



TUGAS AKHIR - TF 181801

**PERANCANGAN SISTEM MONITORING
PREDIKSI KEANDALAN *REAL-TIME* PADA
AKTUATOR HIDROLIK BERBASIS ESTIMASI
KESALAHAN MENGGUNAKAN *OBSERVER***

RIFYAN ALANA
NRP. 0231154000075

Dosen Pembimbing:
Dr. Katherin Indriawati, S.T., M.T.
Moh Kamalul Wafi, S.T., MSc.DIC

DEPARTEMEN TEKNIK FISIKA
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2019



FINAL PROJECT - TF 181801

**REAL-TIME RELIABILITY PREDICTION
SYSTEM DESIGN ON HYDRAULIC ACTUATOR
POSITION CONTROL SYSTEM BASED ON
FAULT ESTIMATION USING OBSERVER**

RIFYAN ALANA
NRP. 02311540000075

Dosen Pembimbing:
Dr. Katherin Indriawati, S.T., M.T.
Moh Kamalul Wafi, S.T., MSc.DIC

ENGINEERING PHYSICS DEPARTMENT
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya 2019

“Halaman ini sengaja dikosongkan”

PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini

Nama : Rifyan Alana
NRP : 02311540000075
Departemen/Prodi : Teknik Fisika/S1 Teknik Fisika
Fakultas : Fakultas Teknologi Industri
Perguruan Tinggi : Institut Teknologi Sepuluh Nopember

Dengan ini menyatakan bahwa Tugas Akhir dengan judul "Perancangan Sistem Monitoring Keandalan *Real-time* Pada Sistem Pengendalian Posisi Aktuator Hidrolik Berbasis Estimasi Kesalahan Menggunakan *Observer*" adalah benar karya saya sendiri dan bukan plagiat dari karya orang lain. Apabila di kemudian hari terbukti terdapat plagiat pada Tugas Akhir ini maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Demikian pernyataan ini saya buat dengan sebenar-benarnya

Surabaya, 22 Juli 2019
Yang membuat pernyataan



Rifyan Alana
NRP.02311540000075

“Halaman ini sengaja dikosongkan”

LEMBAR PENGESAHAN I
TUGAS AKHIR

PERANCANGAN SISTEM MONITORING PREDIKSI
KEANDALAN *REAL-TIME* PADA SISTEM
PENGENDALIAN POSISI AKTUATOR HIDROLIK
BERBASIS ESTIMASI KESALAHAN MENGGUNAKAN
OBSERVER

Oleh:

RIFYAN ALANA
NRP.0231154000075

Surabaya, 22 Juli 2019

Menyetujui,

Dosen Pembimbing I

Dosen Pembimbing II



Dr. Katherin Indriawati, S.T., M.T.
NIPN.19760523 200012 2 001



Moh Kamalul Wafi, S.T., MSc.DIC

Mengetahui,

**Keapala Departemen
Teknik Fisika FTI-ITS**



Agus Muhammad Hatta, S.T., M.Si., Ph.D.

NIPN.19780902 200312 1 002

“Halaman ini sengaja dikosongkan”

**PERANCANGAN SISTEM MONITORING PREDIKSI
KEANDALAN *REAL-TIME* PADA SISTEM
PENGENDALIAN POSISI AKTUATOR HIDROLIK
BERBASIS ESTIMASI KESALAHAN MENGGUNAKAN
*OBSERVER***

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat Memperoleh
Gelar Sarjana Teknik

Pada

Bidang Minat Rekayasa Instrumentasi
Program Studi S-1 Departemen Teknik Fisika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Oleh:

RIFYAN ALANA
NRP.02311540000075

Disetujui oleh Tim Penguji Tugas Akhir:

1. Dr. Katherin Indriawati, S.T.,M.T. (Pembimbing)
2. Moh Kamalul Wafi, S.T., MSc.DIC (Pembimbing)
3. Dr.Bambang L. Widjiantoro, S.T., M.T. (Penguji I)
4. Ir. Ya'umar, M.T. (Penguji II)

SURABAYA
JULI, 2019

“Halaman ini sengaja dikosongkan”

**PERANCANGAN SISTEM MONITORING PREDIKSI
KEANDALAN *REAL-TIME* PADA SISTEM
PENGENDALIAN POSISI AKTUATOR HIDROLIK
BERBASIS ESTIMASI KESALAHAN MENGGUNAKAN
*OBSERVER***

Nama : Rifyan Alana
NRP : 02311540000075
Departemen : Teknik Fisika FTI-ITS
Dosen Pembimbing : Dr. Katherin Indriawati, S.T., M.T.
Moh Kamalul Wafi, S.T., MSc.DIC

ABSTRAK

Kemampuan sistem hidrolik memberikan gaya dan torsi yang besar dengan tingkat akurasi yang tinggi menjadikan sistem hidrolik sebagai sistem penggerak yang sering digunakan pada berbagai aplikasi. Namun karena kondisi kerja dan beban ekstrim yang diterima sistem hidrolik, berbagai kesalahan dan kegagalan dapat terjadi pada sistem. Pada tugas akhir ini dirancang sebuah sistem *monitoring* dan prediksi keandalan *real-time* sistem pengendalian posisi aktuator hidrolik. Langkah utama yang dilakukan dalam perancangan sistem ini adalah perancangan *observer* untuk mengestimasi kesalahan, perancangan algoritma *exponential smoothing* untuk memprediksi kesalahan dan perancangan algoritma *reliability prediction* untuk menghitung keandalan. *Observer* dapat dengan baik mengestimasi nilai kesalahan dengan menggunakan LQR untuk menentukan *gain*-nya. Pada algoritma *exponential smoothing* ditemukan bahwa *time interval* terbaik adalah 1,05s. Pada uji coba algoritma *reliability prediction* ditemukan bahwa keandalan dengan step prediksi 9 step kedepan mampu mengindikasikan akan terjadinya kegagalan pada sistem 4,2 detik lebih cepat dari 5 step kedepan dan 7,35 detik lebih cepat dari 2 step kedepan.

kata kunci – hidrolik, kesalahan, *observer*, *exponential smoothing*, prediksi keandalan

“Halaman ini sengaja dikosongkan”

**REAL-TIME RELIABILITY PREDICTION SYSTEM
DESIGN ON HYDRAULIC ACTUATOR POSITION
CONTROL SYSTEM BASED ON FAULT ESTIMATION
USING OBSERVER**

Name : Rifyan Alana
NRP : 0231154000075
Department : Teknik Fisika FTI-ITS
Supervisors : Dr. Katherin Indriawati, S.T., M.T.
Moh Kamalul Wafi, S.T., MSc.DIC

ABSTRACT

The ability of hydraulic system to exerts great force and torque with a high degree of accuracy making the hydraulic system as a driving tool that is often used in various applications. However, due to the extreme load and working conditions, various errors and failures can occur in the system. In this paper a design of real-time reliability prediction monitoring system has been done to hydraulic actuator control system. Three main steps that are key to the completion were : Designing the observer to provide the failure estimation of the system, designing the exponential smoothing algorithm to predict error's growth and designing the reliability prediction to compute the reliability of the system. The observer could properly estimate the failure's value by using LQR method to find its gain. In the trial exponential smoothing algorithm it was found that the best time interval was 1,05s. In the trial of the reliability prediction algorithm, it was found that the reliability with 9 steps forward yield indication that the system is going to go to a failure earlier than the 5 steps prediction with 4.2 seconds difference and 7.5 seconds difference with 2-steps prediction

keyword – hydraulic, failure, observer, exponential smoothing, reliability prediction.

“Halaman ini sengaja dikosongkan”

KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang senantiasa melimpahkan rahmat dan hidayah-Nya serta shalawat dan salam kepada Nabi Muhammad SAW sehingga dapat terselesaikannya tugas akhir beserta laporan tugas akhir yang berjudul **“Perancangan Sistem Monitoring Keandalan *Real-time* Pada Sistem Pengendalian Posisi Aktuator Hidrolik Berbasis Estimasi Kesalahan Menggunakan *Observer*”**.

Penulis telah banyak memperoleh bantuan dari berbagai pihak dalam penyelesaian tugas akhir dan laporan tugas akhir ini. Penulis mengucapkan terima kasih kepada :

1. Ibu Rini Hermiyati sebagai Ibu saya yang telah memberi dukungan, bimbingan dan pelajaran yang tidak bernilai.
2. Ibu Dr. Katherin Indriawati, S.T., M.T dan Bapak Moh Kamalul Wafi, S.T., MSc.DIC selaku dosen pembimbing yang telah dengan sabar memberikan petunjuk, ilmu serta bimbingan dalam pengerjaan tugas akhir ini.
3. Bapak Syofyan Ahmad sebagai Bapak saya yang telah memberi dukungan moril
4. Yustisa Sofirina dan Rima Novelisia sebagai kakak saya yang telah memberi dukungan materiil dan moril dalam penyelesaian tugas akhir ini

Penulis sadar bahwa penulisan laporan tugas akhir ini masih terdapat kekurangan. Namun, semoga laporan ini dapat memberikan kontribusi yang berarti dan menambah wawasan yang bermanfaat bagi pembaca, keluarga besar Teknik Fisika ITS khususnya, dan civitas akademika ITS pada umumnya. Semoga laporan tugas akhir ini dapat bermanfaat sebagai referensi pengerjaan laporan tugas akhir bagi mahasiswa lainnya.

Surabaya, 21 Juli 2019

Penulis

“Halaman ini sengaja dikosongkan”

DAFTAR ISI

ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR.....	v
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
DAFTAR NOTASI	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Tujuan	3
1.4 Batasan Masalah	4
BAB II TEORI PENUNJANG.....	5
2.1. Sistem Hidrolik	5
2.2. Model Dinamik Aktuator Hidrolik.....	8
2.3. Sensor Posisi <i>Linear Variable Differential</i> <i>Transmitter (LVDT)</i>	13
2.4. Kesalahan Sensor	14
2.5. <i>Observer</i>	15
2.6. Algoritma <i>Exponential Smoothing</i>	17
2.7. Prediksi Keandalan	18
BAB III METODOLOGI PENELITIAN	21
3.1 Studi Literatur	22
3.2 Pemodelan Sistem Pengendalian Posisi Aktuator Hidrolik	22
3.3 Pengujian Kesalahan pada Sistem Pengendalian	25
3.4 Perancangan <i>Observer</i>	26
3.5 Pengujian <i>Observer</i>	32
3.6 Perancangan Algoritma Prediksi Kesalahan	32
3.7 Perancangan Algoritma Prediksi Keandalan <i>Real-</i> <i>time</i>	33
3.8 Pengujian Algoritma Prediksi Keandalan <i>Real-time</i>	34

BAB IV	ANALISIS DATA DAN PEMBAHASAN	37
4.1	Hasil Pemodelan dan Pemberian Kesalahan	37
4.2	Hasil Uji <i>Observer</i>	38
4.3	Hasil Uji Algoritma Prediksi Kesalahan Sensor	39
4.4	Hasil Uji Algoritma Prediksi Keandalan Real-time	51
BAB V	KESIMPULAN DAN SARAN	55
5.1	Kesimpulan	55
5.2	Saran	55
DAFTAR PUSTAKA		
LAMPIRAN		
BIODATA PENULIS		

DAFTAR GAMBAR

Gambar 2. 1 Aktuator Hidrolik Tampak Samping	5
Gambar 2. 2 Diagram Blok Sistem Servo	6
Gambar 2. 3 Kerangka Sistem Aktuator Hidrolik	7
Gambar 2. 4 Konfigurasi <i>Servo Valve</i> (Rydberg, 2014).....	8
Gambar 2. 5 <i>Free Body Diagram</i> Silinder Hidrolik.....	9
Gambar 2. 6 Konstruksi LVDT.....	13
Gambar 2. 7 Sistem dengan <i>Full Order Observer</i>	17
Gambar 3. 1 Diagram Alir Penelitian.....	21
Gambar 3.2 Diagram Blok Sistem Pengendalian Aktuator Hidrolik.....	25
Gambar 3. 3 Diagram blok sistem prediksi keandalan <i>real-time</i>	35
Gambar 4. 1 Respon Sistem Pengendalian.....	37
Gambar 4. 2 Respon Sistem Setelah Diberi Kesalahan <i>Scaling</i>	37
Gambar 4. 3 Grafik hasil estimasi kesalahan sensor	38
Gambar 4. 4 Grafik Perbandingan Hasil Nilai Prediksi dengan <i>Time Interval = 1,05s</i>	43
Gambar 4. 5 Grafik Perbandingan Hasil Nilai Prediksi dengan <i>Time Interval = 1,25s</i>	47
Gambar 4. 6 Grafik Perbandingan Hasil Nilai Prediksi dengan <i>Time Interval = 2s</i>	51
Gambar 4. 7 Posisi Piston x_p mencapai <i>threshold</i>	52
Gambar 4.8 Grafik keandalan hasil uji algoritma prediksi keandalan <i>real-time</i>	52

“Halaman ini sengaja dikosongkan”

DAFTAR TABEL

Tabel 3. 1 Parameter Pemodelan Aktuator Hidrolik	22
Tabel 4. 1 Hasil Prediksi $\widehat{f}(k + 1)$ dengan <i>Time Interval</i> 1,05s	39
Tabel 4.2 Hasil Prediksi $\widehat{f}(k + 4)$ dengan <i>Time Interval</i> 1,05s.	40
Tabel 4. 3 Hasil Prediksi $\widehat{f}(k + 8)$ dengan <i>Time Interval</i> 1,05s	41
Tabel 4. 4 Hasil Prediksi $\widehat{f}(k + 1)$ dengan <i>Time Interval</i> 1,25s	43
Tabel 4. 5 Hasil Prediksi $\widehat{f}(k + 4)$ dengan <i>Time Interval</i> 1,25s	44
Tabel 4. 6 Hasil Prediksi $\widehat{f}(k + 8)$ dengan <i>Time Interval</i> 1,25s	45
Tabel 4. 7 Hasil Prediksi $\widehat{f}(k + 1)$ dengan <i>Time Interval</i> 2s....	47
Tabel 4. 8 Hasil Prediksi $\widehat{f}(k + 4)$ dengan <i>Time Interval</i> 2s...	48
Tabel 4. 9 Hasil Prediksi $\widehat{f}(k + 8)$ dengan <i>Time Interval</i> 2s....	49
Tabel 4. 10 Hasil uji prediksi keandalan <i>real-time</i>	53

“Halaman ini sengaja dikosongkan”

DAFTAR NOTASI

q	=	<i>Aliran cairan melalui spool valve</i>
p	=	<i>Tekanan</i>
u	=	<i>Sinyal kontrol</i>
x_v	=	<i>Posisi spool valve</i>
c	=	<i>Koefisien orifice valve</i>
F	=	<i>Gaya</i>
A	=	<i>Luas</i>
m	=	<i>Massa</i>
K	=	<i>Konstanta pegas</i>
B	=	<i>Modulus bulk cairan</i>
V	=	<i>Volume</i>
x_p	=	<i>Posisi piston</i>
l	=	<i>Step prediksi</i>
T	=	<i>Time interval</i>
A	=	<i>Matrix keadaan</i>
B	=	<i>Matrix kendali</i>
C	=	<i>Matrix keluaran</i>
F_a	=	<i>Matrix kesalahan aktuator</i>
F_s	=	<i>Matrix kesalahan sensor</i>
$G(x)$	=	<i>Matrix bagian non-linear</i>
z	=	<i>Augmented state</i>
\bar{x}	=	<i>Vektor keadaan augmented state</i>
$\hat{\bar{x}}$	=	<i>Vektor keadaan observer dengan augmented state</i>
$\hat{\bar{x}}$	=	<i>Vektor keadaan observer dengan augmented state dan kesalahan</i>
u_n	=	<i>Hasil estimasi yang diperhalus</i>
s_n	=	<i>Faktor pertumbuhan linear</i>
a, b	=	<i>Parameter Holt's method</i>

”Halaman ini sengaja dikosongkan”

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sistem hidrolik banyak digunakan di industri yang membutuhkan objek yang berat atau gaya yang besar. Sebagai contoh, sistem hidrolik digunakan pada sistem tangan robot pemindah barang berat, aktuator dalam sistem pengendalian posisi permukaan pesawat terbang, excavator dan aktuator pengereman pada kendaraan bermotor (An & Sepehri, 2003) (Khan dkk, 2005). Meskipun sistem hidrolik dapat memberikan gaya dan torsi yang besar dengan akurasi yang tinggi, kondisi ekstrim dan kerumitan cara kerja sebuah sistem hidrolik menimbulkan banyaknya kerusakan dan kegagalan yang dapat terjadi pada sebuah sistem hidrolik beberapa diantaranya adalah dapat terjadinya kebocoran (*leakage*) cairan hidrolik pada komponen, komponen yang aus (*wear*), perubahan suplai tekanan hidrolik (Sharifi dkk, 2017) dan juga kesalahan-kesalahan yang terjadi pada sistem pengendalian sistem hidrolik yaitu pada sensor yang mengalami perubahan pada karakteristik statiknya (Isermann, 2006). Kegagalan dan kesalahan yang dapat terjadi pada sistem hidrolik ini dapat menurunkan umur berfungsinya (*useful lifetime*), sehingga sebuah sistem yang dapat memonitoring tingkat “kesehatan” sistem hidrolik dibutuhkan untuk alasan kualitas performa sistem dan keamanan. Oleh karena itu sebuah sistem *monitoring* keandalan secara *real time* beserta prediksi keandalan untuk menjejaki penurunan keandalan sistem agar kegagalan sistem dapat dideteksi lebih dini akan sangat bermanfaat untuk aktuator hidrolik dan keamanan sistem yang menggunakan aktuator hidrolik untuk bekerja.

Evaluasi dan prediksi keandalan secara *real-time* secara kasar dapat dibagi menjadi dua yaitu analisis regresi dan analisis deret waktu. Metode analisis regresi sering mengadopsi fungsi jalur

degradasi yang dapat menggambarkan variasi kasar yang sesuai dengan variabel performansi selama siklus sistem beroperasi secara keseluruhan dan pengukuran online pada variabel performansi digunakan untuk mengidentifikasi fungsi jalur degradasi (Xu, Ji, & Zhou, 2009). Metode analisis deret waktu mengambil pengukuran *online* pada variabel performansi sebagai deret waktu, dan analisis deret waktu dengan beberapa pendekatan statistik atau cerdas.

Pada (Lu, Kolarik, & Lu, 2001) dijelaskan teknik prediksi keandalan dengan analisa deret waktu pada sebuah unit pada kondisi operasi tertentu. Pada penelitian ini ditentukan nilai batas dimana sistem dapat mengalami kegagalan. Pada penelitian sebelumnya, (Xu, Ji, & Zhou, 2009) melakukan perhitungan keandalan pada sebuah sistem 3 tank air dengan komponen yang terdegradasi. Evaluasi keandalan dilakukan dengan analisa deret waktu, dimana secara keseluruhan terdapat 3 algoritma yang digunakan yaitu estimasi *state*, estimasi dan prediksi kesalahan dan prediksi keandalan. Nilai keandalan didapatkan dengan metode Monte Carlo dan Estimasi kesalahan dilakukan dengan algoritma OTK-SIR. Sama halnya dengan (Lu, Kolarik, & Lu, 2001) disini juga digunakan algoritma *exponential smoothing*.

Dalam (Ayurani, 2019), dilakukan perancangan prediksi keandalan sistem pengendalian kecepatan generator turbin angin. Kesalahan diberikan pada sensor berupa kesalahan *bias*. Algoritma *exponential smoothing* juga digunakan untuk memprediksi kesalahan, sementara estimasi kesalahan didapatkan dengan menggunakan *observer*. Namun sistem prediksi keandalan yang dirancang (Ayurani, 2019) ditujukan untuk keandalan seluruh sistem pengendalian. Dibutuhkan sistem prediksi keandalan yang dilakukan hanya untuk sebuah unit sehingga didapatkan pengertian yang lebih mendalam tentang pengaruh kerusakan atau deteriorasi pada sebuah parameter dalam unit tersebut.

Berdasarkan penelitian sebelumnya maka diusulkan perancangan prediksi keandalan pada sistem pengendalian posisi aktuator hidrolik dimana terdapat kesalahan sensor posisi pada sistem pengendalian tersebut. Berbeda dengan (Xu, Ji, & Zhou, 2009), estimasi dan prediksi kesalahan akan menggunakan *State Observer* berdasarkan (Indriawati dkk, 2015).

1.2 Perumusan Masalah

Dari latar belakang yang telah diuraikan diatas, maka didapatkan permasalahan dalam tugas akhir ini adalah sebagai berikut:

- Bagaimana cara merancang *observer* yang dapat digunakan untuk mendapatkan estimasi kesalahan sensor pada sistem pengendalian posisi aktuator hidrolik?
- Bagaimana cara memprediksi kesalahan pada sistem pengendalian posisi aktuator hidrolik menggunakan algoritma *Exponential Smoothing* ?
- Bagaimana cara menghitung keandalan sistem pengendalian posisi aktuator hidrolik menggunakan algoritma *Reliability Prediction* (RP) ?

1.3 Tujuan

Berdasarkan rumusan masalah tersebut, maka tujuan pada tugas akhir ini adalah sebagai berikut:

- Merancang *observer* pada sistem pengendalian posisi actuator hidrolik yang mampu mengestimasi kesalahan
- Memprediksi nilai kesalahan pada sistem pengendalian posisi actuator hidrolik menggunakan algoritma *Exponential Smoothing*.
- Menghitung keandalan sistem pengendalian posisi actuator hidrolik secara *real-time* menggunakan algoritma *Reliability Prediction* (RP).

1.4 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

- *Valve* yang digunakan berupa *valve* direksional 4/3
- Kesalahan yang terjadi pada sistem berupa *loss of sensitivity* atau kesalahan *scaling/gain* pada sensor posisi
- Keandalan yang dimaksud dalam penelitian ini adalah nilai keandalan oleh algoritma RP, bukan keandalan yang diperoleh dari data maintenance
- Simulasi dilakukan pada Matlab

BAB II TEORI PENUNJANG

2.1. Sistem Hidrolik

Sistem hidrolik menggunakan sistem yang menggunakan cairan hidrolik yang *incompressible* untuk mentransmisikan energy dari suatu sumber ke tujuan. Penggunaan sistem hidrolik dalam permesinan seperti pesawat terbang dan kendaraan beroda dikarenakan sistem hidrolik mampu memberikan tingkat akurasi dan fleksibilitas yang baik, energy besar yang dihasilkan dengan ukuran yang kecil, kecepatan performansinya dan presisi yang baik dalam aplikasinya.

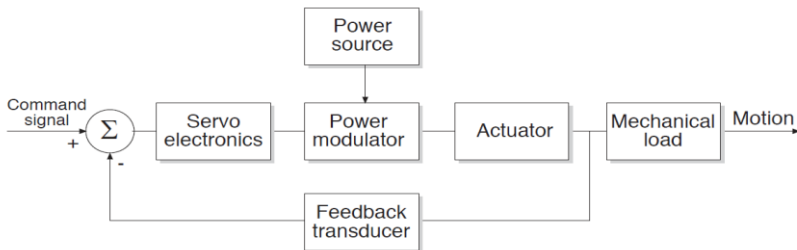


Gambar 2. 1 Aktuator Hidrolik Tampak Samping (Xu dkk, 2016)

Sistem hidrolik memiliki beragam rangkaian hidrolik yang dapat memberikan variasi gerak dan gaya. Namun secara dasar seluruh rangkaian hidrolik pada sistem terdiri dari 4 komponen yaitu : *reservoir* untuk menampung cairan hidrolik, pompa yang digunakan untuk memasok cairan melalui rangkaian/pipa, *valve* yang mengatur besar tekanan dan besar aliran cairan, dan yang terakhir aktuator yang mengubah energy hidrolik menjadi energy mekanik. Gambar 2.1 menunjukkan sebuah aktuator hidrolik dari tampak samping.

2.1.1 Sistem Servo

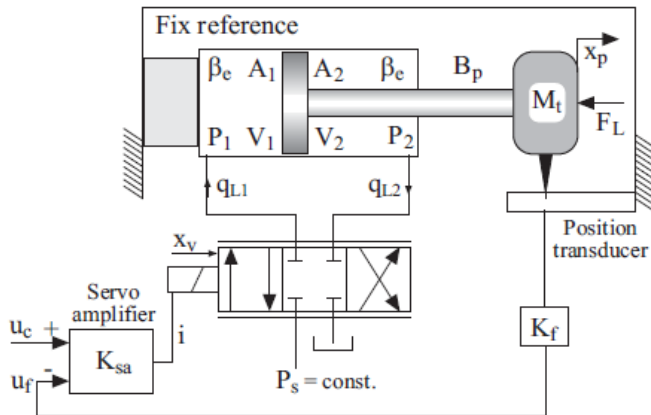
Sistem servo atau *servomechanism* adalah sebuah sistem pengendalian, yang mana melakukan pengukuran dari outputnya sendiri dan mengatur nilai output agar mencapai sinyal perintah. Sehingga pengaruh dari anomal pada divais pengendali dan beban sistem dapat dikecilkan, begitu juga dengan gangguan eksternal. Sebuah sistem servo dapat didesain untuk mengendalikan besaran fisik seperti gerak, gaya, tekanan, temperature dan tegangan listrik Gambar 2.2 menunjukkan diagram blok sebuah sistem servo yang digunakan untuk menggerakkan beban mekanik (Rydberg, 2014).



Gambar 2. 2 Diagram Blok Sistem Servo (Rydberg, 2014)

2.1.2 Aktuator Hidrolik

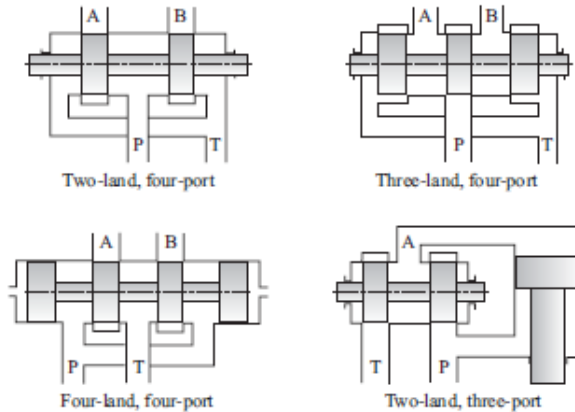
Aktuator hidrolik merupakan sebuah sistem servo yaitu silinder servo posisi terkendali oleh *control valve*, seperti yang ditunjukkan pada Gambar 2.3. *Valve* mengalirkan cairan hidrolik bertekanan tinggi ke silinder daya yang memiliki piston. Karena cairan hidrolik yang *incompressible*, gaya hidrolik yang besar



Gambar 2. 3 Kerangka Sistem Aktuator Hidrolik (Rydberg, 2014)

dapat tercipta untuk menggerakkan beban (Ogata, System Dynamics, 2004). *Control valve* yang sering digunakan untuk aktuator hidrolik merupakan *sliding valve* yang memiliki *spool* (kumparan) atau lebih sering disebut *spool valve*. *Spool valve* dapat dikelompokkan berdasarkan jumlah jalur aliran masuk dan keluar pada *valve (port)* dan jumlah *spool (land)*. Karena seluruh *valve* membutuhkan setidaknya satu aliran suplai, satu aliran kembali dan satu aliran ke aktuator/beban, *valve* setidaknya adalah tipe 3 atau 4 *port*.

Pada Gambar 2.4, *valve* dengan 4 *port* memiliki 2, 3 dan 4 *land*. Untuk menggerakkan *spool* utama, 2 atau 3 *land* dapat digunakan. Dengan 4 atau 3 *land* pada *valve* dimungkinkan untuk menggunakan tekanan sebagai penggerak *valve*. Hal ini merupakan dasar dari *servo valve* dengan respon yang cepat (Rydberg, 2014). Namun dalam tugas akhir ini *spool valve* hanya digerakkan oleh sinyal elektrik yang dihasilkan oleh kontroler



Gambar 2. 4 Konfigurasi *Servo Valve* (Rydberg, 2014)

2.2. Model Dinamik Aktuator Hidrolik

2.2.1 Model Dinamik *Spool Valve*

Spool Valve atau juga sering disebut *servo valve* memiliki peran penting dalam sebuah sistem aktuator hidrolik. *Spool valve* adalah komponen yang berlaku sebagai penghubung antara sinyal input elektrik (atau mekanik) dan besarnya nilai aliran cairan hidrolik. *Spool valve* dapat dimodelkan sebagai sistem dinamik orde pertama dengan *time constant* τ .

$$\tau \dot{x}_v + x_v = k_v u \quad (2.1)$$

τ = *Time Constant*

x_v = Posisi *spool valve*

u = Sinyal kontrol

k_v = Gain *spool valve*

Jika dilakukan transformasi Laplace pada (2.1) maka akan didapatkan transfer fungsi dinamik orde satu berupa (2.2)

$$G_v(s) = \frac{x_v(s)}{u(s)} = \frac{k_v}{\tau s + 1} \quad (2.2)$$

Dengan mengabaikan terjadinya kebocoran pada *valve*, ditemukan hubungan nilai posisi *spool valve* dengan besar aliran cairan rem yang dialirkan melalui *spool valve* yaitu :

$$q_1 = \begin{cases} c_1 x_v \sqrt{p_s - p_1} & x_v \geq 0 \\ c_2 x_v \sqrt{p_1 - p_r} & x_v < 0 \end{cases} \quad (2.3)$$

$$q_2 = \begin{cases} -c_3 u \sqrt{p_2 - p_r} & x_v \geq 0 \\ -c_4 u \sqrt{p_s - p_2} & x_v < 0 \end{cases} \quad (2.4)$$

q_1 = Aliran cairan silinder utama

q_2 = Aliran cairan ruangan pendorong

p_s = Tekanan sumber

p_r = Tekanan reservoir

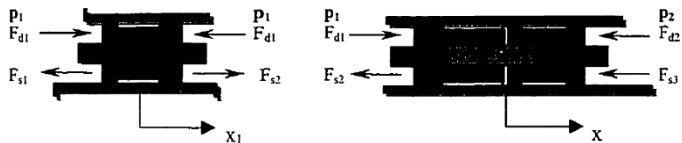
p_1 = Tekanan pada silinder utama

p_2 = Tekanan pada ruangan pendorong

c_1, c_2, c_3, c_4 = koefisien *orifice valve*

2.2.2 Gaya pada Silinder Hidrolik

Silinder hidrolik dapat dibagi menjadi dua rangkaian berbeda seperti pada Gambar 2.5



Gambar 2.5 Free Body Diagram Silinder Hidrolik (Fu, 2000)

Dengan mengabaikan inersia kecil dari piston, tekanan pada silinder utama dan ruang pendorong dapat ditemukan dengan persamaan (2.5) dan (2.6).

$$p_1 = F_{d1}/A_1 \quad (2.5)$$

$$p_2 = F_{d2}/A_2 \quad (2.6)$$

F_{d1} = Gaya hidrolik pada silinder utama

F_{d2} = Gaya hidrolik pada ruang pendorong

A_1 = Luas piston pada silinder utama

A_2 = Luas piston pada ruang pendorong

Menggunakan hukum Newton II, persamaan gerak dari silinder dapat ditemukan

$$F_{d1} - F_{d2} - F_{s2} - F_{s3} = m\ddot{x}_p \quad (2.7)$$

$$F_{s2} - F_{s1} = m_m\ddot{x}_{p1} \quad (2.8)$$

F_{s1} = Gaya pegas pada ruang pertama silinder utama

F_{s2} = Gaya pegas pada ruang kedua silinder utama

F_{s3} = Gaya pegas pada ruang pendorong

m = Massa batang dan piston pada ruang pendorong

m_m = Massa piston pada silinder utama

x_{p1} = Posisi piston pada silinder utama

x_p = Posisi piston pada ruang pendorong

k = Konstanta pegas

Gaya balik masing-masing pegas adalah :

$$F_{s1} = kx_{p1} \quad (2.9)$$

$$F_{s2} = k(x_p - x_{p1}) \quad (2.10)$$

$$F_{s3} = kx_p \quad (2.11)$$

Efek inersia gerak dari piston pada silinder utama cukup kecil dibandingkan dengan gaya-gaya yang berlaku pada silinder utama. Dengan mengabaikan gaya inersia ini, didapatkan :

$$F_{s2} - F_{s1} = 0 \quad (2.12)$$

Dengan substitusi persamaan (2.10) ke (2.9), didapatkan

$$x_{p1} = \frac{1}{2}x_p \quad (2.13)$$

Sehingga gaya balik pegas menjadi

$$F_{s1} = F_{s2} = \frac{1}{2}kx_p \quad (2.14)$$

$$F_{s3} = kx_p \quad (2.15)$$

2.2.3 Model Dinamik Cairan Hidrolik

Ketika aktuasi hidrolik terjadi, cairan hidrolik akan disuplai ke ruang pendorong, menyebabkan meningkatnya tekanan dan mendorong diafragma maju (Sohl & Bobrow, 1999). Menggunakan *modulus bulk*, didapat model dinamik cairan hidrolik pada silinder utama dan ruangan pendorong yang berupa :

$$\dot{p}_1 = \frac{\beta}{V_{11}}(-\dot{V}_{11} + q_{11}) = \frac{\beta}{V_{12}}(-\dot{V}_{12} + q_{12}) \quad (2.16)$$

$$\dot{p}_2 = \frac{\beta}{V_2}(-\dot{V}_2 + q_2)$$

$$V_{11} = \frac{1}{2}V_{10} + x_{p1}A_1$$

$$V_{12} = \frac{1}{2}V_{10} + (x_p - x_{p1})A_1 \quad (2.17)$$

$$V_2 = V_{20} - x_pA_2$$

β = Modulus *bulk* cairan

V_{11} = Volume total cairan pada ruangan pertama silinder utama

V_{12} = Volume total cairan pada ruangan kedua silinder utama

V_1 = Volume total cairan silinder utama

V_2 = Volume total cairan ruangan pendorong

V_{10} = Volume cairan awal pada silinder utama

V_{20} = Volume cairan awal pada silinder utama

q_{11} = Aliran cairan menuju ruangan pertama silinder utama

q_{12} = Aliran cairan menuju ruangan kedua silinder utama

Dengan mensubstitusi persamaan (2.14) pada (2.17) didapatkan

$$\dot{p}_1 = \frac{\beta}{V_1} (-\dot{x}_p A_1 + q_1) \quad (2.18)$$

$$\dot{p}_2 = \frac{\beta}{V_2} (\dot{x}_p A_2 + q_2)$$

$$V_{11} = V_{12} = \frac{1}{2} V_1 = \frac{1}{2} V_{10} + x_p A_1 \quad (2.19)$$

$$V_2 = V_{20} - x_p A_2$$

2.2.4 Model Dinamik Gerak Piston

Dengan mensubstitusi persamaan (2.3) dan (2.4) ke persamaan (2.18) persamaan dinamik cairan hidrolik menjadi

$$\dot{p}_1 = \begin{cases} \frac{\beta}{V_1} (-A_1 \dot{x}_p + c_1 x_v \sqrt{p_s - p_1}) & x_v \geq 0 \\ \frac{\beta}{V_1} (-A_1 \dot{x}_p + c_2 x_v \sqrt{p_1 - p_r}) & x_v < 0 \end{cases} \quad (2.20)$$

$$\dot{p}_2 = \begin{cases} \frac{\beta}{V_2} (A_2 \dot{x}_p - c_3 x_v \sqrt{p_2 - p_r}) & x_v \geq 0 \\ \frac{\beta}{V_2} (A_2 \dot{x}_p - c_4 x_v \sqrt{p_s - p_2}) & x_v < 0 \end{cases}$$

Gaya cairan dan gaya pegas yang berlaku pada piston di ruang pendorong adalah

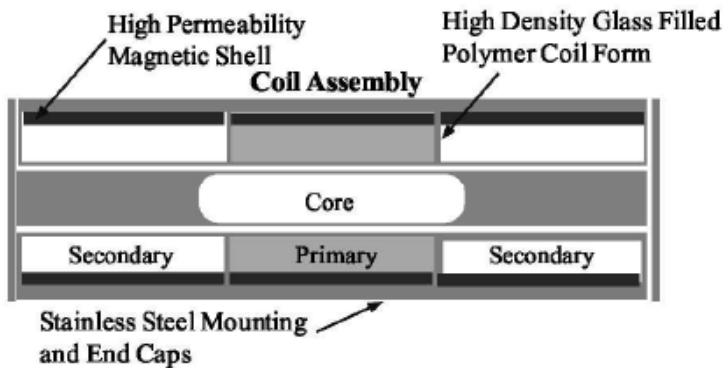
$$F = F_{d1} - F_{d2} - F_{s2} - F_{s3} \quad (2.21)$$

Dengan mensubstitusi persamaan (2.5), (2.6), (2.10) dan (2.11) didapatkan

$$\dot{F} = \dot{p}_1 A_1 - \dot{p}_2 A_2 - \frac{3}{2} k \dot{x}_p \quad (2.22)$$

Dari persamaan (2.1) dan (2.20), dapat diketahui bahwa posisi *spool valve* (x_v) yang diatur sinyal kontrol (u) dapat diatur untuk mengikuti besar gaya yang menentukan posisi piston (x_p) sesuai *set-point* (Sohl & Bobrow, 1999).

2.3. Sensor Posisi *Linear Variable Differential Transmitter* (LVDT)



Gambar 2. 6 Konstruksi LVDT (Balaban dkk, 2009)

LVDT adalah sensor posisi yang sering digunakan pada silinder hidrolik. LVDT memiliki keluaran elektrik dari sensor yang sebanding atau proporsional dengan posisi dari inti atau *core* magnetic dalam LVDT (Balaban dkk, 2009). Inti magnetic pada LVDT bergerak secara linear di dalam transformer yang terdiri dari kumparan primer dan dua kumparan sekunder sefasa yang dililitkan pada sebuah silinder kosong seperti yang ditunjukkan Gambar 2.6. Ketika arus dialirkan pada kumparan primer, akan dihasilkan medan magnetic. Ketika inti dari transformer digerakkan, efek medan magnet oleh kumparan primer akan mempengaruhi besar tegangan yang dihasilkan oleh kumparan sekunder.

Berikut bentuk kesalahan yang sering terjadi pada LVDT :

- Ketidaksamaan property antar kedua kumparan sekunder dapat mengakibatkan kesalahan bias
- Kebocoran medan magnetic antar kumparan sekunder dapat mengakibatkan kesalahan bias
- Perubahan pada tegangan primer dapat mengakibatkan perubahan pada tegangan di kumparan primer yang mengakibatkan kesalahan *scaling*.

Berdasarkan referensi yang digunakan untuk memodelkan sebuah aktuator hidrolik dalam (Sohl & Bobrow, 1999), (Fu, 2000) dan (Xu dkk, 2016) sensor posisi yang memiliki respon yang sangat cepat dapat dimodelkan dengan $G_S(s) = \frac{1}{10^{-4}s+1}$.

2.4. Kesalahan Sensor

Sebuah kesalahan adalah penyimpangan dari kondisi biasa atau standar yang tidak diinginkan pada setidaknya salah satu sifat atau property dalam sebuah sistem. Kesalahan dapat dikatakan sebagai sebuah *state* dalam sistem dan kesalahan dapat menjadi pemicu terjadinya kegagalan. Kegagalan adalah gangguan permanen yang mempengaruhi kemampuan sistem untuk melakukan fungsi yang dibutuhkan pada kondisi operasi tertentu (Isermann, 2006).

Sensor memiliki peran vital dalam memberikan performa yang baik dalam sebuah sistem. Kesalahan yang terjadi pada sensor dapat terjadi karena beberapa sebab seperti, kesalahan saat proses manufaktur, kerusakan yang terjadi karena lama waktu penggunaan, kalibrasi yang salah atau penggunaan sensor yang tidak benar. Kesalahan pada sensor dapat dibagi menjadi lima kategori (Balaban dkk, 2009).

- Bias
Offset atau penyimpangan konstan pada kondisi normal sinyal sensor. Bias juga dapat dideskripsika sebagai sinyal keluaran sensor ketika tidak mendapatkan input atau *input* = 0. Bias dapat terjadi karena kesalahan proses kalibrasi atau perubahan

secara fisik pada sistem pengukuran. Secara matematis bias memiliki persamaan

$$Y_f = X + \beta + noise \quad (2.23)$$

Dimana β adalah besar penyimpangan konstan. β yang berubah terhadap waktu menghasilkan kesalahan *Drift*

- *Drift*

Offset atau penyimpangan yang nilai berubah terhadap waktu pada kondisi normal sinyal sensor. Biasanya hanya *drift* linear yang dimodelkan dalam literature. Namun, *drift* non-linear dapat terjadi. Kesalahan *drift* direpresentasikan sebagai

$$Y_f = X + \delta(t) + noise \quad (2.24)$$

Dimana $\delta(t)$ merupakan factor penyimpangan yang berubah terhadap waktu.

- *Scaling*

Scaling atau sering disebut kesalahan *gain* terjadi ketika terdapat sebuah factor pengali $a(t)$ yang merubah sinyal sensor dari kondisi normalnya. *Scaling* direpresentasikan sebagai $Y_f = a(t) * X + noise, 0 < a(t) < \infty$ (2.25)

Dimana $a(t)$ adalah parameter *scaling* yang dapat bernilai konstan ataupun berubah terhadap waktu.

- *Noise*

Noise adalah adalah sinyal acak atau sinyal tak pasti yang memengaruhi sinyal keluaran sensor

- *Hard Fault*

Hard Fault terjadi ketika keluaran sensor tertahan pada satu nilai dimana dapat direpresentasikan dengan

$$Y_f = C + noise \quad (2.26)$$

Dimana C adalah konstanta.

2.5. Observer

Dalam pendekatan *pole-placement* pada perancangan sistem kontrol, diasumsikan bahwa semua variabel keadaan tersedia

untuk umpan balik. Namun, dalam praktiknya, tidak semua variabel keadaan tersedia untuk umpan balik. Maka diperlukan perkiraan variabel keadaan yang tidak tersedia. Estimasi variabel keadaan yang tidak terukur biasanya disebut observasi Perangkat (atau sebuah program komputer) yang memperkirakan atau mengamati variabel keadaan disebut *state observer*, atau bisa disebut *observer*. Jika *state observer* mengamati semua variabel keadaan sistem, terlepas dari apakah beberapa variabel keadaan tersedia atau tidak tersedia untuk pengukuran langsung, ini disebut *full order state observer*. Ada kalanya tidak seluruh *state* perlu dilakukan observasi. Sebagai contoh, karena variabel output dapat diamati dan berhubungan secara linear dengan variabel keadaan, kita tidak perlu mengamati semua variabel keadaan, namun hanya mengamati variabel $n-m$ *state*, di mana n adalah dimensi vektor keadaan dan m adalah dimensi dari Vektor keluaran (Ogata, 2010).

Observer yang memperkirakan kurang dari n variabel keadaan, disebut *reduced-order state observer* atau secara sederhana disebut *reduced-order observer*. Jika *observer* dengan tingkat reduksi minimal, maka *observer* disebut *minimum-order state observer* atau *minimum-order observer* (Ogata, 2010).

Model *state space* dari *plant* ditunjukkan oleh persamaan (2.27) dan (2.28)

$$\dot{x} = Ax + Bu \quad (2.27)$$

$$y = Cx \quad (2.28)$$

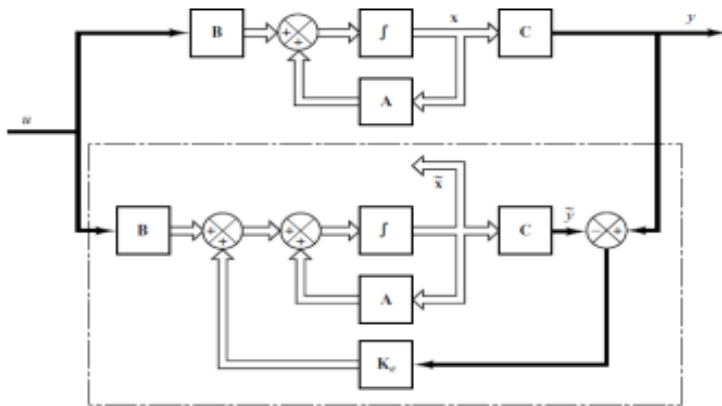
Dimana A, B dan C adalah matrix keadaan, matrix *input* dan matrix *output* secara berurutan, sedangkan x, u dan y adalah vektor keadaan, vektor *input* dan vektor *output* secara berurutan

Observer adalah subsystem untuk merekonstruksi vektor *state* dari *plant*. Model matematis dari *observer* pada dasarnya sama dengan *plant* namun dimasukkan syarat tambahan yang didalamnya terdapat perkiraan *error* untuk mengkompensasi ketidakakuratan dalam matrix A dan B , seperti yang ditunjukkan pada Gambar 2.6 perkiraan eror atau eror observasi adalah perbedaan diantara output hasil pengukuran dengan hasil estimasi output. Model matematis dari *observer* adalah

$$\dot{\hat{x}} = A\hat{x} + Bu + K_e(y - C\hat{x}) \quad (2.29)$$

$$\hat{\dot{x}} = (A - K_e C)\hat{x} + Bu + K_e y \quad (2.30)$$

Dimana \hat{x} adalah perkiraan keadaan dan $\hat{y} = C\hat{x}$ merupakan perkiraan output. Masukan ke *observer* adalah output y dan input *control* u . Matrix K_e , yang disebut matrix gain *observer*, adalah matrix pembobotan dengan istilah koreksi yang melibatkan perbedaan antara output terukur y dan hasil perkiraan $\hat{y} = C\hat{x}$.



Gambar 2. 7 Sistem dengan *Full Order Observer* (Ogata, 2010)

2.6. Algoritma *Exponential Smoothing*

Metode *Holt* dapat diadopsi untuk memproses prediksi sinyal. Prediksi dapat dihitung melalui *exponential smoothing algorithm* sebagai berikut :

$$u_n = a\hat{f}_n + (1 - a)(u_{n-1} + s_{n-1}) \quad (2.31)$$

$$s_n = b(u_n - u_{n-1}) + (1 - b)s_{n-1} \quad (2.32)$$

u_n = Hasil estimasi yang diperhalus

s_n = Faktor pertumbuhan linear

\hat{f}_n = Hasil estimasi kesalahan

a & b = Parameter Holt

$$u_1 = \hat{f}_1 = \hat{f}(k) \quad (2.33)$$

$$u_2 = \hat{f}_2 = \hat{f}(k + 1) \quad (2.34)$$

$$s_2 = u_2 - u_1 \quad (2.35)$$

Diberikan hasil estimasi kesalahan $\hat{f}_n (n = 1, 2, \dots)$ yang diperoleh dari *observer*. Dimana $0 < a < 1$ & $0 < b < 1$ adalah parameter *Holt's method*. Kondisi awal untuk *Holt's method* sebagai berikut: Setelah menghitung estimasi pada saat nT dan estimasi faktor pertumbuhan, maka prediksi sinyal dapat dihitung menggunakan algoritma berikut:

$$\hat{f}_{n+l} = u_n + s_n lT \quad (2.36)$$

Dengan

l = step prediksi

T = *time interval* untuk prediksi

2.7. Prediksi Keandalan

Keandalan didefinisikan sebagai kemampuan suatu sistem atau komponen untuk melakukan fungsi yang diperlukan dalam kondisi yang ditentukan untuk waktu yang ditentukan (Connor, 2012). Menurut (Dhillon, 2004) keandalan adalah kemungkinan untuk tidak mengalami kegagalan atau dapat melaksanakan fungsinya selama periode waktu (t). Evaluasi keandalan dapat membantu dalam memperkirakan peluang sebuah sistem atau komponen untuk dapat melaksanakan fungsinya dalam jangka waktu tertentu dan dalam kondisi tertentu. Sistem atau komponen masih dapat beroperasi walaupun tidak dapat memenuhi fungsinya secara maksimal. Nilai fungsi *reliability* merupakan nilai probabilitas, maka nilai fungsi *reliability* (R) bernilai $0 \leq R \leq 1$ (Ebeling, 1997)

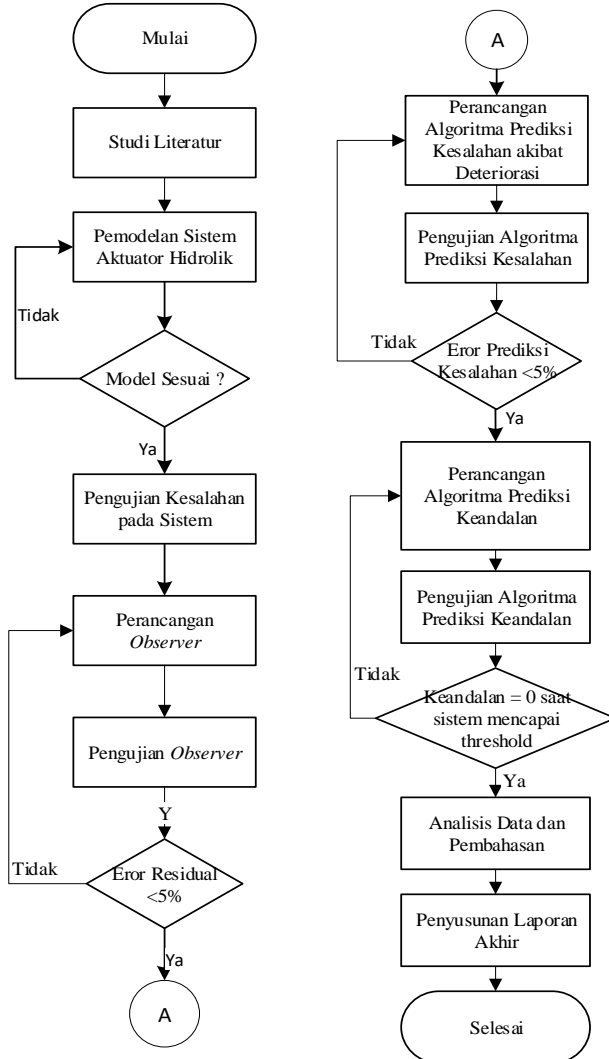
Prediksi berkaitan dengan evaluasi sebuah desain sebelum dilaksanakan konstruksi nyata. Prediksi merupakan sebuah usaha untuk mengetahui konsekuensi dari keputusan sebelum sistem berfungsi seutuhnya. Prediksi biasanya dilakukan menganalisa model dari sistem yang memberikan dasar untuk pengujian perencanaan, pembuatan dan evaluasi pertumbuhan keandalan, *maintenance* dan aktivitas manajemen lain..

Untuk memprediksi keandalan sistem, ada beberapa hal penting yang harus diperhatikan. Pertama, selain dari definisi kegagalan sebagai nilai kesalahan yang telah mencapai batas, kegagalan sistem juga didefinisikan sebagai keadaan sistem yang telah memasuki area terlarang \mathcal{M}_0 . Kedua, keadaan sistem tidak dapat diukur secara langsung, harus melalui estimasi dikarenakan kegagalan sistem didefinisikan dalam sebuah keadaan. Keadaan diestimasi menggunakan algoritma OTK-SIR (Xu et al., 2009).

“Halaman ini sengaja dikosongkan”

BAB III METODOLOGI PENELITIAN

Flowchart penelitian tugas akhir terdapat pada Gambar 3.1 ::



Gambar 3.1 Diagram Alir Penelitian

3.1 Studi Literatur

Studi literatur merupakan tahapan awal untuk menunjang pengerjaan penelitian ini. Literatur yang dapat digunakan antara lain buku, jurnal penelitian, artikel ilmiah dan laporan tesis maupun disertasi. Kegiatan yang dilakukan dalam studi literatur yaitu pengayaan materi yang berkaitan dengan topik prediksi kehandalan. Teori yang dipelajari antara lain pemodelan sistem pengendalian posisi pada aktuator hidrolis, kesalahan dalam sistem pengendalian, perancangan *observer* sebuah sistem, algoritma prediksi kesalahan dan algoritma penghitungan keandalan.

3.2 Pemodelan Sistem Pengendalian Posisi Aktuator Hidrolis

Pemodelan sistem merupakan langkah awal dalam penelitian ini. Pemodelan aktuator hidrolis mengacu pada model (Fu, 2000). Pada tabel 3.1 ditunjukkan nilai parameter yang digunakan untuk melakukan pemodelan sistem.

Tabel 3. 1 Parameter pemodelan aktuator hidrolis (Fu, 2000)

Parameter	Nilai	Satuan
Massa beban piston	0,1	<i>kg</i>
Konstanta pegas	2200	<i>N/m</i>
Modulus <i>bulk</i> fluida	$6,9 \cdot 10^5$	<i>kPa</i>
Volum fluida di silinder utama piston	155	<i>cm</i> ³
Volum fluida di ruang pendorong piston	83	<i>cm</i> ³
Luas permukaan silinder utama piston	6.8	<i>cm</i> ²
Luas permukaan ruang pendorong piston	9,6	<i>cm</i> ²
Koefisien bukaan <i>spool valve</i>	1	
Tekanan suplai	6895	<i>kPa</i>
Tekanan <i>reservoir</i>	100	<i>kPa</i>
Gain <i>spool valve</i>	1	
<i>Time constant spool valve</i>	0,1	<i>s</i>

Sebuah aktuator hidrolik bekerja dengan prinsip perbedaan tekanan hidrolik yang terdapat pada ruang silinder utama dan ruang pendorong. Besar tekanan hidrolik yang diberikan pada ruang silinder dan ruang pendorong ditentukan oleh *spool valve* 4/3 yang telah menerima sinyal perintah dari sebuah kontroler. Kontroler bertugas untuk memberikan posisi torak yang sesuai dengan *set point* sistem pengendalian.

Pemodelan sistem pengendalian posisi aktuator hidrolik dapat dibagi menjadi 2 subsistem yaitu *spool valve dynamics* dan *piston dynamics*.

- **Pemodelan Aktuator : *Spool Valve* .**

Terdapat 2 proses pemodelan pada *spool valve* yaitu model dinamik *spool valve* dan model dinamik aliran yang dihasilkan oleh bukaan *spool valve* (x_v) tersebut. Model dinamik *spool valve* ditunjukkan pada persamaan (2.2), (2.3) dan (2.4)

$$G_v(s) = \frac{x_v(s)}{u(s)} = \frac{k_v}{\tau s + 1} \quad (2.2)$$

$$q_1 = \begin{cases} c_1 x_v \sqrt{p_s - p_1} & x_v \geq 0 \\ c_2 x_v \sqrt{p_1 - p_r} & x_v < 0 \end{cases} \quad (2.3)$$

$$q_2 = \begin{cases} -c_3 u \sqrt{p_2 - p_r} & x_v \geq 0 \\ -c_4 u \sqrt{p_s - p_2} & x_v < 0 \end{cases} \quad (2.4)$$

- **Pemodelan Plant Piston**

Piston yang dapat dikatakan sebagai *plant* dalam sistem aktuator hidrolik memiliki tingkat non-linearitas yang tinggi karena terdapat aspek non-linear seperti dinamika cairan hidrolik yang berlaku dalam proses pemodelannya. Penurunan persamaan dinamis cairan hidrolik telah dijelaskan sebelumnya pada Bab II, sehingga pada bagian ini cukup dipaparkan persamaan yang langsung berkaitan dengan gerak piston. Pemodelan pada piston dapat dibagi menjadi dua. Pertama, pemodelan besarnya tekanan hidrolik $p_{1,2}$ yang

diakibatkan aliran cairan hidrolis $q_{1,2}$. Kedua, besarnya tekanan hidrolis tersebut akan digunakan untuk mendapatkan dinamika gerak piston yang akan memberikan informasi posisi piston x_p . Persamaan (2.20) menunjukkan model matematika p_1 dan p_2 . Sedangkan persamaan (2.22) menunjukkan model matematika gerak piston.

$$\dot{p}_1 = \begin{cases} \frac{\beta}{V_1} (-A_1 \dot{x}_p + c_1 x_v \sqrt{p_s - p_1}) & x_v \geq 0 \\ \frac{\beta}{V_1} (-A_1 \dot{x}_p + c_2 x_v \sqrt{p_1 - p_r}) & x_v < 0 \end{cases} \quad (2.20)$$

$$\dot{p}_2 = \begin{cases} \frac{\beta}{V_2} (A_2 \dot{x}_p - c_3 x_v \sqrt{p_2 - p_r}) & x_v \geq 0 \\ \frac{\beta}{V_2} (A_2 \dot{x}_p - c_4 x_v \sqrt{p_s - p_2}) & x_v < 0 \end{cases}$$

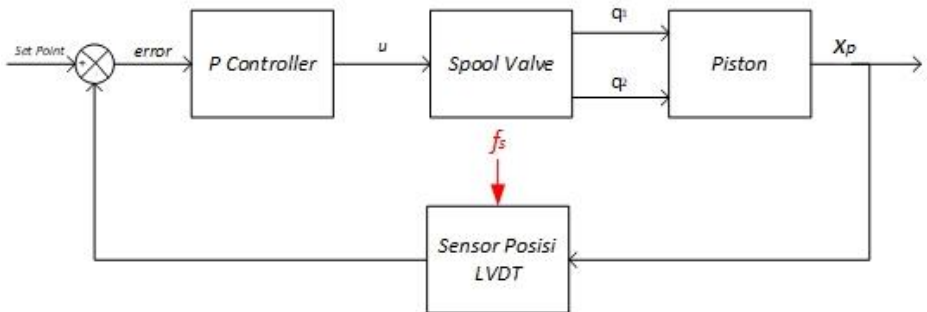
$$\dot{F} = \dot{p}_1 A_1 - \dot{p}_2 A_2 - \frac{3}{2} k \dot{x}_p \quad (2.22)$$

- **Pemodelan Sensor : *Linear Variable Differential Transmitter (LVDT)***

Berdasarkan referensi yang digunakan untuk memodelkan sebuah aktuatur hidrolis dalam (Sohl & Bobrow, 1999), (Fu, 2000) dan (Xu dkk, 2016) sensor posisi yang memiliki respon yang sangat cepat dapat dimodelkan dengan $G_s(s) = \frac{1}{10^{-4}s+1}$.

- ***Controller***

Sebuah aktuatur hidrolis akan membutuhkan algoritma kontrol yang sesuai untuk mendapatkan performa atau nilai variabel kontrol yang sesuai dengan *set point* yang ditentukan. Dapat dilihat bahwa model matematis dari sebuah aktuatur hidrolis memiliki komponen non-linear, sehingga bermacam variasi strategi kontrol diajukan untuk mendapatkan sistem pengendalian yang baik. Pada penelitian ini digunakan pengendali *Proportional (P)*. Pengendali P dipengaruhi oleh



Gambar 3.2 Diagram Blok Sistem Pengendalian Aktuator Hidrolik

besar nilai gain P. Hanya digunakan pengendali P karena dengan pengendali tersebut sudah didapatkan performa yang baik dimana error *steady state* dari sistem adalah sebesar 0 dan waktu untuk mencapai nilai *steady state* dapat ditentukan dengan besar *Gain* P tersebut. Dalam penelitian ini penentuan parameter P dilakukan dengan cara *trial and error*

3.3 Pengujian Kesalahan pada Sistem Pengendalian

Sebuah sistem pengendalian dapat mengalami kegagalan dikarenakan adanya kesalahan atau kegagalan pada sensor atau aktuator dalam bekerja. Dalam penelitian ini, beberapa kesalahan telah dicoba diberikan pada sistem. Kesalahan pertama adalah terjadinya *wear-out* pada pegas dalam silinder utama. Kesalahan ini dimodelkan dengan pengurangan nilai konstanta pegas k . Namun setelah dilakukan pengujian, kesalahan tersebut tidak memberikan efek yang signifikan pada performa sistem. Kesalahan kedua yang telah dicoba diberikan pada adalah terjadinya *internal leakage* atau kebocoran pada silinder utama piston. Kesalahan ini dicoba dimodelkan dengan terjadinya pengurangan pada besar koefisien bukaan *valve*. Namun sama halnya dengan kesalahan *wear out* pegas, tidak terjadi perubahan yang signifikan pada performa sistem. Percobaan ketiga adalah dengan memberikan kesalahan pada sensor posisi *LVDT*.

. Terdapat 5 jenis kesalahan atau kegagalan yang sering terjadi pada sensor yaitu : *Bias*, *Drift*, *Scaling*, *Noise* dan *Hard fault* (Balaban dkk, 2009). Pada penelitian ini kesalahan yang akan diaplikasikan pada sistem pengendalian posisi aktuator hidrolik adalah kesalahan sensor *Scaling*. Kesalahan sensor *Scaling* atau juga sering disebut *Gain Failure* merupakan kesalahan sensor dimana bacaan sensor akan melenceng dari yang sesungguhnya secara berskala dengan magnitudo $a(t)$.

$$Y_f = a(t) \cdot X \quad (3.1)$$

Dalam simulasi telah diberikan kesalahan *scaling* dimana sinyal kesalahan $a(t)$ merupakan sinyal berbentuk *ramp* dengan kemiringan $-0.005/t$.

3.4 Perancangan *Observer*

Kesalahan yang telah diaplikasikan pada model sistem dalam praktik nyatanya tidak dapat diketahui ataupun diukur nilainya menggunakan sensor. Oleh karena itu *observer* dapat digunakan untuk menghasilkan estimasi kesalahan pada sistem pengendalian. *Observer* menggunakan bentuk *state space* atau ruang keadaan sebagai model dasar algoritmanya. Persamaan *state space* dapat diperoleh berdasarkan model matematis yang telah ditampilkan pada sub bab 3.2. Bentuk *state space* yang digunakan untuk memodelkan sistem pengendalian posisi aktuator hidrolik adalah

$$\dot{x} = A \cdot x + B \cdot u + G \cdot g(x) \quad (3.2)$$

$$y = C \cdot x + F_s \cdot f_s \quad (3.3)$$

Persamaan (3.2) dan (3.3) menunjukkan bentuk persamaan ruang keadaan yang memiliki kesalahan pada sensor. Persamaan ruang keadaan ini akan digunakan sebagai model dasar *observer*. Dimana x adalah vektor keadaan, y adalah vektor output, A merupakan matrix *state* sistem, B adalah matrix input dan C adalah matrix output sistem. Sedangkan $g(x)$ merupakan bagian non-linear dari model sistem, f_s merupakan vektor kesalahan pada aktuator dan sensor.

Matrix *state* A tersusun oleh model matematis yang telah ditampilkan pada sub bab 3.2. Sistem memiliki 5 vektor *state* x yaitu $x = [p_1 \ p_2 \ x_p \ \dot{x}_p \ x_v]^T$

$$\begin{aligned}
 \begin{bmatrix} \dot{p}_1 \\ \dot{p}_2 \\ \dot{x}_p \\ \ddot{x}_p \\ \dot{x}_v \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 0 & -\frac{B_p A_1}{V_{10}} & 0 \\ 0 & 0 & 0 & \frac{B_p A_1}{V_{20}} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \frac{A_1}{m} & -\frac{A_2}{m} & -\frac{(3k/2)}{m} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{\tau} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ x_p \\ \dot{x}_p \\ x_v \end{bmatrix} \\
 &+ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{k_v}{\tau} \end{bmatrix} u \\
 &+ \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} g_1(x) = \begin{cases} c_1 x_v \sqrt{p_s - p_1} & x_v \geq 0 \\ c_2 x_v \sqrt{p_1 - p_r} & x_v < 0 \end{cases} \\ g_2(x) = \begin{cases} -c_3 x_v \sqrt{p_2 - p_r} & x_v \geq 0 \\ -c_4 x_v \sqrt{p_s - p_2} & x_v < 0 \end{cases} \end{bmatrix}
 \end{aligned} \tag{3.4}$$

$$y = [0 \ 0 \ 1 \ 0 \ 0] \begin{bmatrix} p_1 \\ p_2 \\ x_p \\ \dot{x}_p \\ x_v \end{bmatrix} + [1] f_s \tag{3.5}$$

Dimana

$$A = \begin{bmatrix} 0 & 0 & 0 & -\frac{B_p A_1}{V_{10}} & 0 \\ 0 & 0 & 0 & \frac{B_p A_1}{V_{20}} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \frac{A_1}{m} & -\frac{A_2}{m} & -\frac{(3k/2)}{m} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{\tau} \end{bmatrix}; x = \begin{bmatrix} p_1 \\ p_2 \\ x_p \\ \dot{x}_p \\ x_v \end{bmatrix};$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{k_v}{\tau} \end{bmatrix}; G = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}; y = \begin{bmatrix} p_1 \\ p_2 \\ x \end{bmatrix}; F_s = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$g(x) = \begin{bmatrix} g_1(x) = \begin{cases} \frac{\beta}{v_1} c_1 x_v \sqrt{p_s - p_1} & x_v \geq 0 \\ \frac{\beta}{v_1} c_2 x_v \sqrt{p_1 - p_r} & x_v < 0 \end{cases} \\ g_2(x) = \begin{cases} -\frac{\beta}{v_2} c_3 x_v \sqrt{p_2 - p_r} & x_v \geq 0 \\ -\frac{\beta}{v_2} c_4 x_v \sqrt{p_s - p_2} & x_v < 0 \end{cases} \end{bmatrix};$$

Langkah selanjutnya adalah mengubah *state space* pada persamaan (3.2) dan (3.3) menjadi *augmented state space*. *Augmented State Space* digunakan untuk memindahkan vektor kesalahan sensor dari persamaan output menjadi persamaan keadaan. Hal ini dilakukan dengan menciptakan *state* baru z .

$$\dot{z} = A_z(y - z) \quad (3.6)$$

Dengan mensubstitusi persamaan (3.5) ke dalam persamaan (3.6), diperoleh persamaan (3.7)

$$\dot{z} = A_z \cdot C \cdot x + A_z \cdot F_s \cdot f_s - A_z z \quad (3.7)$$

Dengan menjadikan z sebagai *state* baru pada persamaan keadaan maka didapatkan *augmented state space*

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A & 0 \\ A_z C & -A_z \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} G \\ 0 \end{bmatrix} g(x) + \begin{bmatrix} 0 \\ A_z F_s \end{bmatrix} [f_s] \quad (3.8)$$

$$y = [0 \quad I] \begin{bmatrix} x(k) \\ z(k) \end{bmatrix} \quad (3.9)$$

A_z merupakan matrix identitas yang nilainya dapat diubah dengan mengalikannya dengan konstanta.

Persamaan (3.8) dan (3.9) menunjukkan ruang keadaan yang baru yang telah mengandung kesalahan sensor dalam persamaan keadaannya. Persamaan (3.10) menunjukkan bentuk ruang keadaan yang kompak

$$\dot{\bar{x}}_{6 \times 1} = \bar{A}_{6 \times 6} \bar{x}_{6 \times 1} + \bar{B}_{6 \times 1} \cdot u + \bar{G}_{6 \times 2} \cdot g(x) + \bar{F}_{6 \times 1} \cdot \bar{f} \quad (3.10)$$

$$z_{1 \times 1} = \bar{y} = \bar{C}_{1 \times 6} \cdot \bar{x}_{6 \times 1} \quad (3.11)$$

Dengan nilai matrix sebagai berikut;

$$\bar{x} = \begin{bmatrix} x \\ z \end{bmatrix}; \bar{A} = \begin{bmatrix} 0 & 0 & 0 & -\frac{B_p A_1}{V_{10}} & 0 & 0 \\ 0 & 0 & 0 & \frac{B_p A_1}{V_{20}} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{A_1}{m} & -\frac{A_2}{m} & -\frac{(3k/2)}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{\tau} & 0 \\ 0 & 0 & A_z & 0 & 0 & -A_z \end{bmatrix};$$

$$\bar{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{k_p}{\tau} \\ 0 \end{bmatrix}; \bar{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}; \bar{F} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ A_z \end{bmatrix}; \bar{C} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}^T$$

Berdasarkan persamaan ruang keadaan yang telah didapat, *observer* dari sistem akan memiliki persamaan berupa :

$$\begin{aligned} \dot{\hat{x}}_{6x1} &= \bar{A}_{6x6} \cdot \hat{x}_{6x1} + \bar{B}_{6x1} \cdot u + \bar{F}_{6x1} \hat{f} + \bar{G}_{6x2} \cdot g(x) \\ &\quad + K_{e(6x1)} (\bar{y}_{1x1} - \hat{\bar{y}}_{1x1}) \end{aligned} \quad (3.12)$$

$$\hat{\bar{y}}_{1x1} = \bar{C}_{1x6} \hat{x}_{6x1} \quad (3.13)$$

Kesalahan yang terdapat pada sistem diestimasi menggunakan persamaan berikut :

$$\dot{\hat{f}} = L \cdot e + \hat{f} \quad (3.14)$$

Dengan L adalah *gain observer* untuk *state* kesalahan dan e adalah $\bar{y} - \hat{\bar{y}}$. Dengan menjadikan persamaan (3.14) sebagai *state* baru, didapatkan persamaan *observer* berupa:

$$\begin{bmatrix} \dot{\hat{x}} \\ \dot{\hat{f}} \end{bmatrix} = \begin{bmatrix} \bar{A} & \bar{F} \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{f} \end{bmatrix} + \begin{bmatrix} \bar{B} \\ 0 \end{bmatrix} u + \begin{bmatrix} \bar{G} \\ 0 \end{bmatrix} g(x) + \begin{bmatrix} K \\ L \end{bmatrix} e \quad (3.15)$$

$$\hat{\bar{y}} = [\bar{C} \quad I] \begin{bmatrix} \hat{x} \\ \hat{f} \end{bmatrix} \quad (3.16)$$

Sehingga bentuk kompak dari persamaan *observer* yang telah ditambahkan *state* kesalahan dapat dituliskan menjadi :

$$\begin{aligned} \dot{\hat{x}}_{7x1} &= \tilde{A}_{7x7} \cdot \hat{x}_{7x1} + \tilde{B}_{7x1} \cdot u + \tilde{G}_{7x2} \cdot g(x) + \\ &\quad \tilde{K}_{e(7x1)} \cdot (\tilde{y}_{1x2} - \hat{\tilde{y}}_{2x1}) \end{aligned} \quad (3.17)$$

$$\hat{\tilde{y}}_{2x1} = \tilde{C}_{2x7} \cdot \hat{x}_{7x1} \quad (3.18)$$

Dengan nilai matrix sebagai berikut :

$$\tilde{A} = \begin{bmatrix} 0 & 0 & 0 & -\frac{B_p A_1}{V_{10}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{B_p A_1}{V_{20}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \frac{A_1}{m} & -\frac{A_2}{m} & -\frac{(3k/2)}{m} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{\tau} & 0 & 0 \\ 0 & 0 & A_z & 0 & 0 & -A_z & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\tilde{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{k_v}{\tau} \\ 0 \\ 0 \\ 0 \end{bmatrix}; \tilde{G} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}; \tilde{C} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}^T$$

Setelah persamaan *observer* berhasil ditemukan, tugas selanjutnya adalah menentukan nilai *gain observer*. Pada penelitian ini *gain observer* ditemukan dengan menggunakan metode *Linear Quadratic Regulator* (LQR). LQR sesungguhnya adalah metode untuk menentukan *gain* kontroler pada sistem dengan *state feedback*, namun dikarenakan terdapatnya dualitas antara *controllability* dan *observability*, metode penentuan *gain state feedback* dapat digunakan untuk menentukan *gain observer* (Oppenheim & Verghese, 2010). Berbeda dengan metode *pole placement* dimana *pole* harus ditentukan terlebih dahulu untuk mendapatkan *gain* yang sesuai, LQR memiliki parameter Q dan R sebagai *weighing matrix* dalam proses minimisasi fungsi objektif. Berikut fungsi objektif minimisasi LQR:

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

Dengan Q dan R merupakan *weighing matrix* positif definit yang nilainya dapat diubah.

Secara sederhana, matrix Q bekerja sebagai matrix yang menentukan performa *state* yang ditunjukkan, dimana jika elemen pada matrix Q yang berkorespondensi dengan *state* tertentu bernilai besar, maka performa kerja dari *state* tersebut akan bernilai baik atau dengan kata lain memiliki eror yang kecil. Sementara matrix R merupakan matrix yang mengindikasikan *cost* atau usaha dalam bentuk sinyal kontrol *u* dalam mencapai performa yang diinginkan. Semakin besar elemen dalam matrix R maka besar usaha sistem akan semakin diminimisasi. Karena dalam penelitian ini LQR digunakan sebagai *gain observer* yang tidak mempengaruhi performa dari *real plant*, maka nilai Q dan R dapat

dipilih dengan mengabaikan efek yang akan terjadi pada performa sistem pengendalian. Pemilihan Q dan R berfokus hanya untuk mendapatkan $e = 0, t \rightarrow \infty$. Dalam penelitian ini, nilai matrix Q dan R yang digunakan adalah:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 9e8 \end{bmatrix}$$

$$R = [1]$$

3.5 Pengujian *Observer*

Pengujian *observer* dilakukan untuk memastikan apakah *observer* mampu mengestimasi nilai kesalahan sensor \hat{f} . Persamaan (3.20) menunjukkan besar nilai residual sebagai sinyal kesalahan sensor yang sesungguhnya

$$f_{residual} = \hat{f} - x_{p,true} \quad (3.20)$$

Dengan

\hat{f} = Hasil pengukuran sensor setelah terjadi kesalahan

$x_{p,true}$ = Hasil pengukuran sensor sebelum terjadi kesalahan

3.6 Perancangan Algoritma Prediksi Kesalahan

Setelah perancangan algoritma *observer*, langkah berikutnya perancangan algoritma prediksi kesalahan yang mengacu pada (Xu, Ji, & Zhou, 2009) yaitu algoritma *exponential smoothing*. Algoritma *exponential smoothing* membutuhkan sinyal berbentuk diskrit agar algoritma dapat bekerja. Oleh karena itu keluaran estimasi kesalahan \hat{f} oleh *observer* harus didiskritkan menjadi $\hat{f}(k)$ dengan metode *zero-order-hold* dengan *time sample* T_s . Algoritma *Exponential Smoothing* ditunjukkan dalam persamaan (2.27) hingga (2.32).

Penjelasan mengenai algoritma prediksi kesalahan adalah sebagai berikut :

1. Step estimasi kesalahan yang diperhalus: Jika $n = 1$, atau $n = 2$, menghitung estimasi kesalahan yang diperhalus u_n menggunakan persamaan (2.29), dan (2.30). Jika $n \geq 3$ menghitung estimasi kesalahan yang diperhalus u_n menggunakan persamaan (2.27).
2. Step estimasi faktor pertumbuhan: Menghitung n -step prediksi faktor pertumbuhan s_n dengan menggunakan (2.30) jika $n = 2$, dan menghitung n -step estimasi faktor pertumbuhan s_n dengan menggunakan (2.28) jika $n \geq 3$.
3. Step prediksi kesalahan: Menghitung prediksi kesalahan $\bar{\alpha}_{n+l}$ dengan menggunakan persamaan (2.32).

3.7 Perancangan Algoritma Prediksi Keandalan *Real-time*

Langkah terakhir dalam perancangan sistem monitoring dan prediksi keandalan *real-time* ini adalah perancangan algoritma prediksi keandalan. Algoritma prediksi keandalan membutuhkan hasil prediksi kesalahan dari algoritma *exponential smoothing* dan nilai hasil pengukuran sensor *real plant* yang mengandung kesalahan.

$$\hat{x}_{p,true} = x_p - \hat{f}_s \quad (3.21)$$

Dalam algoritma prediksi keandalan, nilai prediksi kesalahan akan diberikan pada hasil pengukuran sensor yang mengandung kesalahan untuk mengetahui nilai hasil ukur yang sesungguhnya. Jika nilai posisi $x_{p,true}$ melewati nilai *threshold*, maka nilai keandalan akan berkurang

$$R(nT, (n+r)T) = \Pr(x_{n+r} \in , \mathcal{M}_0 | x_n \in \mathcal{M}_0) \quad (3.22)$$

Penjelasan mengenai algoritma RP (*Reability Prediction*) adalah sebagai berikut :

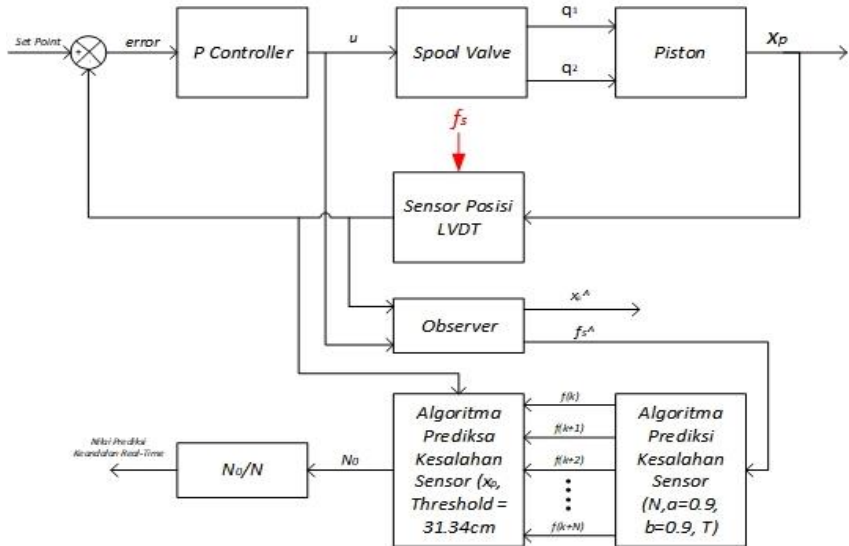
1. Step inisialisasi : l yang telah dinotasikan pada algoritma prediksi kesalahan dijadikan $l = 1$. Inisialisasi partikel menjadi $\bar{X}_{n+l-1|n}^{(i)} = \bar{X}_{n+0|n}^{(i)} = \hat{X}_{n|n}^{(i)}$, $i = 1, \dots, N$.

2. Step estimasi kesalahan : Jalankan algoritma *observer* untuk menghasilkan estimasi kesalahan \hat{f}_1 dan \hat{f}_2
3. Step prediksi kesalahan: Jalankan algoritma prediksi kesalahan dengan l langkah untuk menghitung prediksi kesalahan \hat{f}_{n+l} .
4. Step prediksi partikel : untuk $i = 1, \dots, N$, hitung partikel baru $\bar{X}_{n+l-1|n}^{(i)}$ dari sistem persamaan $\bar{X}_{n+l-1|n}^{(i)} = f(\bar{X}_{n+l-1|n}^{(i)}) + \Gamma \tilde{\alpha}_{n+l} + W_{n+l}^{(i)}$ dimana $W_{n+l}^{(i)}$ adalah realisasi dari w_{n+l} untuk partikel ke- i .
5. Step pemilihan partikel : jika $l < r$, maka $l = l + 1$, dan kembali ke step 2, jika sebaliknya lanjut pada step 6.
6. Step perhitungan prediksi keandalan : hitung jumlah partikel dari himpunan $\{\bar{X}_{n+r|n}^{(i)}\} (i = 1, \dots, N)$ atau \mathcal{M}_0 , yaitu himpunan nilai $x_{p,true}$ yang melewati batas nilai *threshold*, dan N_0 jumlah himpunan $x_{p,true}$ yang belum melewati *threshold*. Maka sistem prediksi keandalan selama $(nT, (n+r)T]$ dihitung oleh N_0/N .

Batas nilai *threshold* dalam penentuan tingkat kesehatan sistem adalah panjang maksimum dari silinder hidrolik. Silinder hidrolik yang digunakan memiliki panjang maksimum 31,34 cm. Jika nilai posisi piston $x_{p,true}$ telah mencapai *threshold* hal ini dapat menjadi penyebab kerusakan pada piston karena tekanan dari supply yang tetap diberikan pada piston meskipun ruang untuk bergerak bagi piston sudah tidak tersedia (Raza dkk,1995).

3.8 Pengujian Algoritma Prediksi Keandalan *Real-time*

Pengujian algoritma prediksi keandalan *real-time* dilakukan untuk mengetahui bentuk hasil algoritma prediksi keandalan *real-time* dan mengetahui parameter yang mempengaruhi hasil algoritma. Pada persamaan (3.22) telah ditunjukkan persamaan yang menunjukkan algoritma prediksi keandalan. Beberapa parameter yang sekiranya akan mempengaruhi algoritma terdapat pada algoritma prediksi kesalahan, sehingga pengujian dan variasi



Gambar 3. 3 Diagram blok sistem prediksi keandalan *real-time*

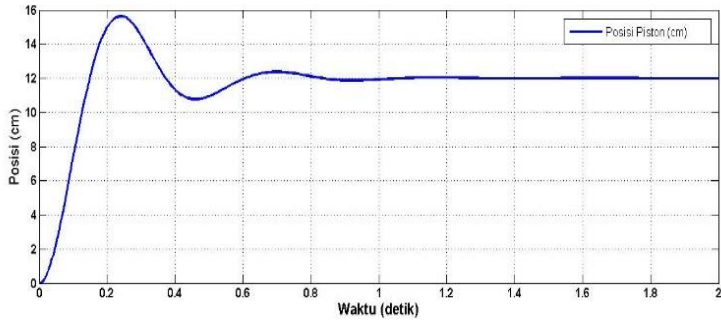
parameter yang diberikan pada algoritma prediksi kesalahan akan berpengaruh secara langsung pada algoritma prediksi keandalan *real-time*.

Parameter metode *Holt* disimbolkan dengan a dan b yang nilainya $0 < a, b < 1$. Parameter lain yang akan mempengaruhi algoritma ini adalah *threshold* atau nilai variabel proses (dalam hal ini posisi piston x_p) yang diperbolehkan dan jumlah partikel N , yang sesungguhnya adalah jumlah step prediksi kesalahan $l+2$. Nilai batas *threshold* dalam algoritma ini akan digunakan sebesar 31,34 cm berdasarkan model yang dikembangkan Fu dalam (Fu, 2000). Pengujian kebenaran algoritma RP dilakukan dengan meninjau waktu simulasi dimana posisi piston x_p mencapai nilai batas *threshold* dan membandingkannya dengan waktu dimana algoritma RP menyatakan keandalan sistem $R(t+nT) = 0$.

"Halaman ini sengaja dikosongkan"

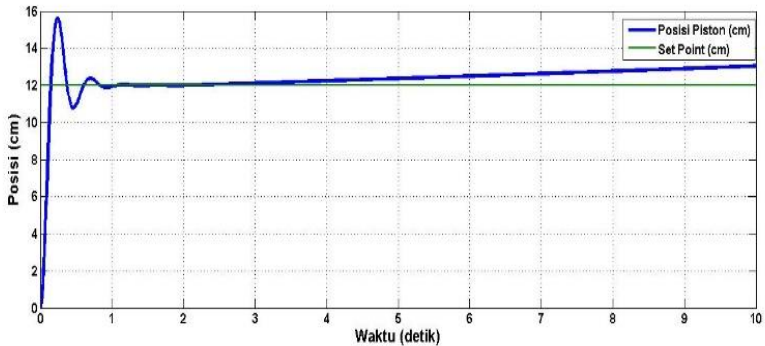
BAB IV ANALISIS DATA DAN PEMBAHASAN

4.1 Hasil Pemodelan dan Pemberian Kesalahan



Gambar 4. 1 Respon Sistem Pengendalian

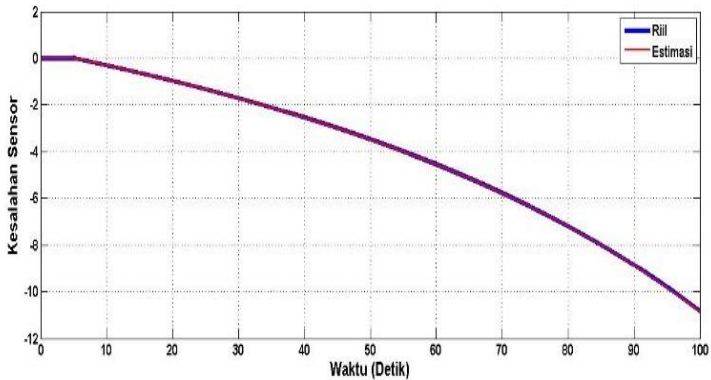
Telah dilakukan pengujian hasil pemodelan dengan melakukan pemodelan sistem pengendalian aktuator hidrolik pada Simulink dalam Matlab R2014a. Dengan *set point* berupa sinyal *step* sebesar 12 cm, ditemukan pada Gambar 4.1 bahwa respon dari sistem pengendalian memiliki karakteristik respon yang sama dengan hasil respon simulasi yang telah dilakukan Sohl *et al* dalam (Sohl & Bobrow, 1999).



Gambar 4. 2 Respon Sistem Setelah Diberi Kesalahan *Scaling*

Sistem lalu diberi kesalahan berupa kesalahan *scaling* pada sensor posisi. Kesalahan berupa sinyal *ramp* dengan *slope* $-0,005/t$ yang dimulai pada detik ke-3 simulasi. Pada Gambar 4.2 dapat dilihat bahwa nilai posisi sistem mulai menjauhi nilai set point yang telah dicapai sebelumnya. Deviasi yang dimiliki sistem dari nilai *set point* ini akan terus bertambah selama efek kesalahan *scaling* tetap diberikan pada sensor posisi.

4.2 Hasil Uji Observer



Gambar 4.3 Grafik hasil estimasi kesalahan sensor

Pengujian *observer* dilakukan untuk mengetahui kemampuan *observer* dalam mengestimasi kesalahan sensor. Sebelumnya kesalahan sensor disimulasikan sebagai kesalahan *scaling* dengan tingkat penurunan sensitivitas sebesar $a(t)$. Kesalahan $a(t)$ disimulasikan sebagai sinyal *ramp* dengan kemiringan $-0,005/t$ dan dimulai pada detik ke-5. Garis berwarna biru menunjukkan hasil kesalahan riil atau residual dari sensor plant dan garis berwarna merah menunjukkan hasil estimasi kesalahan sensor oleh *observer*.

Gambar 4.3 menunjukkan *observer* mampu mengestimasi kesalahan sensor dengan baik. Tidak terdapat *overshoot* dan waktu yang dibutuhkan untuk estimasi memiliki nilai yang sama dengan nilai riil juga spontan. Hal ini menunjukkan bahwa *gain observer*

yang diperoleh menggunakan LQR berhasil memberikan *observer* yang memiliki hasil estimasi yang optimal.

4.3 Hasil Uji Algoritma Prediksi Kesalahan Sensor

Prediksi kesalahan sensor diterapkan menggunakan hasil estimasi kesalahan sensor dari *observer*. Algoritma prediksi kesalahan sensor atau algoritma *exponential smoothing*, memiliki parameter *Holt' method* (a dan b). Dalam pengujian ini nilai $a = 0,9$ dan $b = 0,9$. Pengujian ini dilakukan dengan memberikan variasi nilai T (*time interval*). Nilai *time interval* dipilih dengan berbagai variasi nilai untuk mengetahui prediksi kesalahan yang dihasilkan. Nilai kesalahan *true Residual* akan menjadi validasi dari hasil prediksi algoritma *exponential smoothing*. Nilai sinyal residual atau f_{true} setara dengan $\hat{f}(k)$ dan algoritma prediksi kesalahan *exponential smoothing* memberikan kesalahan $\hat{f}(k+n)$, $n \geq 1$. Pengujian algoritma prediksi kesalahan sensor ini dilakukan dengan memberikan kesalahan *scaling* pada sensor dengan a berupa sinyal *ramp* dengan kemiringan $-0,005/t$ dan kesalahan dimulai pada detik ke-5 simulasi.

Tabel 4. 1 Hasil Prediksi $\hat{f}(k+1)$ dengan *Time Interval* 1,05s

(t+nT)	f_{true}	$\hat{f}(k+1)$	Error(%)
29,95	-1,8243	-1,8241	0,0116
31	-1,9084	-1,9083	0,0004
32,05	-1,9935	-1,9937	0,0100
33,1	-2,0796	-2,0801	0,0196
34,15	-2,1669	-2,1675	0,0286
35,2	-2,2552	-2,2560	0,0369
36,25	-2,3446	-2,3457	0,0448
37,3	-2,4352	-2,4365	0,0521
38,35	-2,5269	-2,5284	0,0589
39,4	-2,6198	-2,6215	0,0654

Tabel 4. 1 Lanjutan

$(t+nT)$	f_{true}	$\hat{f}(k+1)$	Eror(%)
40,45	-2,7139	-2,7158	0,0715
41,5	-2,8092	-2,8113	0,0772
42,55	-2,9057	-2,9081	0,0826
43,6	-3,0035	-3,0061	0,0877
44,65	-3,1026	-3,1055	0,0926
45,7	-3,2030	-3,2061	0,0972
46,75	-3,3047	-3,3081	0,1015
47,8	-3,4079	-3,4115	0,1057
48,85	-3,5124	-3,5162	0,1096
49,9	-3,6183	-3,6225	0,1134

Tabel 4.1 menunjukkan nilai kesalahan sensor sesungguhnya dan hasil prediksi kesalahan sensor step prediksi $\hat{f}(k+1)$ dengan *time interval* = 1,05s. Nilai f_{true} merupakan nilai kesalahan sesungguhnya pada waktu $t+nT$, sedangkan nilai prediksi $\hat{f}(k+1)$ menunjukkan nilai kesalahan pada 1x*time interval* lebih cepat. Terlihat pada Tabel 4.1, bahwa besar eror prediksi $\hat{f}(k+1)$ sangat minim.

Tabel 4. 2 Hasil Prediksi $\hat{f}(k + 4)$ dengan *Time Interval* 1,05s

$(t+nT)$	f_{true}	$\hat{f}(k+4)$	Eror(%)
29,95	-1,8243	-1,8272	0,1619
31	-1,9084	-1,9115	0,1646
32,05	-1,9935	-1,9968	0,1671
33,1	-2,0796	-2,0832	0,1693
34,15	-2,1669	-2,1706	0,1713
35,2	-2,2552	-2,2591	0,1731
36,25	-2,3446	-2,3487	0,1748
37,3	-2,4352	-2,4395	0,1762
38,35	-2,5269	-2,5314	0,1776

Tabel 4. 2 Lanjutan

$(t+nT)$	f_{true}	$\hat{f}(k+4)$	Eror(%)
39,4	-2,6198	-2,6245	0,1788
40,45	-2,7139	-2,7187	0,1798
41,5	-2,8092	-2,8142	0,1808
42,55	-2,9057	-2,9110	0,1816
43,6	-3,0035	-3,0090	0,1823
44,65	-3,1026	-3,1083	0,1829
45,7	-3,2030	-3,2089	0,1834
46,75	-3,3047	-3,3108	0,1839
47,8	-3,4079	-3,4142	0,1842
48,85	-3,5124	-3,5189	0,1845
49,9	-3,6183	-3,6250	0,1847

Tabel 4.2 menunjukkan nilai kesalahan sensor sesungguhnya dan hasil prediksi kesalahan sensor step prediksi $\hat{f}(k+4)$ dengan *time interval* = 1,05s. Nilai f_{true} merupakan nilai kesalahan sesungguhnya pada waktu $t+nT$, sedangkan nilai prediksi $\hat{f}(k+4)$ menunjukkan nilai kesalahan pada 4*time interval* lebih cepat. Terlihat pada Tabel 4.2, bahwa besar eror prediksi $\hat{f}(k+4)$ sama halnya dengan $\hat{f}(k+1)$ sangatlah minim.

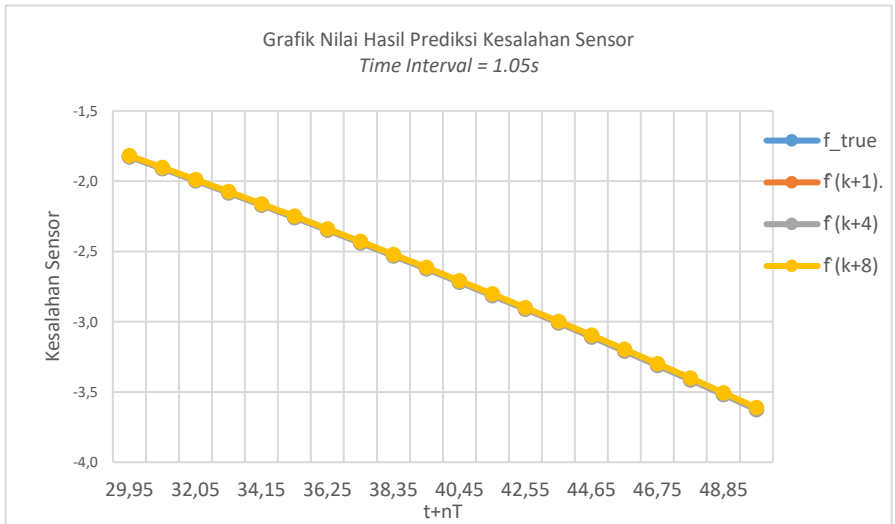
Tabel 4.3 Hasil Prediksi $\hat{f}(k + 8)$ dengan Time Interval 1,05s

$(t+nT)$	f_{true}	$\hat{f}(k+8)$	Eror(%)
29,95	-1,8243	-1,8183	0,3282
31	-1,9084	-1,9023	0,3169
32,05	-1,9935	-1,9874	0,3067
33,1	-2,0796	-2,0735	0,2975
34,15	-2,1669	-2,1606	0,2892
35,2	-2,2552	-2,2488	0,2817
36,25	-2,3446	-2,3382	0,2749
37,3	-2,4352	-2,4286	0,2688

Tabel 4. 3 Lanjutan

$(t+nT)$	f_{true}	$\hat{f}(k+8)$	Eror(%)
38,35	-2,5269	-2,5202	0,2633
39,4	-2,6198	-2,6130	0,2584
40,45	-2,7139	-2,7070	0,2540
41,5	-2,8092	-2,8021	0,2501
42,55	-2,9057	-2,8985	0,2466
43,6	-3,0035	-2,9962	0,2436
44,65	-3,1026	-3,0951	0,2410
45,7	-3,2030	-3,1953	0,2387
46,75	-3,3047	-3,2969	0,2368
47,8	-3,4079	-3,3999	0,2353
48,85	-3,5124	-3,5042	0,2340
49,9	-3,6183	-3,6099	0,2331

Tabel 4.3 menunjukkan nilai kesalahan sensor sesungguhnya dan hasil prediksi kesalahan sensor step prediksi $\hat{f}(k+8)$ dengan *time interval* = 1,05s. Nilai f_{true} merupakan nilai kesalahan sesungguhnya pada waktu $t+nT$, sedangkan nilai prediksi $\hat{f}(k+8)$ menunjukkan nilai kesalahan pada 8*time interval* lebih cepat. Terlihat pada Tabel 4.3 bahwa besar eror prediksi $\hat{f}(k+8)$ sedikit lebih besar dari eror yang dimiliki prediksi $\hat{f}(k+4)$ dan $\hat{f}(k+1)$. Namun secara keseluruhan eror yang dimiliki hasil algoritma *exponential smoothing* dengan *time interval* 1,05s sangatlah minim. Gambar 4. 4 membuktikan hal ini, dimana terlihat bahwa nilai dari prediksi $\hat{f}(k+1)$, $\hat{f}(k+4)$ dan $\hat{f}(k+8)$ tidak memiliki deviasi yang signifikan.



Gambar 4. 4 Grafik Perbandingan Hasil Nilai Prediksi dengan $Time Interval = 1,05s$

Tabel 4. 4 Hasil Prediksi $\hat{f}(k + 1)$ dengan $Time Interval 1,25s$

(t+nT)	f_{true}	$\hat{f}(k+1)$	Error(%)
26,15	-1,4827	-1,5005	1,2061
27,40	-1,5780	-1,5964	1,1646
28,65	-1,6747	-1,6935	1,1275
29,90	-1,7727	-1,7921	1,0940
31,15	-1,8722	-1,8921	1,0637
32,40	-1,9731	-1,9936	1,0362
33,65	-2,0755	-2,0965	1,0111
34,90	-2,1794	-2,2010	0,9881
36,15	-2,2849	-2,3070	0,9670
37,40	-2,3919	-2,4146	0,9476
38,65	-2,5006	-2,5238	0,9296
39,90	-2,6109	-2,6347	0,9130

Tabel 4. 4 Lanjutan

$(t+nT)$	f_{true}	$\hat{f}(k+1)$	Eror(%)
41,15	-2,7229	-2,7473	0,8976
42,40	-2,8366	-2,8617	0,8832
43,65	-2,9521	-2,9778	0,8699
44,90	-3,0694	-3,0957	0,8574
46,15	-3,1886	-3,2155	0,8457
47,40	-3,3096	-3,3373	0,8348
48,65	-3,4326	-3,4609	0,8245
49,90	-3,5576	-3,5866	0,8149

Tabel 4.4 menunjukkan nilai kesalahan sensor sesungguhnya dan hasil prediksi kesalahan sensor step prediksi $\hat{f}(k+1)$ dengan *time interval* = 1,25s. Nilai f_{true} merupakan nilai kesalahan sesungguhnya pada waktu $t+nT$, sedangkan nilai prediksi $\hat{f}(k+1)$ menunjukkan nilai kesalahan pada 1x*time interval* lebih cepat. Terlihat bahwa dengan *time interval* 1,25s besarnya eror yang dimiliki hasil prediksi jauh lebih besar jika dibandingkan hasil prediksi $\hat{f}(k+1)$ dengan *time interval* 1,05s.

Tabel 4. 5 Hasil Prediksi $\hat{f}(k + 4)$ dengan *Time Interval* 1,25s

$(t+nT)$	f_{true}	$\hat{f}(k+4)$	Eror(%)
26,15	-1,4827	-1,5553	4,8957
27,40	-1,5780	-1,6517	4,6735
28,65	-1,6747	-1,7496	4,4743
29,90	-1,7727	-1,8488	4,2946
31,15	-1,8722	-1,9495	4,1317
32,40	-1,9731	-2,0517	3,9835
33,65	-2,0755	-2,1554	3,8481
34,90	-2,1794	-2,2606	3,7239
36,15	-2,2849	-2,3674	3,6096
37,40	-2,3919	-2,4757	3,5040
38,65	-2,5006	-2,5858	3,4063

Tabel 4. 5 Lanjutan

$(t+nT)$	f_{true}	$\hat{f}(k+4)$	Eror(%)
39,90	-2,6109	-2,6975	3,3157
41,15	-2,7229	-2,8109	3,2313
42,40	-2,8366	-2,9260	3,1526
43,65	-2,9521	-3,0430	3,0791
44,90	-3,0694	-3,1618	3,0102
46,15	-3,1886	-3,2825	2,9457
47,40	-3,3096	-3,4051	2,8850
48,65	-3,4326	-3,5297	2,8279
49,90	-3,5576	-3,6563	2,7740

Tabel 4.5 menunjukkan nilai kesalahan sensor sesungguhnya dan hasil prediksi kesalahan sensor step prediksi $\hat{f}(k+4)$ dengan *time interval* = 1,25s. Nilai f_{true} merupakan nilai kesalahan sesungguhnya pada waktu $t+nT$, sedangkan nilai prediksi $\hat{f}(k+4)$ menunjukkan nilai kesalahan pada 4x*time interval* lebih cepat. Terlihat bahwa eror yang dimiliki hasil prediksi jauh lebih besar dari yang dimiliki $\hat{f}(k+4)$ dengan *time interval* 1,05s. Eror prediksi ini juga lebih besar dari hasil prediksi $\hat{f}(k+1)$ dengan *time interval* 1,25s.

Tabel 4. 6 Hasil Prediksi $\hat{f}(k + 8)$ dengan Time Interval 1,25s

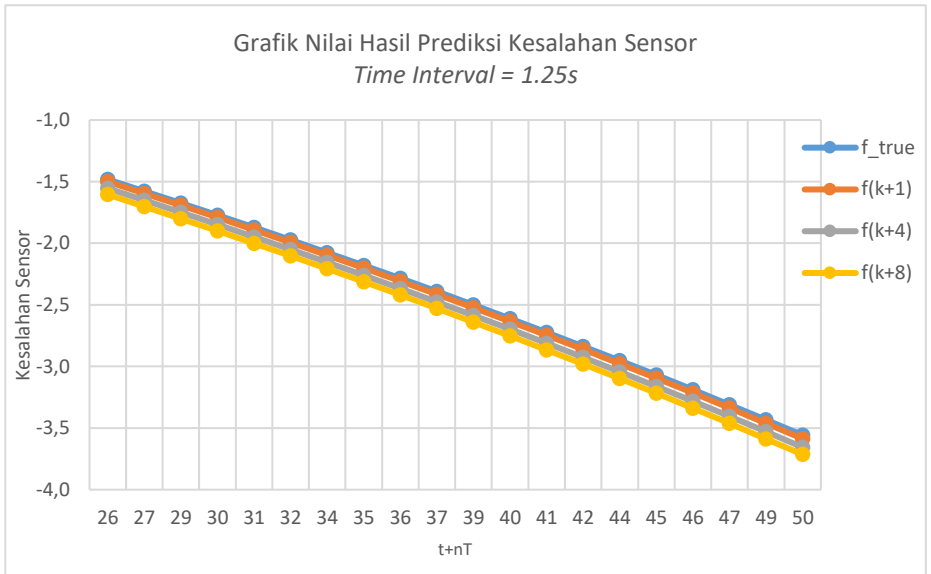
$(t+nT)$	f_{true}	$\hat{f}(k+8)$	Eror(%)
26,15	-1,4827	-1,6049	8,2442
27,40	-1,5780	-1,7018	7,8463
28,65	-1,6747	-1,8001	7,4892
29,90	-1,7727	-1,8998	7,1668
31,15	-1,8722	-2,0009	6,8744
32,40	-1,9731	-2,1035	6,6078
33,65	-2,0755	-2,2076	6,3640
34,90	-2,1794	-2,3132	6,1400
36,15	-2,2849	-2,4205	5,9335

Tabel 4. 6 Lanjutan

$(t+nT)$	f_{true}	$\hat{f}(k+8)$	Eror(%)
37,40	-2,3919	-2,5293	5,7426
38,65	-2,5006	-2,6397	5,5656
39,90	-2,6109	-2,7519	5,4009
41,15	-2,7229	-2,8658	5,2475
42,40	-2,8366	-2,9814	5,1040
43,65	-2,9521	-3,0988	4,9697
44,90	-3,0694	-3,2181	4,8436
46,15	-3,1886	-3,3392	4,7250
47,40	-3,3096	-3,4623	4,6132
48,65	-3,4326	-3,5874	4,5077
49,90	-3,5576	-3,7144	4,4080

Tabel 4.6 menunjukkan nilai kesalahan sensor sesungguhnya dan hasil prediksi kesalahan sensor step prediksi $\hat{f}(k+8)$ dengan *time interval* = 1,25s. Nilai f_{true} merupakan nilai kesalahan sesungguhnya pada waktu $t+nT$, sedangkan nilai prediksi $\hat{f}(k+8)$ menunjukkan nilai kesalahan pada 8*time interval* lebih cepat. Terlihat bahwa eror yang dimiliki $\hat{f}(k+8)$ juga bernilai jauh lebih besar.

Gambar 4.5 membuktikan bahwa dengan penambahan *time interval* menjadi 1,25s, deviasi yang dimiliki hasil prediksi semakin bertambah, terlihat pada gambar 4.5, dimana masing-masing grafik tidak lagi saling bertindih dan mulai memiliki deviasi dari nilai f_{true} . Untuk membuktikan bahwa penambahan *time interval* mengakibatkan penambahan deviasi, dilakukan pengujian dengan peningkatan *time interval* yang lebih besar lagi



Gambar 4. 5 Grafik Perbandingan Hasil Nilai Prediksi dengan
Time Interval = 1,25s

Tabel 4. 7 Hasil Prediksi $\hat{f}(k + 1)$ dengan Time Interval 2s

(t+nT)	f_{true}	$\hat{f}(k+1)$	Error(%)
21	-1,1144	-1,1007	1,2304
23	-1,2593	-1,2456	1,0882
25	-1,4074	-1,3937	0,9736
27	-1,5588	-1,5451	0,8795
29	-1,7137	-1,7000	0,8008
31	-1,8722	-1,8584	0,7343
33	-2,0344	-2,0206	0,6774
35	-2,2004	-2,1866	0,6283
37	-2,3704	-2,3565	0,5855
39	-2,5445	-2,5306	0,5481
41	-2,7229	-2,7089	0,5151

Tabel 4. 7 Lanjutan

$(t+nT)$	f_{true}	$\hat{f}(k+1)$	Eror(%)
43	-2,9057	-2,8916	0,4859
45	-3,0931	-3,0789	0,4600
47	-3,2853	-3,2709	0,4369
49	-3,4824	-3,4679	0,4162
51	-3,6847	-3,6700	0,3978
53	-3,8923	-3,8775	0,3812
55	-4,1055	-4,0904	0,3664
57	-4,3245	-4,3092	0,3531
59	-4,5495	-4,5340	0,3413

Tabel 4.7 menunjukkan nilai kesalahan sensor sesungguhnya dan hasil prediksi kesalahan sensor step prediksi $\hat{f}(k+1)$ dengan *time interval* = 2s. Nilai f_{true} merupakan nilai kesalahan sesungguhnya pada waktu $t+nT$, sedangkan nilai prediksi $\hat{f}(k+1)$ menunjukkan nilai kesalahan pada 1x*time interval* lebih cepat. Terlihat bahwa dengan *time interval* 2s besarnya eror yang dimiliki hasil prediksi masih setara jika dibandingkan hasil prediksi $\hat{f}(k+1)$ dengan *time interval* 1,25s. Hal ini akan berubah ketika besar step prediksi ditambah

Tabel 4.8 Hasil Prediksi $\hat{f}(k + 4)$ dengan *Time Interval* 2s

$(t+nT)$	f_{true}	$\hat{f}(k+4)$	Eror(%)
21	-1,1144	-1,1573	3,8455
23	-1,2593	-1,3026	3,4378
25	-1,4074	-1,4511	3,1056
27	-1,5588	-1,6029	2,8292
29	-1,7137	-1,6029	6,4654
31	-1,8722	-1,7582	6,0894
33	-2,0344	-1,9170	5,7689
35	-2,2004	-2,0795	5,4933
37	-2,3704	-2,2458	5,2547

Tabel 4. 8 Lanjutan

$(t+nT)$	f_{true}	$\hat{f}(k+4)$	Eror(%)
39	-2,5445	-2,4161	5,0469
41	-2,7229	-2,5904	4,8650
43	-2,9057	-2,7690	4,7051
45	-3,0931	-2,9519	4,5642
47	-3,2853	-3,1394	4,4398
49	-3,4824	-3,3316	4,3299
51	-3,6847	-3,5287	4,2327
53	-3,8923	-3,7309	4,1468
55	-4,1055	-3,9384	4,0711
57	-4,3245	-4,1513	4,0046
59	-4,5495	-4,3700	3,9465

Tabel 4.8 menunjukkan nilai kesalahan sensor sesungguhnya dan hasil prediksi kesalahan sensor step prediksi $\hat{f}(k+4)$ dengan *time interval* = 1,25s. Nilai f_{true} merupakan nilai kesalahan sesungguhnya pada waktu $t+nT$, sedangkan nilai prediksi $\hat{f}(k+4)$ menunjukkan nilai kesalahan pada 4*time interval* lebih cepat. Lagi, eror yang dimiliki prediksi $\hat{f}(k+4)$ mengalami peningkatan yang signifikan dari hasil prediksi $\hat{f}(k+1)$ dengan *time interval* yang sama.

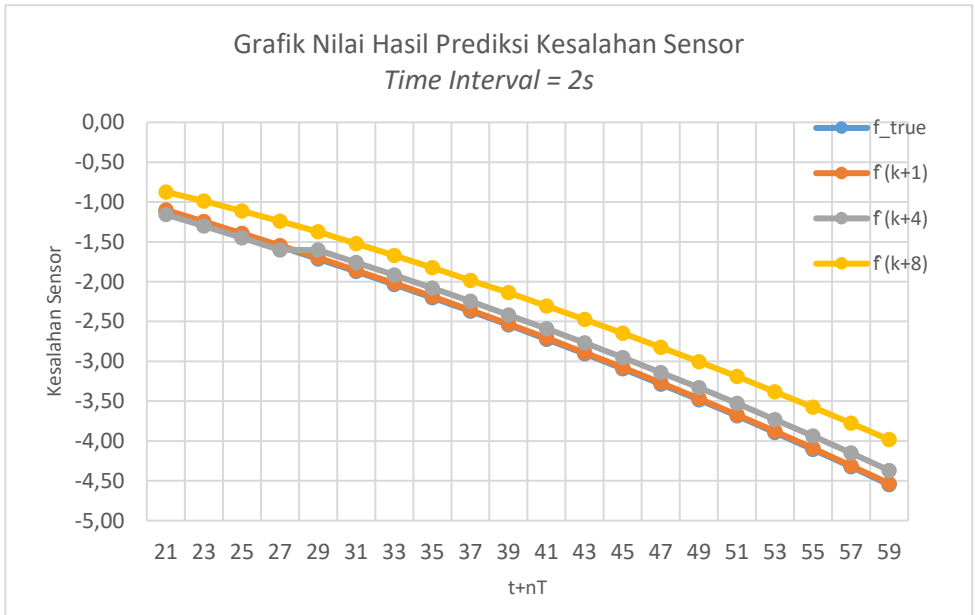
Tabel 4. 9 Hasil Prediksi $\hat{f}(k + 8)$ dengan *Time Interval* 2s

$(t+nT)$	f_{true}	$\hat{f}(k+8)$	Eror(%)
21	-1,1144	-0,8735	21,6180
23	-1,2593	-0,9858	21,7169
25	-1,4074	-1,1137	20,8667
27	-1,5588	-1,2390	20,5165
29	-1,7137	-1,3752	19,7532
31	-1,8722	-1,5221	18,6996
33	-2,0344	-1,6709	17,8667
35	-2,2004	-1,8223	17,1831

Tabel 4. 9 Lanjutan

$(t+nT)$	f_{true}	$\hat{f}(k+8)$	Error(%)
37	-2,3704	-1,9834	16,3260
39	-2,5445	-2,1386	15,9520
41	-2,7229	-2,3040	15,3831
43	-2,9057	-2,4733	14,8819
45	-3,0931	-2,6465	14,4386
47	-3,2853	-2,8238	14,0454
49	-3,4824	-3,0054	13,6962
51	-3,6847	-3,1915	13,3855
53	-3,8923	-3,3820	13,1092
55	-4,1055	-3,5774	12,8635
57	-4,3245	-3,7776	12,6453
59	-4,5495	-3,9830	12,4520

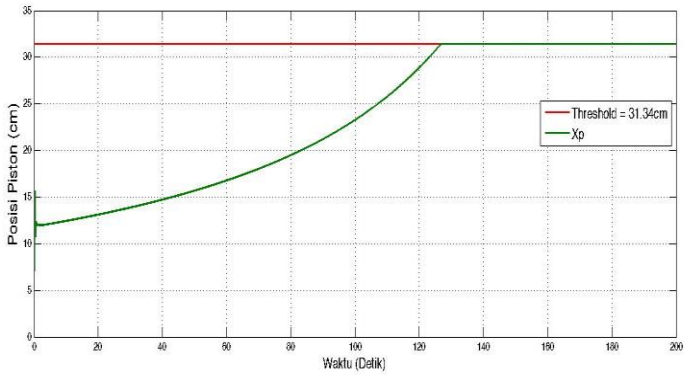
Tabel 4.9 menunjukkan nilai kesalahan sensor sesungguhnya dan hasil prediksi kesalahan sensor step prediksi $\hat{f}(k+8)$ dengan *time interval* = 1,25s. Nilai f_{true} merupakan nilai kesalahan sesungguhnya pada waktu $t+nT$, sedangkan nilai prediksi $\hat{f}(k+8)$ menunjukkan nilai kesalahan pada 8*time interval* lebih cepat. Error prediksi sudah sangat besar memasuki 20% lebih. Berdasarkan tabel 4.9, 4.8 dan 4.7, dapat disimpulkan bahwa penambahan *time interval* mengakibatkan bertambahnya error prediksi. Hal ini juga diperjelas oleh gambar 4.6 yang mana terlihat deviasi dari nilai f_{true} yang semakin membesar dibandingkan gambar 4.5, terutama pada prediksi dengan 8 step. Sehingga untuk pengujian algoritma prediksi keandalan digunakan *time interval* yang memberikan hasil prediksi kesalahan yang terbaik yaitu *time interval* = 1,05s.



Gambar 4. 6 Grafik Perbandingan Hasil Nilai Prediksi dengan *Time Interval = 2s*

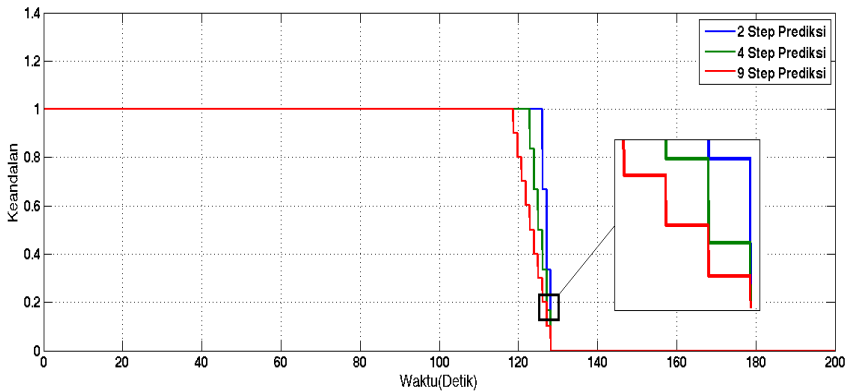
4.4 Hasil Uji Algoritma Prediksi Keandalan Real-time

Algoritma prediksi keandalan *real-time* akan dijalankan dengan menggunakan hasil prediksi kesalahan sensor yang telah diuji sebelumnya. Ditemukan bahwa algoritma prediksi kesalahan sensor bekerja secara optimal dengan *time interval* 1,05s. Hasil uji dari algoritma prediksi keandalan *real-time* dapat digunakan sebagai acuan tingkat “kesehatan” sistem pengendalian posisi aktuator hidrolis ketika terdapat kesalahan *scaling* pada sensor. Pengujian algoritma prediksi keandalan ini akan dilakukan untuk prediksi 2 step, 4 step dan 9 step kedepan. Kesalahan dimulai pada detik ke-5 dengan batas kesalahan atau *threshold* sebesar 31,34cm, yang berarti jika besar posisi piston melebihi 31,34, sistem dikatakan gagal.



Gambar 4.7 Posisi Piston x_p mencapai *threshold*

Pada Gambar 4.7 ditunjukkan bahwa posisi piston sesungguhnya, $x_{p,true}$ mencapai *threshold* pada detik 126,55s.



Gambar 4.8 Grafik keandalan hasil uji algoritma prediksi keandalan *real-time*

Algoritma penghitung keandalan seharusnya dapat memberikan hasil dimana keandalan $R(t) = 0$ disekitar waktu tersebut.

Tabel 4. 10 Hasil uji prediksi keandalan *real-time*

Keandalan	Waktu (Detik)		
	2 Step Prediksi	4 Step Prediksi	9 Step Prediksi
1	0-125,99	0 – 122,84	0-118,64
1–0,75	126-126	122,85-123,89	118,65-120,74
0,75–0,5	126,1-127,04	123,9-124,94	120,75-122,84
0,5–0,25	127,05-128,09	124,95-127,04	122,85-125,99
0,25–0	128,09-128,09	127,05-128,09	126-128,09
0	128,1-200	128,1-200	128,1-200

Pada Gambar 4.8 ditunjukkan grafik keandalan hasil uji algoritma prediksi keandalan *real-time* untuk 2,4 dan 9 step kedepan. Kesalahan dimulai pada detik ke- 5 simulasi, dan sistem terlihat masih dapat mempertahankan nilai posisi piston pada area dibawah batas terlarang untuk waktu yang berbeda masing-masing jumlah step prediksinya. Terlihat pada Gambar 4.8 grafik dengan step prediksi yang lebih banyak akan terlebih dahulu “mengidentifikasi” terjadinya kegagalan dalam sistem dengan penurunan nilai keandalan. Kurva keandalan dengan 9 step prediksi lebih dahulu mengalami penurunan keandalan pada detik 118,64 diikuti oleh keandalan dengan 4 step prediksi pada detik 122,84 dan keandalan dengan 2 step prediksi pada detik 125,99. Dalam hal ini terbukti bahwa algoritma prediksi keandalan dapat digunakan sebagai *tools* untuk melakukan *prognosis* akan tingkat “kesehatan” dari sebuah sistem.

Jika dibandingkan dengan Gambar 4.7, dapat dilihat bahwa algoritma prediksi keandalan tidak dapat memberikan informasi waktu kegagalan sistem atau ketika $R(t) = 0$ yang sesuai dengan keadaan nyata sistem. Hal ini diakibatkan adanya proses diskritisasi dalam algoritma prediksi kesalahan. Diskritisasi ini akan mengakibatkan *delay* dalam penampilan keadaan sistem dalam bentuk keandalan $R(t)$.

Perlu diketahui, dalam perancangan algoritma prediksi keandalan, keandalan dengan step prediksi lebih besar memang mampu memberikan kondisi sistem lebih cepat dibandingkan

dengan step prediksi yang lebih kecil. Namun hal ini jug diiringi dengan penambahan eror prediksi yang terjadi. Besar eror yang terjadi sangat dipengaruhi dalam perancangan algoritma prediksi kesalahan. Oleh karena itu dalam perancangan algoritma prediksi kesalahan, kombinasi yang sesuai antara *time interval* dan banyaknya langkah kedepan n harus diatur agar algoritma dapat memberikan hasil prediksi yang lebih jauh kedepan dengan eror yang minimal.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis data yang telah dilakukan, maka kesimpulan yang diperoleh dari tugas akhir mengenai perancangan sistem monitoring keandalan *real-time* pada aktuator rem hidrolik berbasis *observer* ini adalah :

- *Observer* yang dirancang berdasarkan *augmented state space* dari sistem pengendalian posisi aktuator hidrolik dengan Gain *observer* LQR mampu mengestimasi kesalahan sensor dengan eror *observer* 0.134%.
- Algoritma prediksi kesalahan *Exponential Smoothing* diuji dengan variasi *time interval*. *Time interval* 1,05s memiliki eror rata-rata 0,168%, *time interval* 1,25s memiliki eror rata-rata 3.48%, *time interval* 2s memiliki eror rata-rata 7,17%. Eror prediksi akan bertambah besar dengan penambahan *time interval*
- Algoritma prediksi keandalan real-time berhasil menunjukkan kemampuannya mendeteksi terjadinya kegagalan pada sistem lebih cepat. Pada simulasi dengan 2, 4 dan 9 step prediksi keandalan kedepan, kurva 9 step mendeteksi kegagalan pada sistem 4,2 detik lebih cepat dari 4 step kedepan dan 7,35 detik lebih cepat dari 2 step kedepan.

5.2 Saran

Berdasarkan tugas akhir yang telah dilakukan, terdapat beberapa saran yang dapat diberikan untuk penelitian selanjutnya yaitu sebagai berikut :

- Sistem monitoring keandalan *real-time* pada aktuator hidrolik ini sekiranya dapat dikembangkan dimana penyebab kegagalan dari sistem adalah deteriorasi pada komponen yang terdapat dalam sistem

- Dapat dilakukan studi lebih lanjut mengenai pemodelan kesalahan pada sistem dengan mengacu sumber yang dapat lebih memberikan arti fisis kesalahan yang lebih tepat

DAFTAR PUSTAKA

- An, L., & Sepehri, N. (2003). Hydraulic Actuator Circuit Fault Detection Using Extended Kalman Filter. *Proceeding of the American Control Conference*. Denver.
- Ayurani, L. (2019). *Studi Sistem Monitoring Prediksi Keandalan Real-Time pada Sistem Pengendalian Kecepatan Generator Turbin Angin dengan Kesalahan Sensor*. Surabaya: Institut Teknologi Sepuluh Nopember.
- Badan Pusat Statistik. (n.d.). *Jumlah Kecelakaan, Koban Mati, Luka Berat, Luka Ringan, dan Kerugian Materi yang Diderita Tahun 1992-2017*. Retrieved from Badan Pusat Statistik:
<https://www.bps.go.id/linkTabelDinamis/view/id/1134>
- Balaban, E., Saxena, A., Bansal, P., Goebel, K., & Curran, S. (2009). Modeling, Detection and Disambiguation of Sensor Faults for Aerospace Applications. *IEEE Sensors Journal*, 9(12), 1907-1917.
- Biro Komunikasi dan Informasi Publik Kementerian Perhubungan. (2017, Agustus 22). *Rata-rata Tiga Orang Meninggal Setiap Jam Akibat Kecelakaan Jalan*. Retrieved from Kementerian Komunikasi dan Informatika Republik Indonesia:
https://kominfo.go.id/content/detail/10368/rata-rata-tiga-orang-meninggal-setiap-jam-akibat-kecelakaan-jalan/0/artikel_gpr
- Blischke, W., & Prabhakar, M. (2000). *Reliability : Modeling, Prediction and Optimization*. Canada: John Wiley & Sons, Inc.
- Connor, P. (2012). *Practical Reliability Engineering 5th Edition*. West Sussex: John Wiley & Sons.
- Dhillon, B. (2004). *Reliability, Quality, and Safety for Engineers*. New York: CRC Press.
- Ebeling, C. (1997). *An Introduction to Reliability and Maintainability*. Boston: McGraw-Hill.

- Fu, T. (2000). *Modeling and Performance Analysis of ABS Systems with Non-Linear Control*. Montreal: National Library of Canada.
- Hagenzieker, M., Commandeur, J., & Bijleveld, F. (2014). The History of Road Safety Research: A Quantitative Approach. *Transportation Research*, 150-162.
- Indriawati, K., Sebe, N., Agustinah, T., & Jazidie, A. (2015). Robust Fuzzy Observer-Based Fault Tolerant Tracking Control for Nonlinear Systems with Simultaneous Actuator and Sensor Faults : Application to a DC Series Motor Speed Drive. *International Review of Automatic Control*, 375 - 385.
- Isermann, R. (2006). *Fault-Diagnosis Systems*. Darmstadt: Springer.
- Khan, H., Seraphin, C., & Sepehri, N. (2005). Nonlinear Observer-Based Fault Detection Technique for Electro-Hydraulic Servo Positioning Systems. *Mechatronics*, 15, 1037-1059.
- Li, W. (2010). *ABS Control on Modern Vehicle Equipped with Regenerative Braking*. Delft: Delft Center for Systems and Control (DCSC).
- Lu, H., Kolarik, W., & Lu, S. (2001). Real Time Performance Reliability Prediction. *IEEE Transactions On Reliability*, 353 - 357.
- Ogata, K. (2004). *System Dynamics*. Upper Saddle River: Pearson Prentice Hall.
- Ogata, K. (2010). *Modern Control Engineering 5th Edition*. New Jersey: Prentice Hall.
- Oppenheim, A., & Verghese, G. (2010). Introduction to Communication, Control and Signal Processing. Cambridge: MIT OpenCourseWare.
- Raza, H., Xu, Z., Ioannou, P., & Yang, B. (1995). *Failure Detection of a Computer Controlled Brake System*. Berkeley: University of California.
- Rydberg, K.-E. (2014). *Hydraulic Servo Systems Dynamic Properties and Control*. Linkoping: Department of Management and Engineering Linkoping University.

- Shah, S. (2016). *Design and Implementation of Adaptive Brake Pressure Controller*. Delft: Delft Center for Systems and Control (DCSC).
- Sharifi, S., Tivay, A., Rezaei, S., Zareinejad, M., & Dariani, B. (2017). Leakage Fault Detection in Electro-hydraulic Servo Systems using A Non-Linear Representation Learning Approach. *ISA Transactions*, 1-11.
- Sohl, G., & Bobrow, J. (1999). Experiments and Simulations on The Nonlinear Control of a Hydraulic Servosystem. *IEEE Transactions on Control System Technology*, 7(2), 238-247.
- World Health Organization. (2018). *Global Status Report on Road Safety*. Geneva: World Health Organization.
- Xu, Q.-N., Lee, K.-M., Zhou, H., & Yang, H.-Y. (2016). Model-Based Fault Detection and Isolation Scheme for a Rudder Servo System. *IEEE Transactions on Industrial Electronics*, 62(4), 2384 - 2396.
- Xu, Z., & Yang, B. (1997). *Brake System Analysis, Reliability Testing and Control Using Bench Experiments*. Berkeley: University of California.
- Xu, Z., Ji, Y., & Zhou, D. (2009). A New Real-Time Reliability Prediction Method for Dynamic Systems Based on On-Line Fault Prediction. *IEEE Transactions on Reliability*, 523 - 538.

LAMPIRAN A
SCRIPT SIMULASI SISTEM TURBIN ANGIN PADA
MATLAB R2014a

```

clear
%% Piston Motion
m = 100;           %Massa beban piston
k = 2200;         %Konstanta pegas
Bp = 6.9*10^5*10^3; %Modulus bulk
V10 = 155;       %Volume ruang utama
V20 = 83;        %Volume ruang kembali
A1 = 6.8;        %Luas penampang ruang utama
A2 = 9.6;        %Luas penampang ruang
kembali

%%Spool Valve Dynamics
Ps = 6895*10^3;   %Tekanan suplai
Pr = 100*10^3;   %Tekanan reservoir

%% State Space
sscA = [0 0 0 -(Bp*A1)/V10 0;
        0 0 0 (Bp*A2)/V20 0;
        0 0 0 1 0;
        A1/m -A2/m -(3/2*k)/m 0 0;
        0 0 0 0 -10 ];
sscB = [0;
        0;
        0;
        0;
        10];
sscC = [0 0 1 0 0];
sscG = [1 0;
        0 1;
        0 0;
        0 0;
        0 0];
sscFs = [1];

lengthA = size(sscA,2);
widthC = size(sscC,1);
lengthB = size(sscB,2);

```

```

% Extended State Space
Az = eye(widthC)*10;
barA = [sscA zeros(lengthA,widthC);
        Az*sscC -Az];
barB = [sscB;
        zeros(widthC,lengthB)];
barC = [zeros(widthC,lengthA) eye(widthC)];
barF = [zeros(lengthA,lengthB);
        Az*sscFs];
barG = [sscG;
        zeros(1,2)];
lengthF = size(barF,2);

```

```

%% Persamaan observer

```

```

curlA = [barA barF;
         zeros(lengthF,lengthA+widthC)
         eye(lengthF)];
curlB = [barB;
         zeros(lengthF,lengthB)];
curlC = [barC zeros(widthC,lengthF)];
curlG = [barG;
         zeros(1,2)];

```

```

%LQR Observer Gain

```

```

R = eye(1);
Q = [1 0 0 0 0 0 0 ;
     0 1 0 0 0 0 0 ;
     0 0 0 0 0 0 0 ;
     0 0 0 1 0 0 0 ;
     0 0 0 0 1 0 0 ;
     0 0 0 0 0 1 0 ;
     0 0 0 0 0 0 9e8 ];
Klqr = lqr(curlA',curlC',Q,R)

```

```

%% Exponential Smoothing

```

```

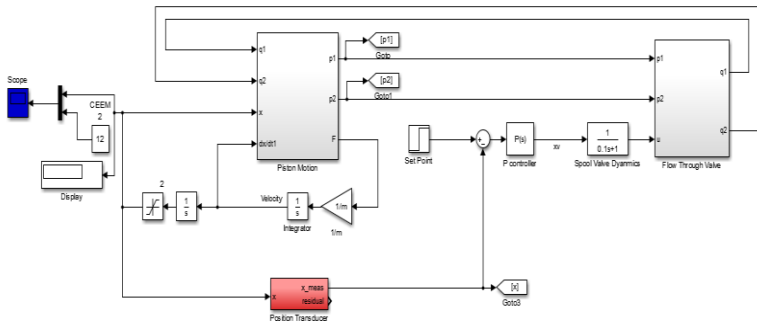
a = 0.9;
b = 0.9;

```

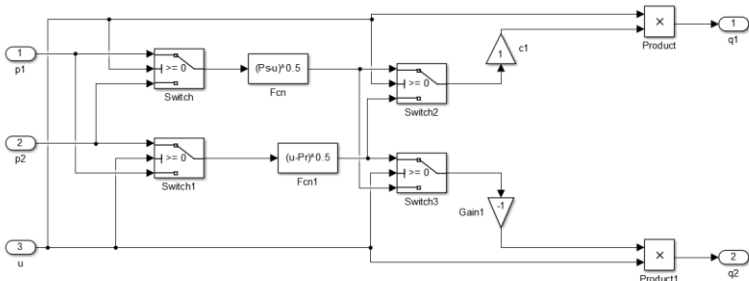
$T_p = 1,05;$
 $T = 0.01;$

LAMPIRAN B
SIMULINK SISTEM MONITORING DAN PREDIKSI
KEANDALAN *REAL-TIME* PADA SISTEM
PENGENDALIAN AKTUATOR HIDROLIK

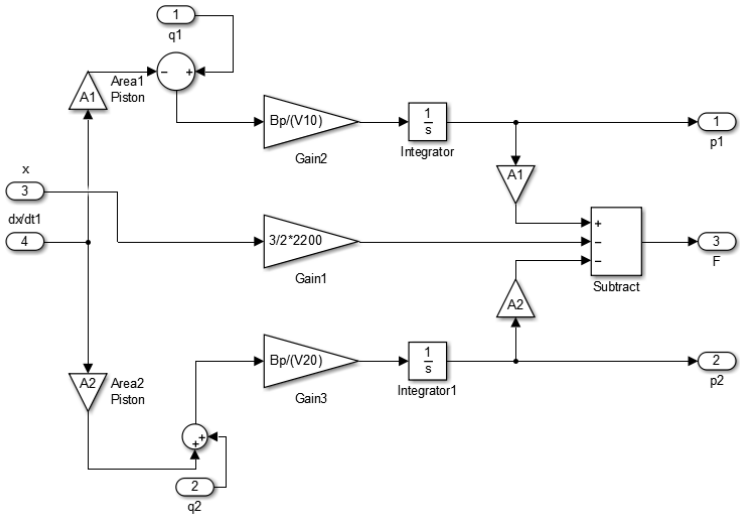
B.1 Sistem Pengendalian Posisi Aktuator Hidrolik



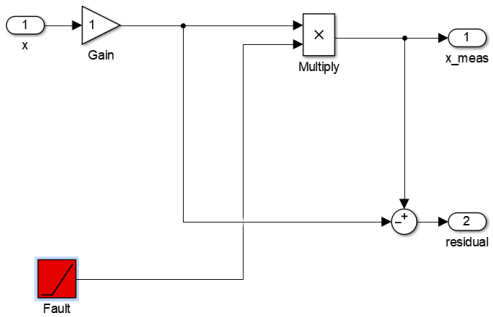
B.2 Model Besar Aliran Fluida melalui Spool Valve



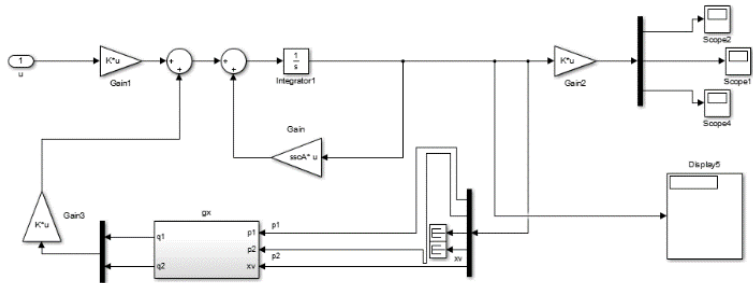
B.3 Model Gerak Piston



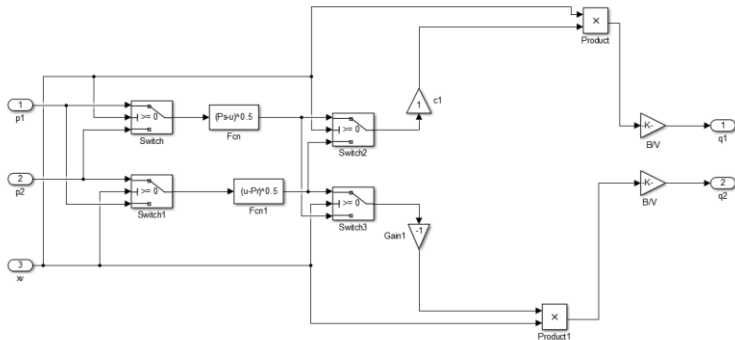
B.4 Model Sensor Posisi



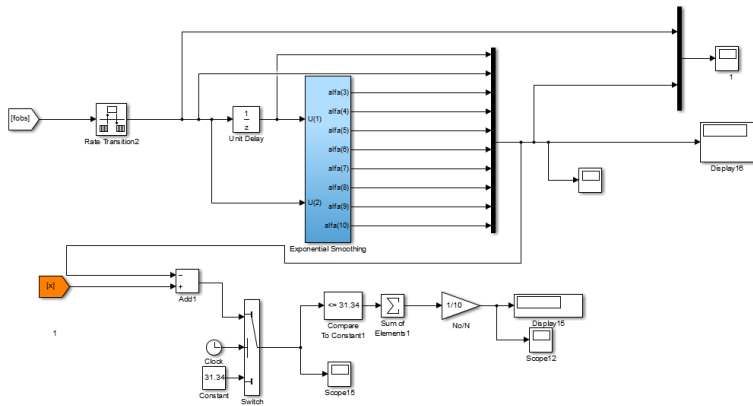
B.5 State Space System dengan Bagian Non-Linear Gx



B.6 Bagian Non-Linear Gx



B.9 Algorithma *Reliability Prediction*



BIODATA PENULIS



Nama penulis Rifyan Alana dilahirkan di Pekanbaru, tanggal 10 Januari 1998. Saat ini penulis tinggal di Bumi Marina Emas III A-70, Surabaya, Provinsi Jawa Timur. Penulis telah menyelesaikan pendidikan di SD Islam As-Shofa, SMP Islam As-Shofa, SMA Islam As-Shofa dan sedang menempuh pendidikan S1 Teknik Fisika FTI di Institut Teknologi

Sepuluh Nopember Surabaya hingga sekarang.

Pada bulan Mei 2019 penulis telah menyelesaikan Tugas Akhir dengan judul **Perancangan Sistem Monitoring Prediksi Keandalan *Real-Time* pada Aktuator Hidrolik Berbasis Estimasi Kesalahan Menggunakan Observer**. Bagi pembaca yang memiliki kritik, saran atau ingin berdiskusi lebih lanjut mengenai tugas akhir ini, maka dapat menghubungi penulis melalui *email*: rifyanalana@gmail.com .