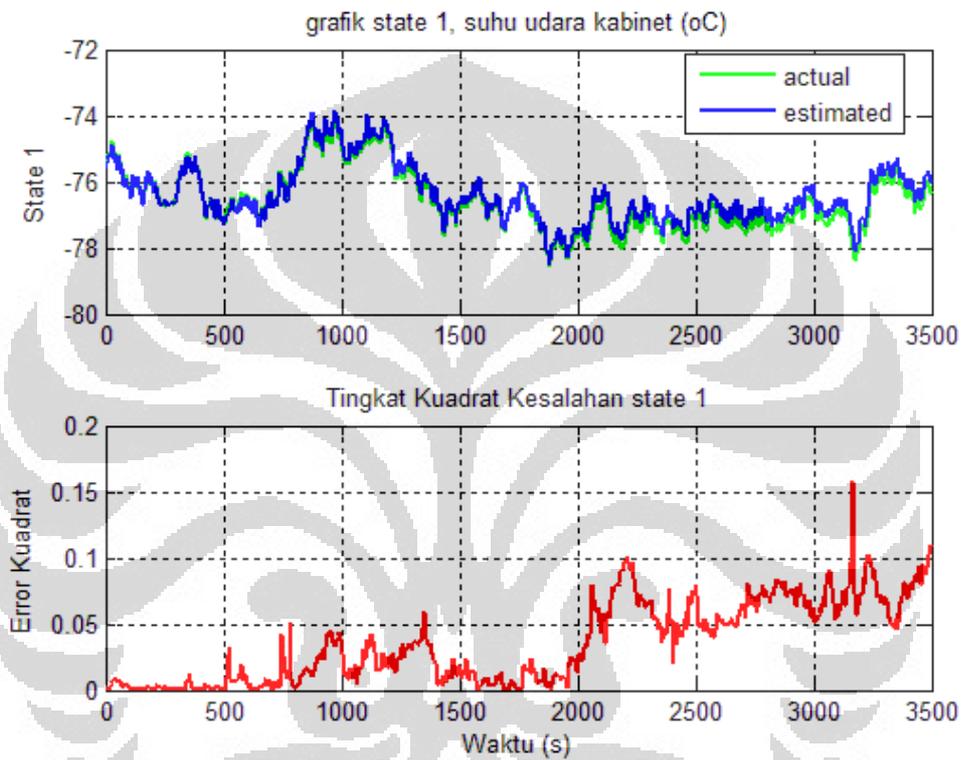
Sehingga, (A.3) dan (A.4) menghasilkan hubungan berikut:

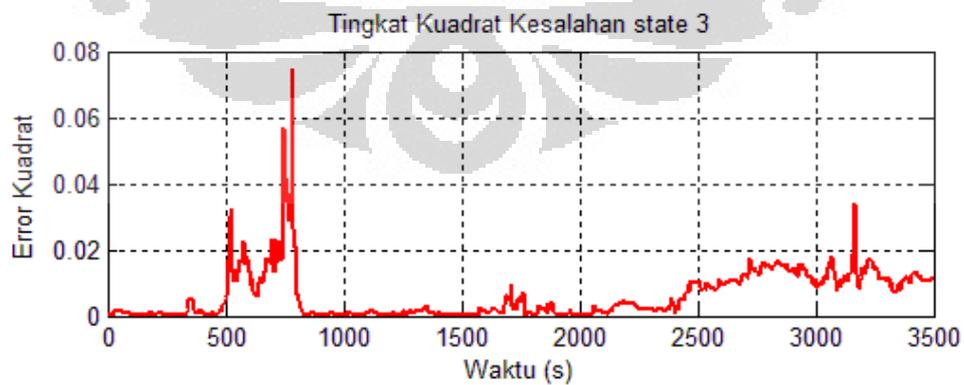
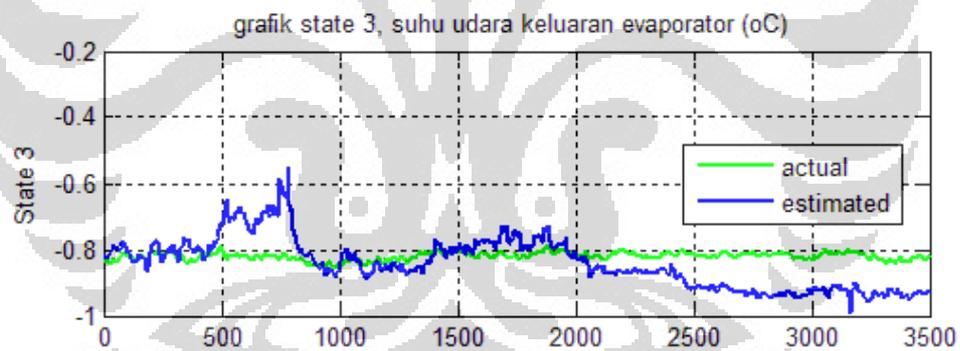
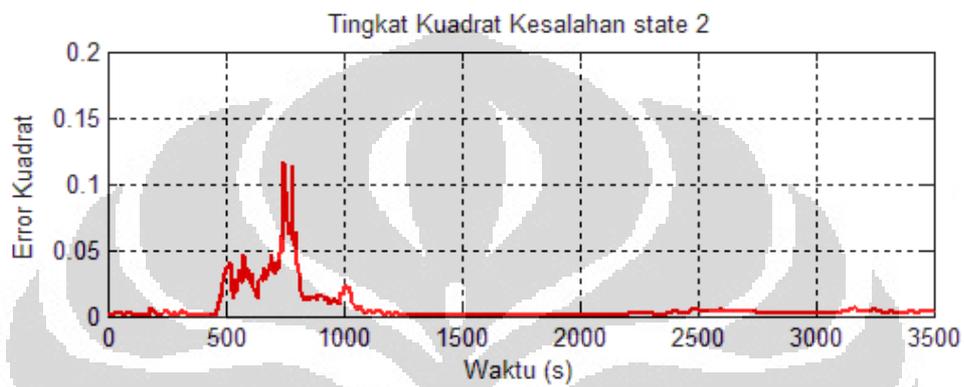
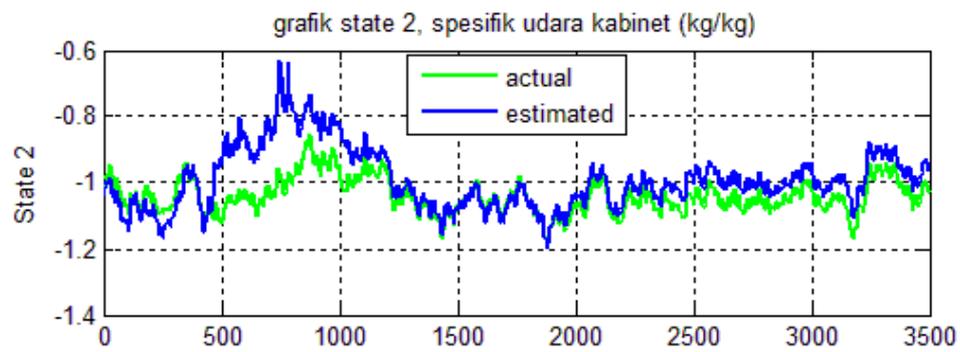
$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_v \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & \cdots & a_v \\ a_2 & & & a_v \\ \vdots & a_v & & \\ a_v & & & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & E_N^{(1)} \end{bmatrix} \begin{bmatrix} D \\ B_T \end{bmatrix} \quad (\text{A.5})$$

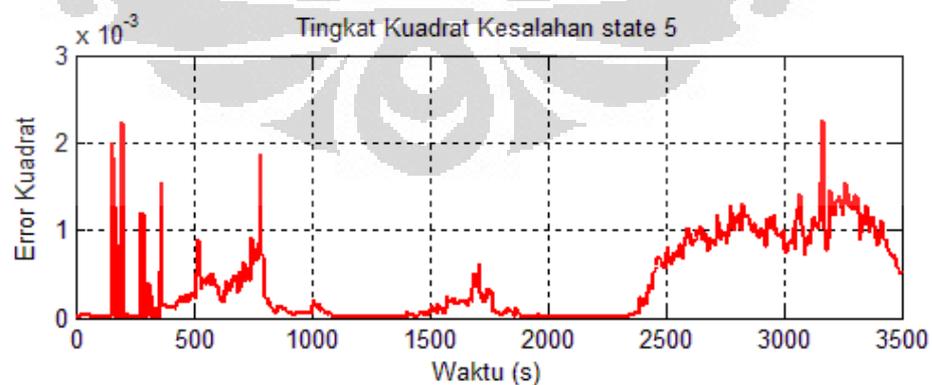
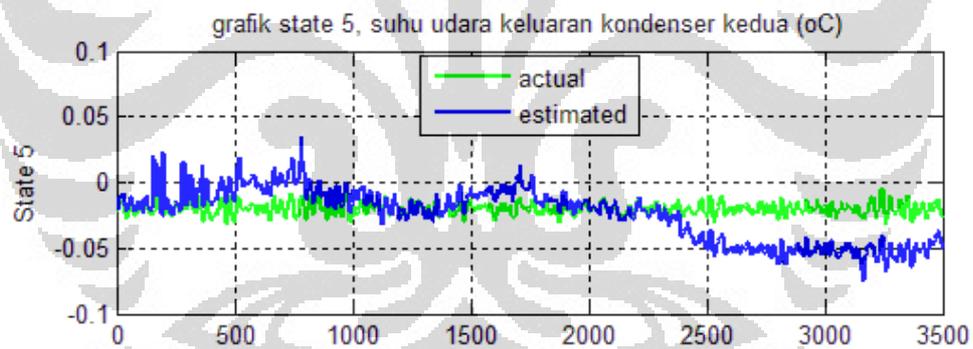
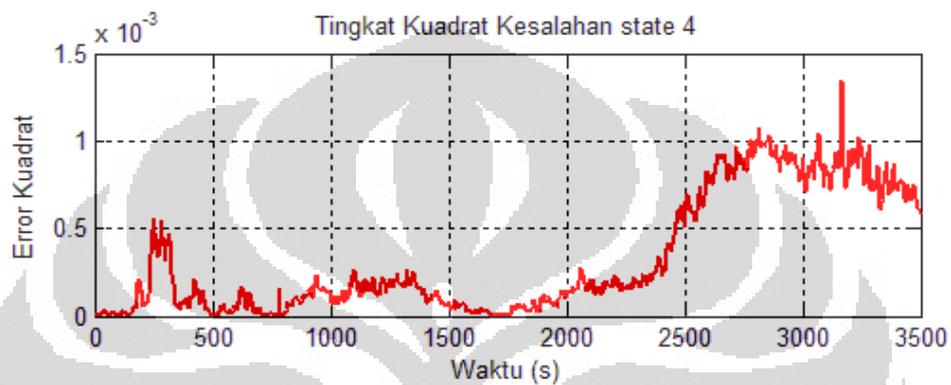
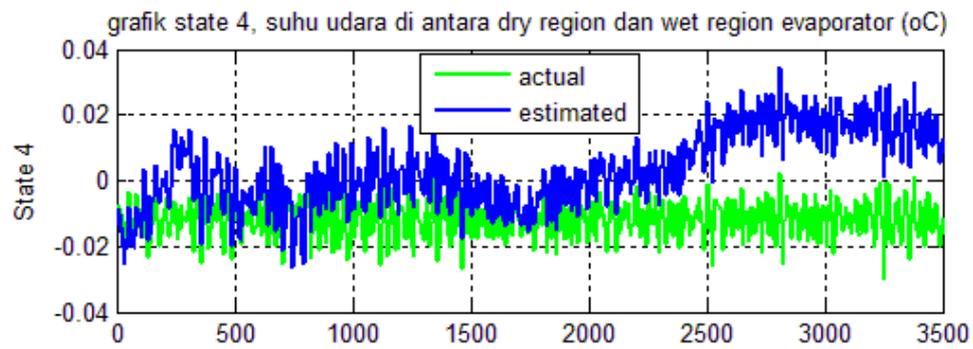
B_T dan D didapat dari persamaan ini.

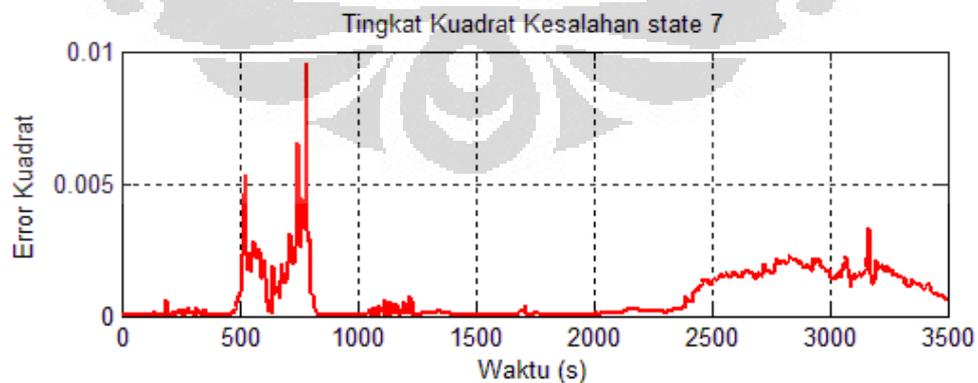
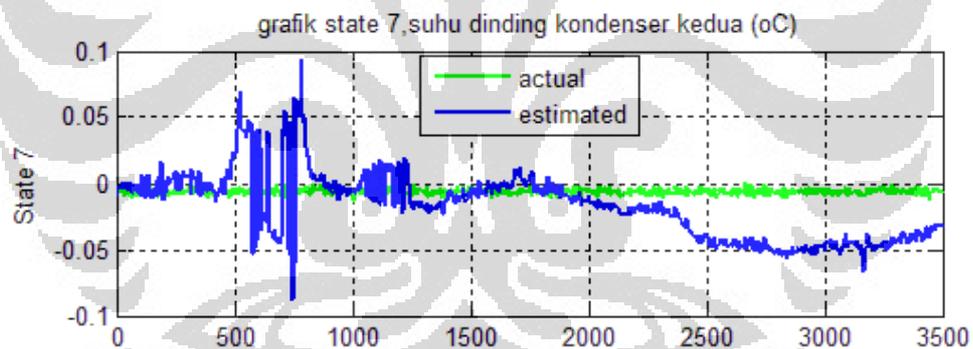
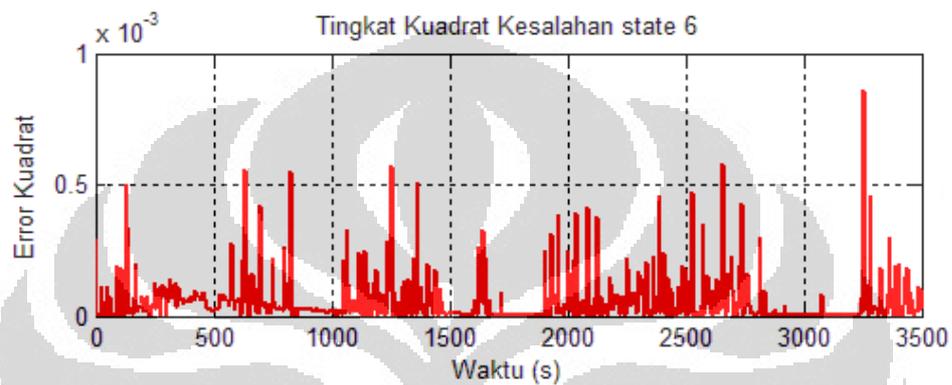
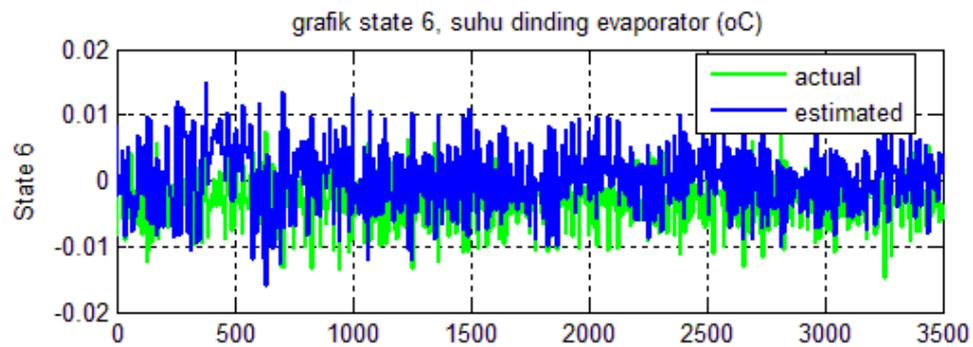
LAMPIRAN B

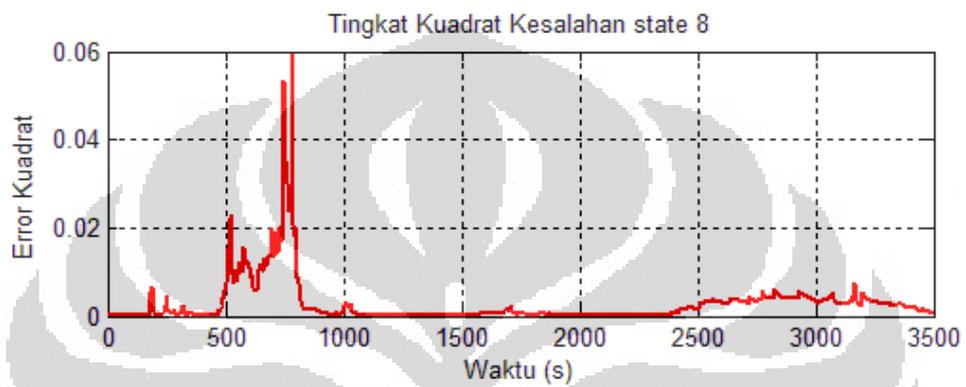
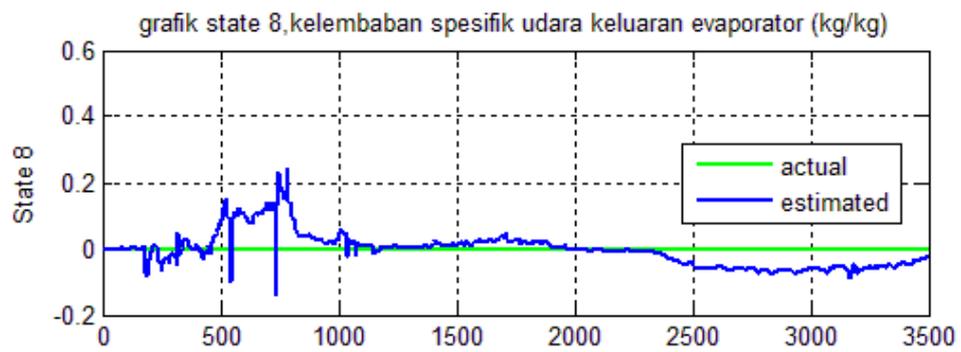
Grafik Perbandingan *state actual* dengan *state* hasil estimasi dan Tingkat Kuadrat Kesalahan dengan *Spectral Density Gaussian Noise 0.0001*





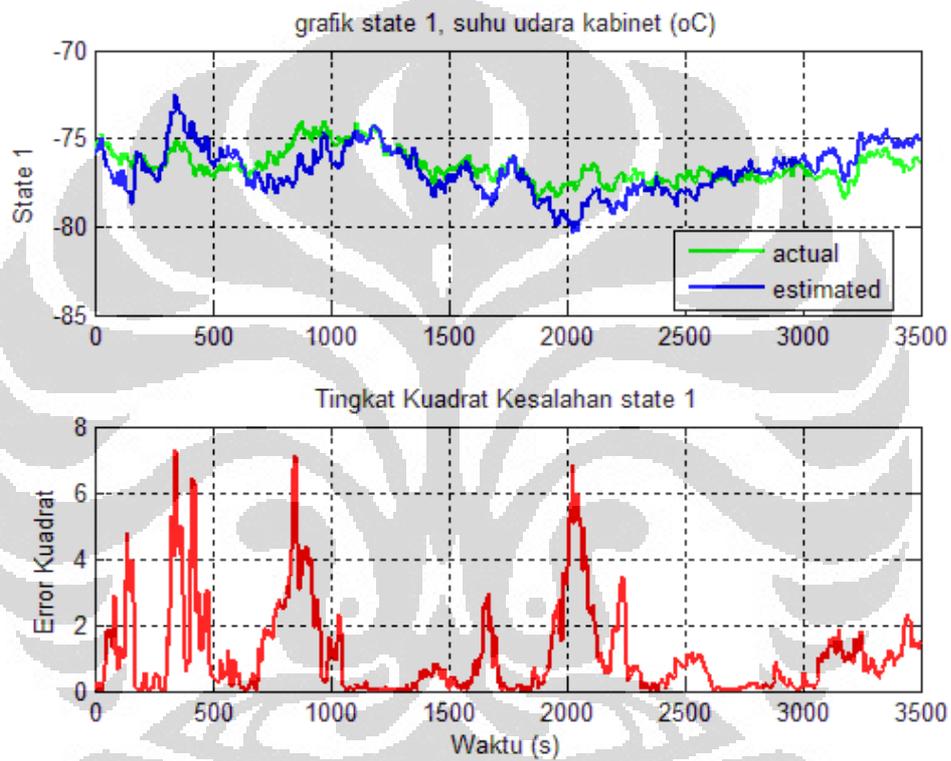


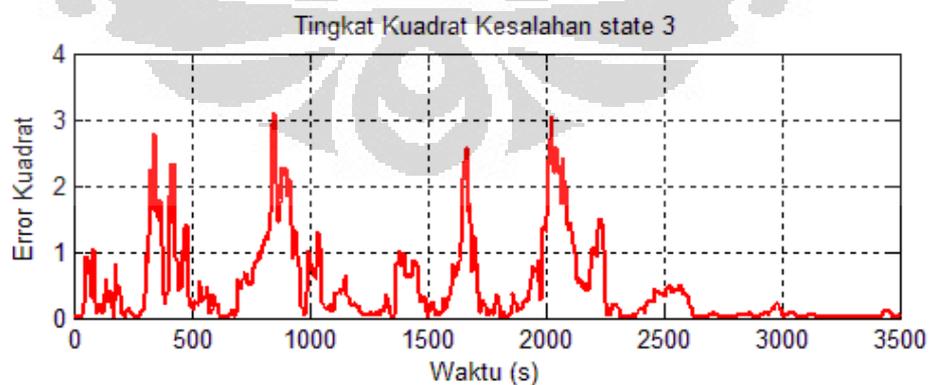
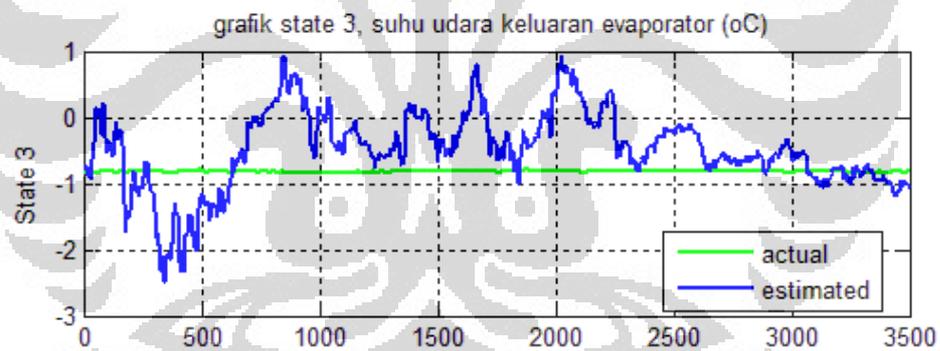
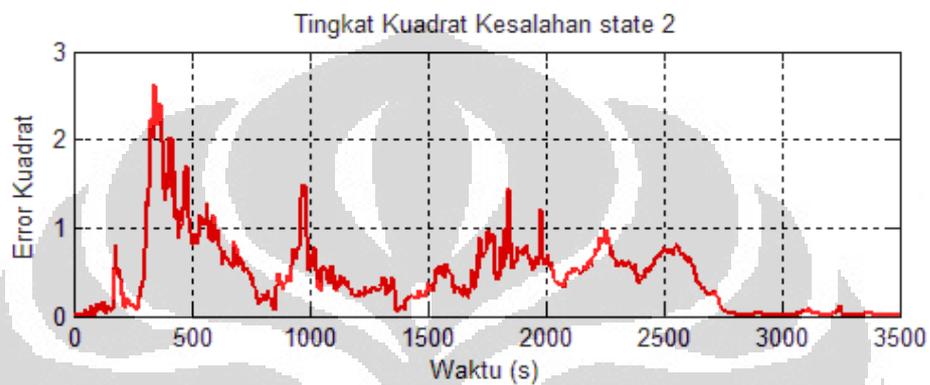
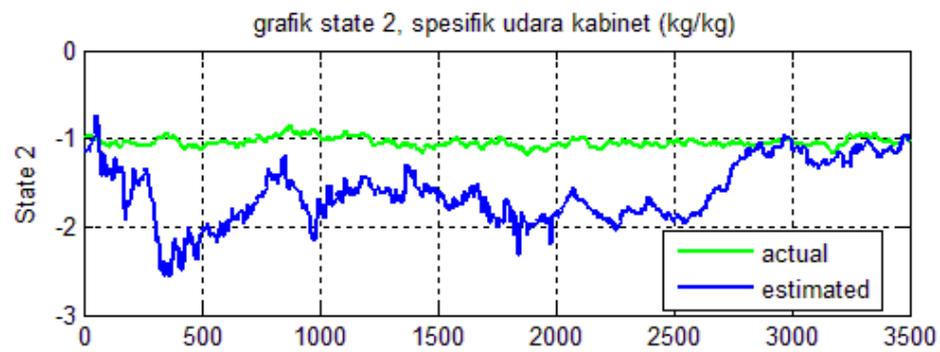


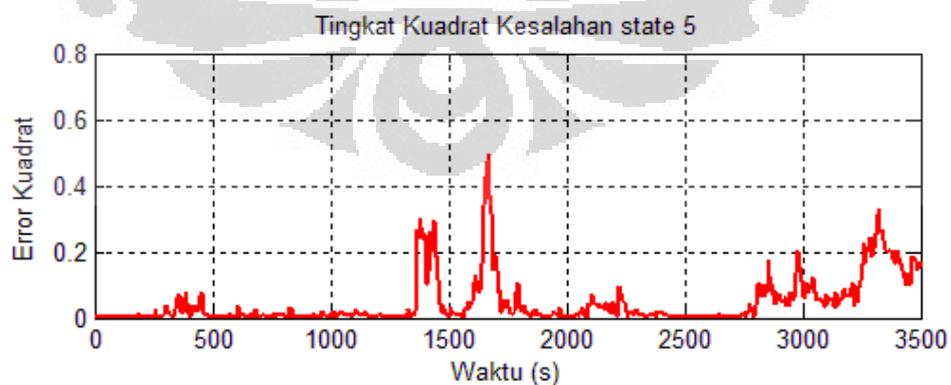
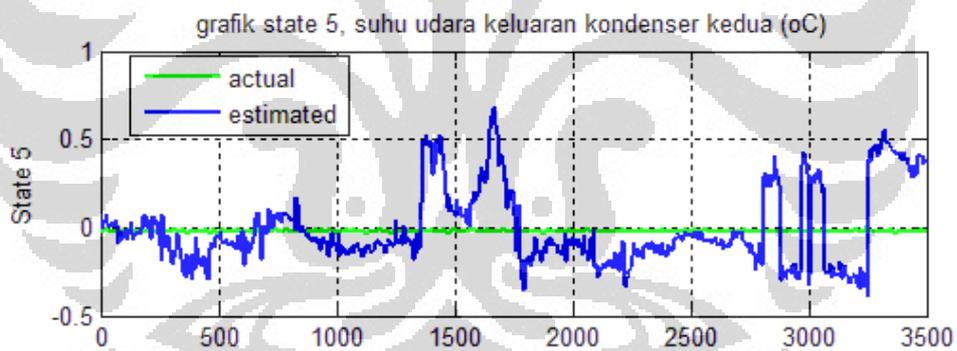
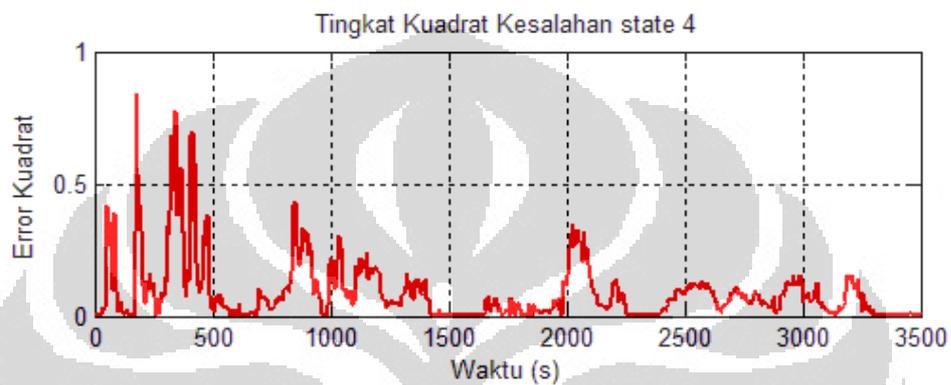
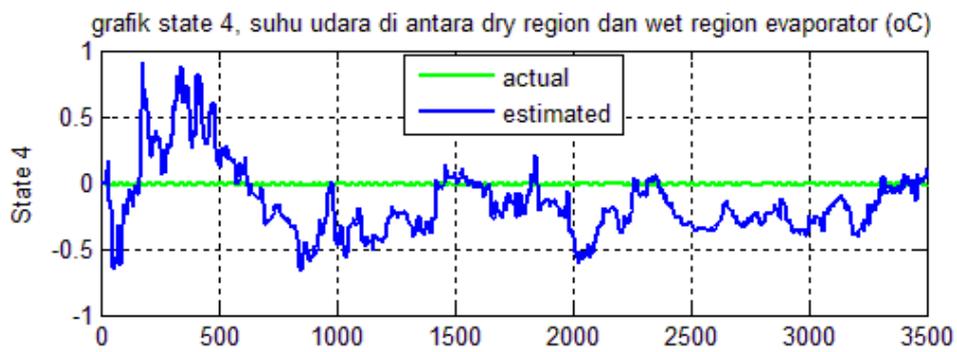


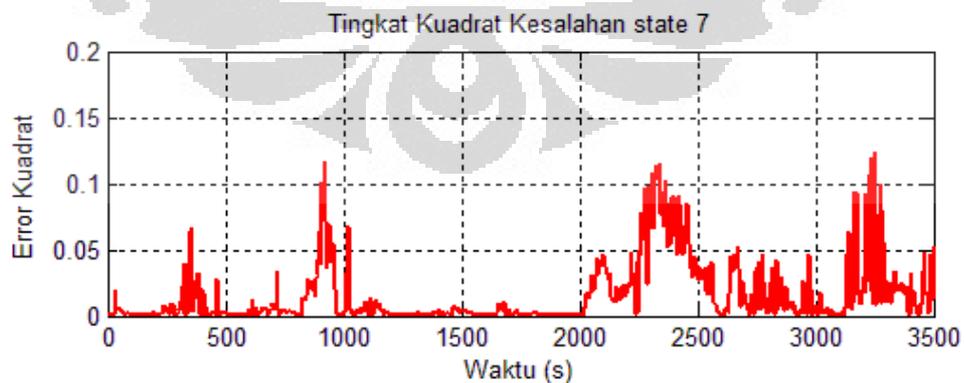
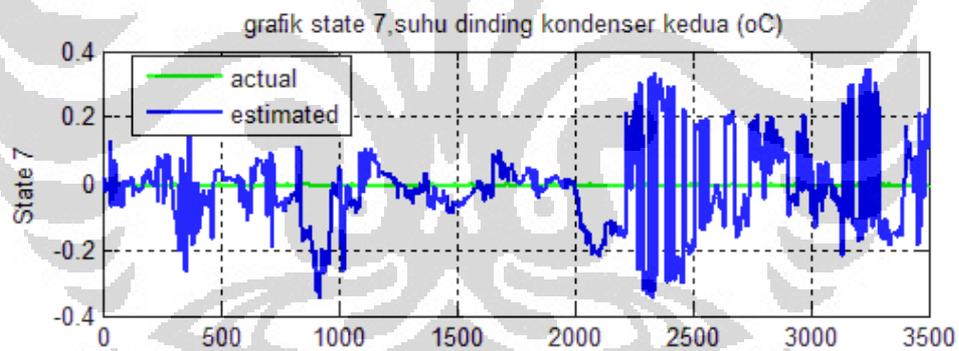
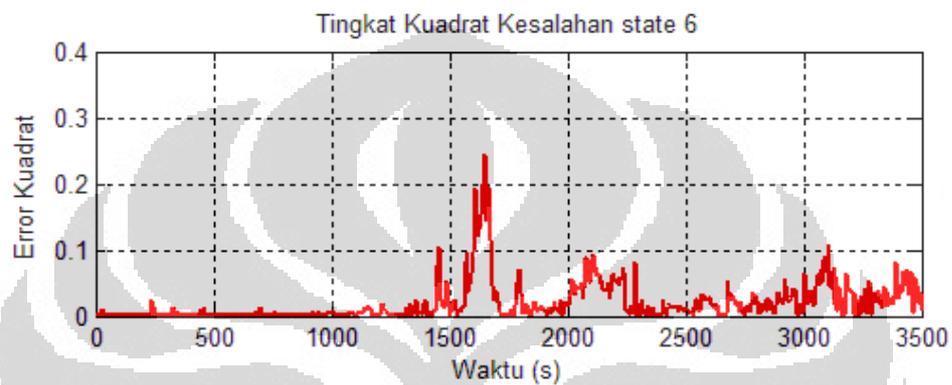
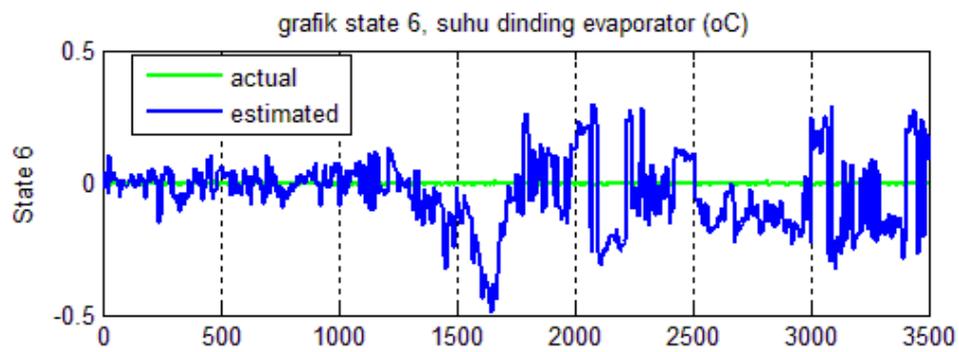
LAMPIRAN C

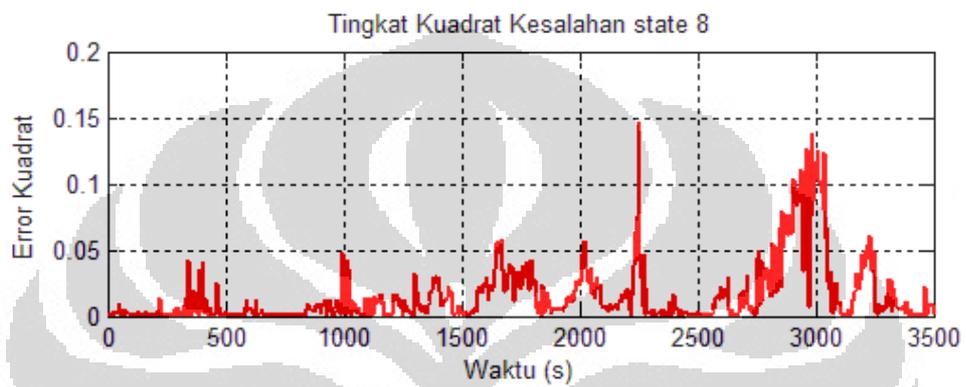
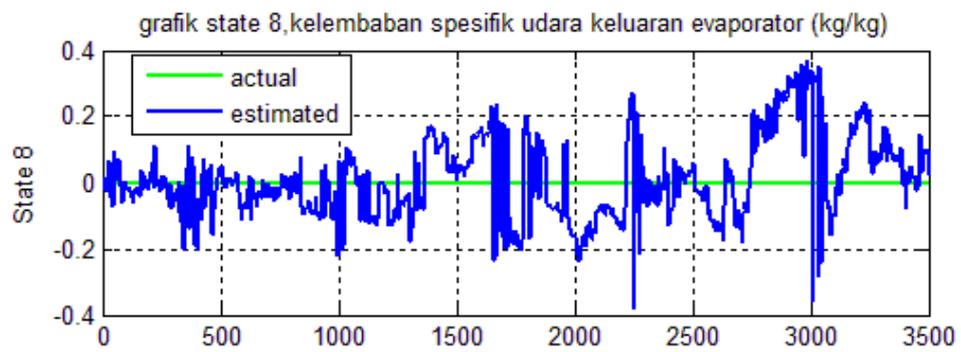
Grafik Perbandingan *state actual* dengan *state* hasil estimasi dan Tingkat Kuadrat Kesalahan dengan *Spectral Density Gaussian Noise 0.01*





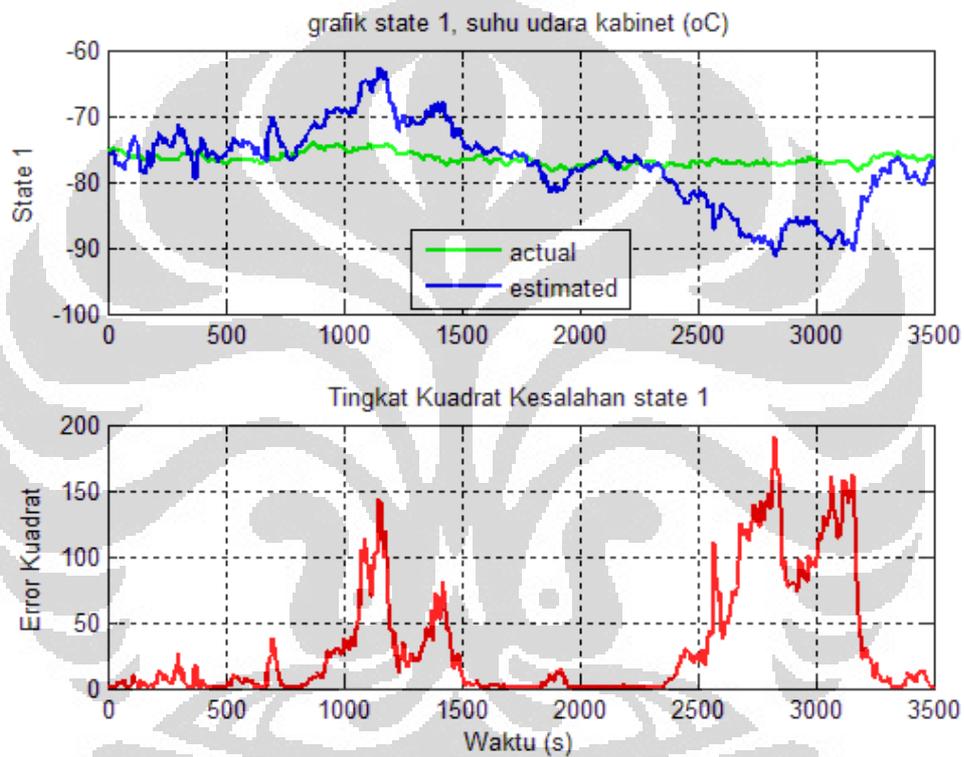


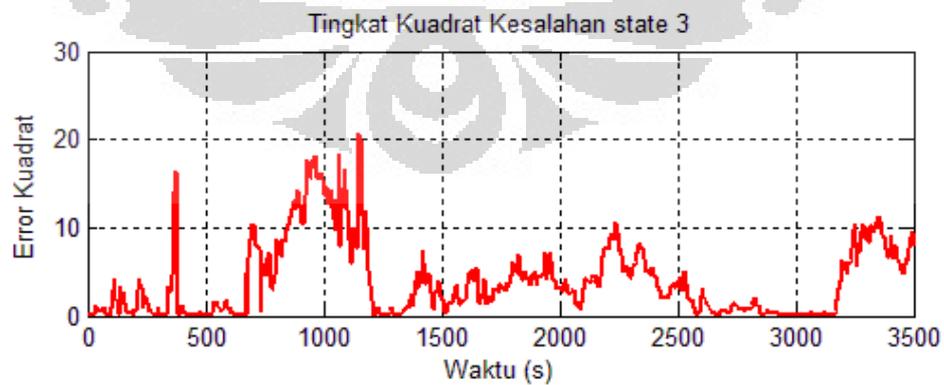
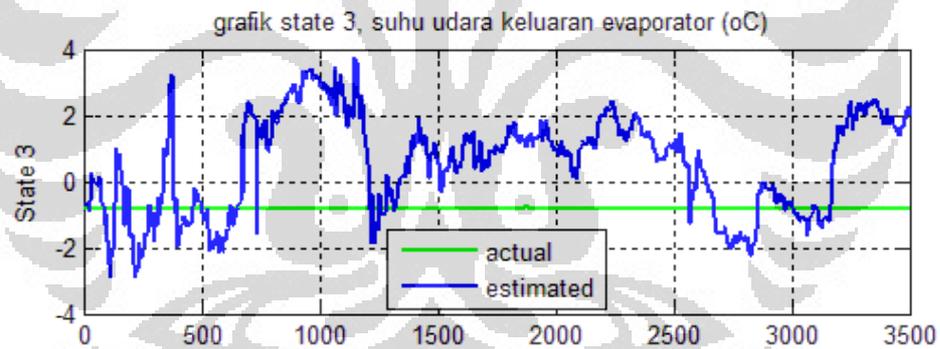
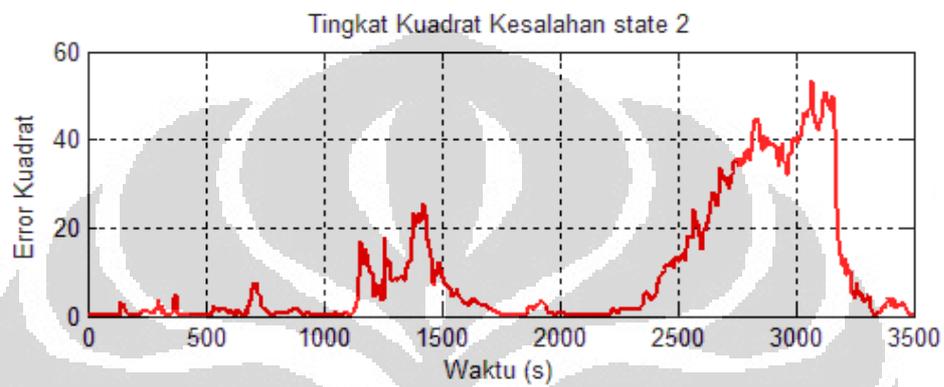


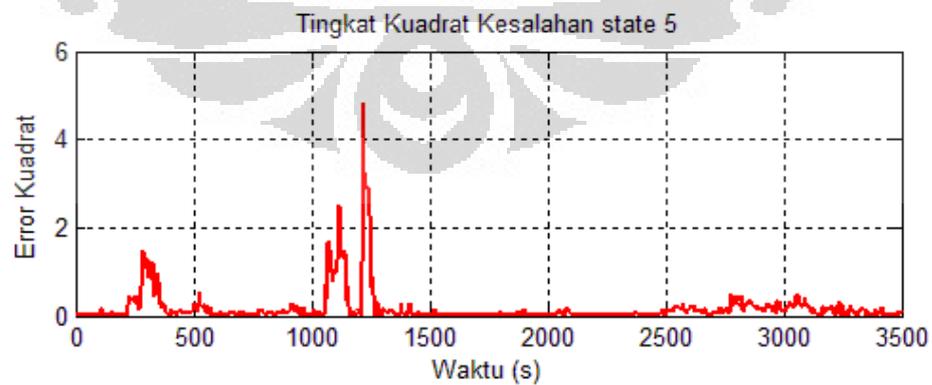
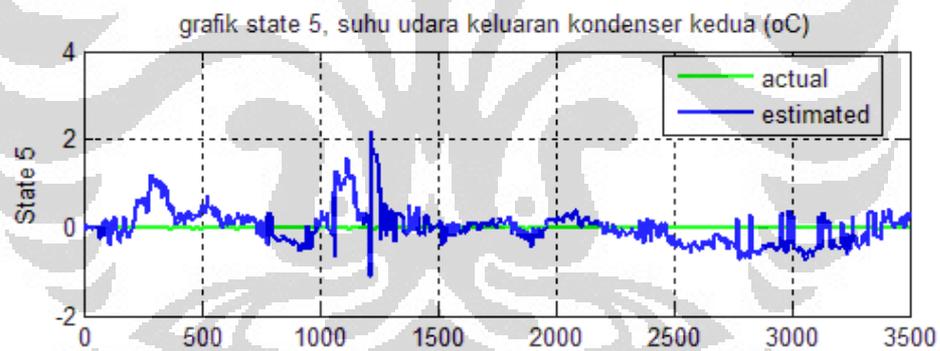
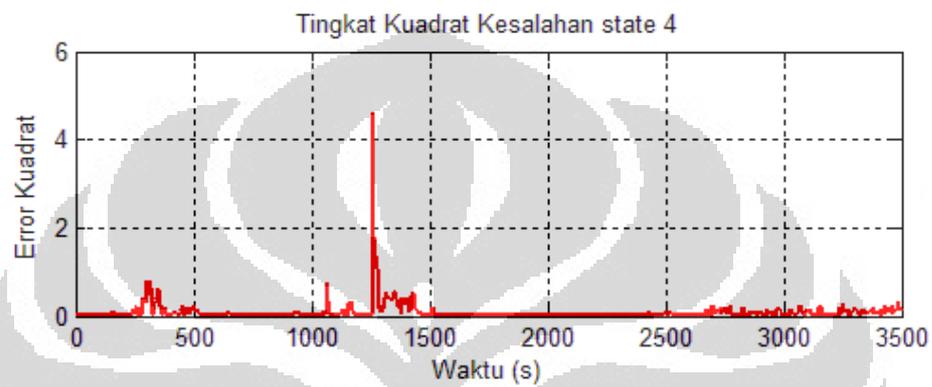
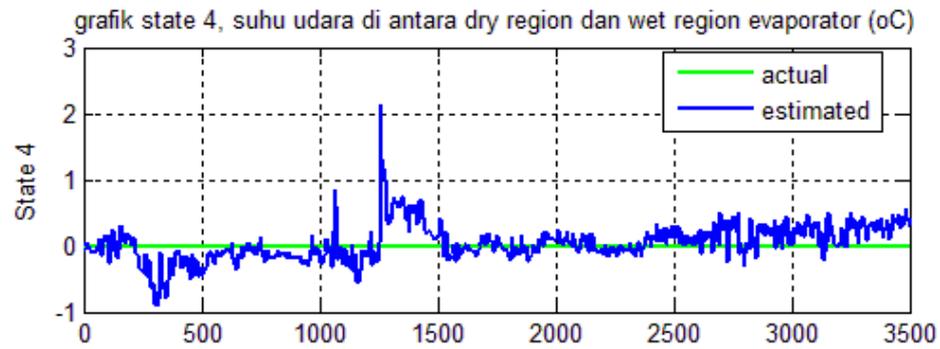


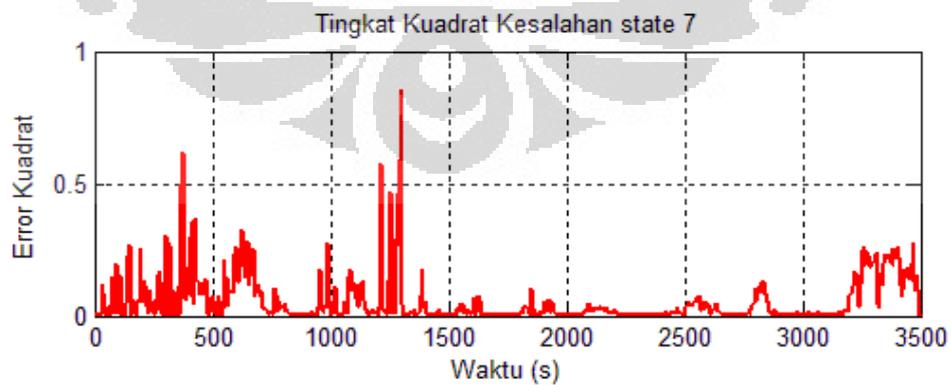
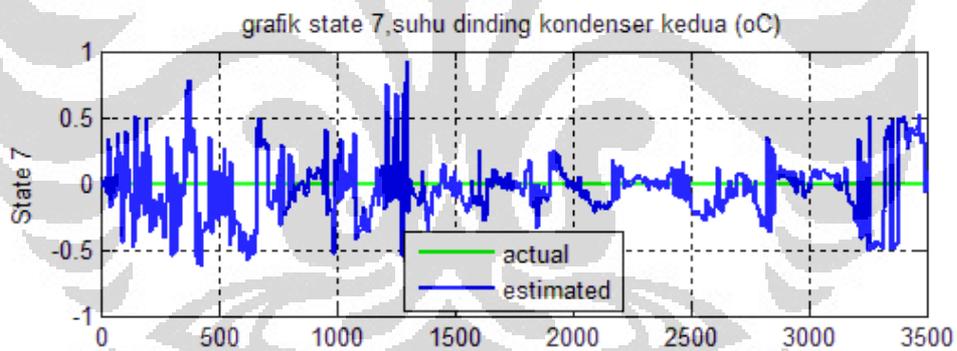
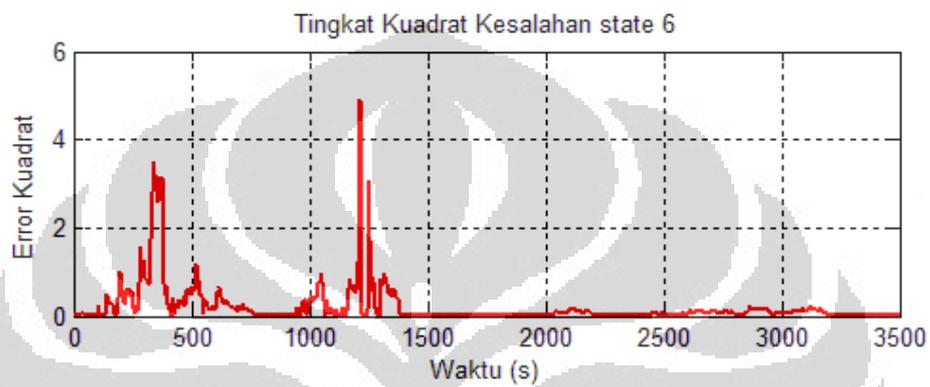
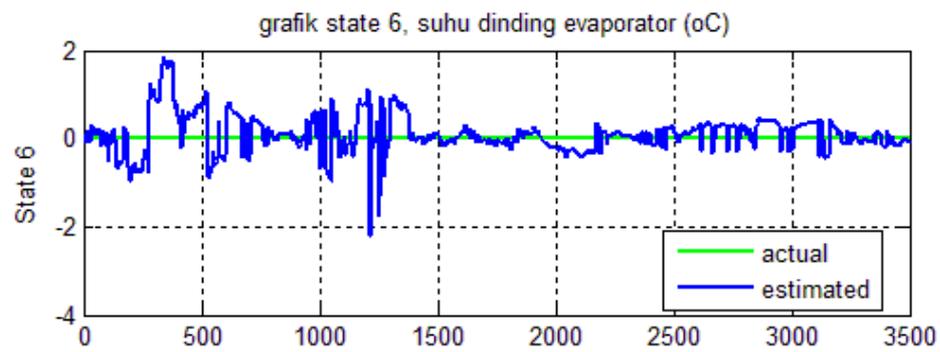
LAMPIRAN D

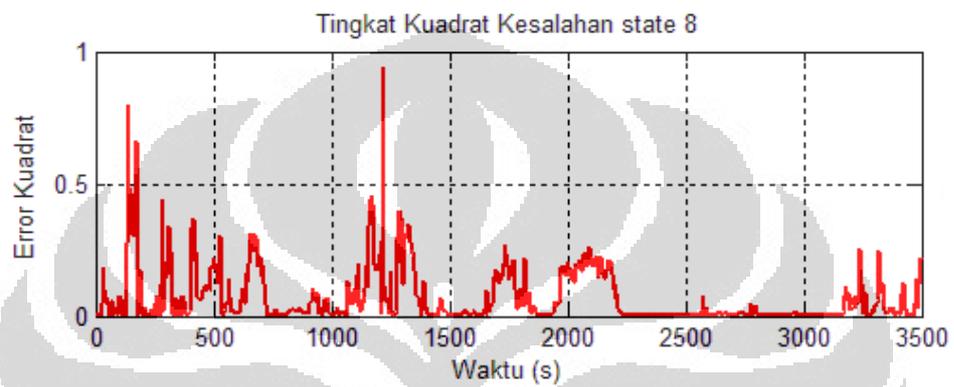
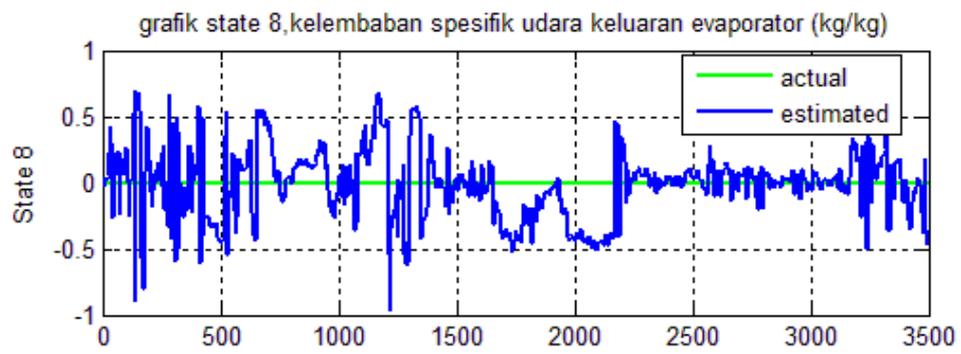
Grafik Perbandingan *state actual* dengan *state* hasil estimasi dan Tingkat Kuadrat Kesalahan dengan *Spectral Density Gaussian Noise 0.1*











LAMPIRAN E

Hasil Estimasi Variabel Keadaan untuk Model nonlinier Sistem Tata Udara Presisi

