

**TUGAS AKHIR - TF 181801**

**PREDIKSI KEANDALAN *REAL-TIME* PADA  
SISTEM PENGENDALIAN *LEVEL SEPARATOR*  
BERDASARKAN PREDIKSI KESALAHAN SENSOR**

**ALOYS YANTRI MARTINA**  
NRP. 0231174500007

Dosen Pembimbing  
Ir. Ya'umar, M.T.

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

*“Halaman ini sengaja dikosongkan”*



FINAL PROJECT - TF 181801

***REAL-TIME RELIABILITY PREDICTION AT  
SEPARATOR LEVEL CONTROL SYSTEM  
BASED ON SENSOR FAULT PREDICTION***

ALOYS YANTRI MARTINA  
NRP. 0231174500007

Supervisor  
Ir. Ya'umar, M.T.

DEPARTMENT OF ENGINEERING PHYSICS  
Faculty of Industrial Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019

*“Halaman ini sengaja dikosongkan”*

## PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini

Nama : Aloys Yantri Martina  
NRP : 02311745000007  
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 “Prediksi Keandalan *Real-Time* Pada Sistem Pengendalian *Level Separator* Berdasarkan Prediksi Kesalahan Sensor” 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 surat pernyataan ini saya buat dengan sebenar-benarnya.

Surabaya, 25 Juli 2019  
Yang membuat pernyataan



Aloys Yantri Martina  
NRP.02311745000007

*“Halaman ini sengaja dikosongkan”*

**LEMBAR PENGESAHAN I  
TUGAS AKHIR**

**PREDIKSI KEANDALAN *REAL-TIME* PADA SISTEM  
PENGENDALIAN *LEVEL SEPARATOR* BERDASARKAN  
PREDIKSI KESALAHAN SENSOR**

Oleh:

Aloys Yantri Martina  
NRP. 0231174500007

Surabaya, 25 Juli 2019

Menyetujui,  
Dosen Pembimbing



Ir. Yaumar, M.T.  
NIP. 19540406 198103 1 003

Mengetahui,  
Kepala Departemen  
Teknik Fisika FTI-ITS



S.T., M.Si, Ph.D.  
NIP. 19780907 200312 1 002

*“Halaman ini sengaja dikosongkan”*



**PREDIKSI KEANDALAN *REAL-TIME* PADA SISTEM  
PENGENDALIAN *LEVEL* SEPARATOR BERDASARKAN  
PREDIKSI KESALAHAN SENSOR**

**TUGAS AKHIR**

Diajukan Untuk Memenuhi Salah Satu Syarat Memperoleh  
Gelar Sarjana Teknik  
pada  
Program Studi S-1 Departemen Teknik Fisika  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember

Oleh:

**ALOYS YANTRI MARTINA**  
**NRP.0231174500007**

Disetujui oleh Tim Penguji Tugas Akhir:

1. Ir. Ya'umar, M.T. .... (Pembimbing)
3. Dr. Bambang L. Widjiantoro, S.T., M.T. .... (Ketua Penguji)
2. Dr. Katherin Indriawati, S.T., M.T. .... (Penguji I)
3. Ir. Wiratno Argo Asmoro, Msc. .... (Penguji II)

**SURABAYA**  
**Juli, 2019**

*“Halaman ini sengaja dikosongkan”*

**PREDIKSI KEANDALAN *REAL-TIME* PADA SISTEM  
PENGENDALIAN *LEVEL* SEPARATOR BERDASARKAN  
PREDIKSI KESALAHAN SENSOR**

**Nama** : Aloys Yantri Martina  
**NRP** : 0231174500007  
**Departemen** : Teknik Fisika FTI-ITS  
**Dosen Pembimbing** : Ir. Ya'umar, M.T.

**Abstrak**

*Separator* merupakan salah satu *plant* utama dalam proses pengolahan minyak dan gas (migas) di Saka Indonesia Pangkah Limited (SIPL). Kinerja dari *separator* dipengaruhi oleh *level* fluida pada *separator*, sehingga pada *separator* diperlukan suatu sistem pengendalian *level* untuk menjaga kemurnian fluida. Salah satu penyebab terjadinya kegagalan pada sistem *separator* dapat disebabkan oleh kesalahan pengukuran sensor. Pada makalah ini dirancang sistem prediksi keandalan *real-time* pada sistem pengendalian *level* pada separator berdasarkan prediksi kesalahan sensor. Bentuk kesalahan sensor yang terjadi pada sistem ini yaitu *loss of sensitivity*. Adapun langkah utama yang dilakukan dalam perancangan sistem ini adalah perancangan *observer* untuk mengestimasi kesalahan sensor, perancangan algoritma *exponential smoothing* untuk memprediksi kesalahan dan perancangan algoritma *reliability prediction* untuk menghitung keandalan. Berdasarkan hasil simulasi didapatkan, *observer* yang telah dirancang dengan menggunakan *Linear Matrix Inequality* (LMI) untuk menentukan *gain*-nya dapat dengan baik mengestimasi nilai kesalahan sensor dengan nilai *error* sebesar 0,297%. Algoritma *exponential smoothing* dapat bekerja dengan optimal dengan menggunakan *time interval* 1s. Pada uji *reliability prediction* ditemukan bahwa jumlah prediksi mempengaruhi ketelitian prediksi keandalan *real-time* yang dihasilkan, semakin besar jumlah prediksi maka semakin teliti. Jumlah prediksi 10 menghasilkan prediksi keandalan *real-time* lebih baik dibandingkan jumlah prediksi sebesar dari 3, 5 dan 7.

**Kata kunci:** *Exponential smoothing*, kesalahan sensor, *observer*, prediksi keandalan, *separator*

*“Halaman ini sengaja dikosongkan”*

**REAL-TIME RELIABILITY PREDICTION AT SEPARATOR  
LEVEL CONTROL SYSTEM BASED ON SENSOR FAULT  
PREDICTION**

**Name** : Aloys Yantri Martina  
**NRP** : 0231174500007  
**Department** : Teknik Fisika FTI-ITS  
**Supervisors** : Ir. Ya'umar, M.T.

**Abstract**

Separator is one of the main plants in the processing of oil and gas (oil and gas) at Saka Indonesia Pangkah Limited (SIPL). The performance of the separator is influenced by the fluid level in the separator, so that at the separator a level control system is needed to maintain fluid purity. One of the causes of failure in the separator system can be caused by a sensor measurement error. In this paper a real-time reliability prediction system is designed on the level control system on the separator based on sensor error prediction. The form of sensor error that occurs in this system is loss of sensitivity. The main step taken in designing this system is designing the observer to estimate sensor errors, designing exponential smoothing algorithms to predict errors and designing reliability prediction algorithms to calculate reliability. Based on the simulation results obtained, observers who have been designed using the Linear Matrix Inequality (LMI) to determine the gain can well estimate the value of sensor errors with an error value of 0.297%. The exponential smoothing algorithm can work optimally by using time interval 1s. In the reliability prediction test it was found that the number of predictions affected the accuracy of the real-time reliability predictions produced, the greater the number of predictions, the more thorough. Prediction number 10 produces better predictions of real-time reliability than the number of predictions of 3, 5 and 7.

**Keyword:** Exponential smoothing, sensor fault, observers, reliability predictions, sensors, separator.

*“Halaman ini sengaja dikosongkan”*

## KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa yang senantiasa melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat terselesaikannya Tugas Akhir yang berjudul **“Prediksi Keandalan Real-Time Pada Sistem Pengendalian Level Separator Berdasarkan Prediksi Kesalahan Sensor”**.

Dalam pengerjaan Tugas Akhir ini, penulis telah banyak memperoleh bantuan dari berbagai pihak. Untuk itu, penulis mengucapkan terima kasih kepada :

1. Orang tua dan saudara penulis, senantiasa tiada hentinya memberikan doa, semangat, serta motivasi kepada penulis dalam pengerjaan tugas akhir.
2. Ir. Ya'umar, M.T. selaku dosen pembimbing yang telah memberikan ilmu, nasihat serta bimbingan yang sangat bermanfaat.
3. Ibu Dr. Katherin Indriawati, S.T., M.T. selaku dosen pembimbing yang juga telah dengan sabar memberikan petunjuk, ilmu serta bimbingan yang sangat bermanfaat.
4. Wahyu Dwi Ignas Putra, Johanna Mei Ekawaty dan Kanisius Catur Christian, yang senantiasa tiada hentinya memberikan doa, semangat, serta motivasi kepada penulis dalam pengerjaan tugas akhir ini.
5. Nur Ummama Sofyana teman seperjuangan dalam pengerjaan tugas akhir ini, terimakasih atas dukungan dan kerjasamanya.
7. Teman-teman TAWan Laboratorium Fisika Rekayasa yang selalu memberi semangat dan dukungannya
9. Seluruh dosen karyawan dan civitas akademik Teknik Fisika, terimakasih atas segala bantuan dan kerjasamanya.
10. Teman-teman Lintas Jalur Departemen Teknik Fisika angkatan 2017 serta semua pihak yang tidak dapat penulis sebutkan satu persatu, terimakasih atas dukungan penuh kepada penulis dalam penyelesaian tugas akhir ini.

Penulis sadar bahwa penulisan laporan tugas akhir ini masih terdapat kekurangan. Untuk itu, penulis memohon maaf atas segala kekurangan dalam buku tugas akhir ini. Semoga tugas akhir ini dapat memberikan manfaat dalam pengembangan ilmu pengetahuan dan teknologi.

Surabaya, 25 Juli 2019

Penulis



## DAFTAR ISI

HALAMAN JUDUL .....	i
<i>COVER PAGE</i> .....	iii
PERNYATAAN BEBAS PLAGIARISME .....	v
LEMBAR PENGESAHAN I .....	vii
ABSTRAK .....	xi
<i>ABSTRACT</i> .....	xiii
KATA PENGANTAR.....	xv
DAFTAR ISI.....	xvii
DAFTAR GAMBAR .....	xix
DAFTAR TABEL .....	xxi
DAFTAR NOTASI .....	xxiii
BAB I. PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan .....	3
1.4 Batasan Masalah .....	3
BAB II. TEORI PENUNJANG .....	5
2.1 <i>Separator</i> .....	5
2.2 Sistem Kontrol <i>Separator</i> .....	7
2.3 Pemodelan Dinamik <i>Separator</i> .....	8
2.4 <i>Level Transmitter</i> .....	9
2.5 <i>Control Valve</i> .....	11
2.6 Kesalahan Sensor .....	12
2.7 <i>Observer</i> .....	13
2.8 Algoritma <i>Exponential Smoothing</i> Untuk Prediksi Kesalahan.....	17
2.9 Keandalan .....	18
2.10 Prediksi Keandalan <i>Real-Time</i> .....	19
BAB III. METODOLOGI PENELITIAN .....	21
3.1 Studi Literatur .....	22
3.2 Pemodelan Sistem.....	22
3.3 Perancangan Sistem Kendali PI.....	27
3.4 Perancangan <i>Observer</i> .....	28

3.5	Pengujian Algoritma <i>Observer</i> .....	34
3.6	Perancangan Algoritma Prediksi Kesalahan .....	34
3.7	Pengujian Algoritma Prediksi Kesalahan Sensor.....	35
3.8	Perancangan Algoritma Prediksi Keandalan <i>Real-Time</i> .....	36
3.9	Pengujian Algoritma Prediksi Keandalan <i>Real-time</i> ...	38
BAB IV	ANALISA DATA DAN PEMBAHASAN.....	39
4.1	Hasil Uji <i>Observer</i> .....	39
4.2	Hasil Uji Algoritma Prediksi Kesalahan Sensor .....	41
4.3	Hasil Uji Algoritma Prediksi Keandalan <i>Real-time</i> .....	46
BAB V	PENUTUP .....	49
5.1	Kesimpulan .....	49
5.2	Saran .....	49
DAFTAR PUSTAKA		
LAMPIRAN		
BIODATA PENULIS		

## DAFTAR GAMBAR

Gambar 2.1	<i>Separator</i> Dua Fasa (P&ID <i>Production separator</i> 132-V-02 Saka Indonesia Pangkah Limited.....	6
Gambar 2.2	Diagram Blok Sistem Pengendalian <i>Level Separator</i> .....	7
Gambar 2.3	LDT ( <i>Differential Pressure Transmitter</i> ) .....	10
Gambar 2.4	ATC dan ATO <i>Control Valve</i> .....	11
Gambar 2.5	<i>Observer State</i> .....	14
Gambar 3.1	Diagram Alir Penelitian Tugas Akhir .....	21
Gambar 3.2	Respon <i>Open Loop</i> .....	27
Gambar 3.3	Respon <i>Closed Loop</i> .....	28
Gambar 3.4	Diagram blok sistem pengendalian yang telah terintegrasi dengan <i>observer</i> .....	34
Gambar 3.5	Diagram Blok Sistem Prediksi Keandalan <i>Real –Time</i> .....	37
Gambar 4.1	Perbandingan Hasil Estimasi degan Kesalahan Sensor (Rill) .....	39
Gambar 4.2	Perbandingan Hasil Estimasi dengan Kesalahan Sensor (perbesaran gambar 4.1) .....	40
Gambar 4.3	Grafik Hasil Algoritma Prediksi Kesalahan Sensor dengan <i>Time Interval</i> 1s .....	41
Gambar 4.4	Grafik Hasil Algoritma Prediksi Kesalahan Sensor <i>Time Interval</i> 2s.....	43
Gambar 4.5	Grafik Hasil Algoritma Prediksi Kesalahan Sensor <i>Time Interval</i> 5s.....	45
Gambar 4.6	Grafik Hasil Uji Algoritma Prediksi Keandalan <i>Real-Time</i> dengan <i>Time Interval</i> 1s .....	47

*“Halaman ini sengaja dikosongkan”*

## DAFTAR TABEL

Tabel 3.1	Data Spesifikasi <i>Separator</i> .....	22
Tabel 4.1	Perbandingan Hasil Estimasi dengan Kesalahan Sensor .....	40
Tabel 4.2	Hasil Prediksi dengan <i>Time Interval</i> 1s.....	42
Tabel 4.3	Hasil Prediksi dengan <i>Time Interval</i> 2s.....	43
Tabel 4.4	Hasil Prediksi dengan <i>Time Interval</i> 2s (Lanjutan)...	44
Tabel 4.5	Hasil Prediksi dengan <i>Time Interval</i> 5s.....	45
Tabel 4.6	Hasil Prediksi dengan <i>Time Interval</i> 5s (Lanjutan)...	46
Tabel 4.7	Hasil Uji Prediksi Keandalan <i>Real-Time</i> .....	47

*“Halaman ini sengaja dikosongkan”*

## DAFTAR NOTASI

$A$	=	<i>Luas penampang separator (<math>m^2</math>)</i>
$H$	=	<i>Level (<math>m</math>)</i>
$Q$	=	<i>Flowrate (<math>m^3</math>)</i>
$\rho$	=	<i>Massa jenis (<math>kg/m^3</math>)</i>
$D$	=	<i>Diameter separator (<math>m</math>)</i>
$L$	=	<i>Tinggi tangki (<math>m</math>)</i>
$l$	=	<i>Step prediksi</i>
$A$	=	<i>Matriks keadaan</i>
$B$	=	<i>Matriks kendali</i>
$C$	=	<i>Matriks keluaran</i>
$F_a$	=	<i>Matriks kesalahan aktuator</i>
$F_s$	=	<i>Matriks kesalahan sensor</i>
$G(x)$	=	<i>Matriks 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>
$\hat{\hat{x}}$	=	<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>
$\mathcal{M}_0$	=	<i>Batas nilai variabel</i>

*"Halaman ini sengaja dikosongkan"*





# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Saka Indonesia Pangkah Limited (SIPL) merupakan salah satu perusahaan yang bergerak dalam bidang industri minyak dan gas yang beroperasi di 2 tempat yaitu *offshore* yang berada di Ujung Pangkah dan *onshore* yang berada di Kawasan Industri Maspion Manyar, Kabupaten Gresik Jawa Timur. Salah satu unit utama yang berperan penting dalam proses produksi minyak di Saka Indonesia Pangkah Limited (SIPL) adalah *separator*.

Pada proses pengolahan minyak dan gas di SIPL terdapat *separator* dua fasa yang berfungsi untuk memisahkan gas murni dan liquid. Adapun prinsip kerja dari *separator* yaitu dengan gaya pemisahan fluida berdasarkan massa jenis (densitas) dan gravitasi, dimana fluida dengan densitas lebih berat akan berada dibagian bawah sedangkan fluida dengan densitas lebih ringan akan berada pada bagian atas (Rubiandani, 2001).

Terjadinya kegagalan fungsi dari suatu komponen instrument pada sistem pengendalian *level separator* akan menyebabkan kinerja *separator* menjadi tidak efisien. Kegagalan fungsi pada instrument juga akan menghambat kegiatan produksi minyak jika terjadi *trip/shutdown*. Semakin besar peluang sistem mengalami kegagalan maka semakin rendah keandalan sistem tersebut. Sementara itu, nilai keandalan merupakan salah satu aspek yang dapat mempengaruhi keberhasilan proses produksi. Salah satu penyebab terjadinya kegagalan pada sistem pengendalian *level separator* adalah kesalahan yang terjadi pada sensor.

Prediksi keandalan *real-time* perlu diterapkan untuk sistem pengendalian *level* pada *separator*, sehingga kegagalan yang akan terjadi pada *separator* dapat dideteksi lebih dini. Dengan dilakukannya prediksi keandalan dapat meningkatkan umur peralatan, mengurangi waktu *downtime* sehingga kerusakan yang lebih parah dapat dihindari.

Pada penelitian sebelumnya (Zhengguo Xu, 2009) telah memperkenalkan sebuah metode baru untuk memonitoring

keandalan *real-time* untuk sistem dinamik yang didasarkan pada algoritma prediksi kesalahan *on-line*. Faktor yang dapat mengurangi keandalan pada sebuah sistem dimodelkan sebagai sebuah masukan kesalahan tambahan pada sistem, dan kesalahan diasumsikan dengan waktu linear. Kesalahan terhadap waktu terlebih dahulu diestimasi berdasarkan pada modifikasi *particle filtering algorithm*. Kemudian, sebagai sebuah *time-series*, urutan estimasi kesalahan diatur dan diprediksi oleh metode *exponential smoothing*. Analisis matematika menunjukkan bahwa efek pada sistem dan gangguan pengukuran pada estimasi kesalahan dapat berkurang secara signifikan oleh *exponential smoothing*, yang mana mengindikasikan bahwa perbandingan akurasi tinggi pada estimasi kesalahan dan prediksi dapat dijamin. Berdasarkan pada *particle filtering* dan hasil prediksi kesalahan, keseluruhan prediksi keandalan sistem dihitung melalui strategi simulasi Monte Carlo.

Algoritma *monitoring* keandalan *real-time* dapat dinyatakan dalam sebuah *observer*. Sebelumnya (Fitria, 2018) telah melakukan penelitian mengenai penggunaan *observer* menghasilkan estimasi kesalahan pada pembuatan algoritma *monitoring* keandalan secara *real-time*. Hasil dari penelitian ini berupa data *monitoring* yang menunjukkan bahwa semakin besar nilai kesalahan sensor, maka semakin cepat sistem akan mengalami kegagalan. Namun pada penelitian ini, *observer* hanya mampu melakukan estimasi untuk satu *step* ke depan sehingga hanya didapatkan estimasi kesalahan untuk satu *step* ke depan.

Oleh karena itu, pada penelitian ini dilakukan perancangan prediksi keandalan *real-time* pada sistem pengendalian *level separator* berdasarkan prediksi kesalahan sensor yang diharapkan mampu menghasilkan nilai keandalan *real-time* yang akurat. Dengan diterapkannya prediksi keandalan *real-time* pada sistem pengendalian *level separator* diharapkan dapat meningkatkan umur peralatan, mengurangi *downtime* dan menghindari kerusakan yang lebih parah pada sistem *separator* sehingga dapat mencegah kerugian dan menghemat *maintenance-cost*.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, maka permasalahan yang terdapat dalam tugas akhir ini, yaitu:

- Bagaimana merancang *observer* yang dapat digunakan untuk mendapatkan estimasi kesalahan pada sensor pada sistem pengendalian *level separator* ?
- Bagaimana merancang prediksi kesalahan sensor pada sistem pengendalian *level separator* menggunakan algoritma *Exponential Smoothing* ?
- Bagaimana cara menghitung keandalan sistem pengendalian *level separator* 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 *level separator* yang mampu mengestimasi kesalahan.
- Memprediksi nilai kesalahan pada sistem pengendalian *level separator* menggunakan algoritma *Exponential Smoothing*.
- Menghitung keandalan sistem pengendalian *level separator* secara *real-time* menggunakan algoritma *Reliability Prediction* (RP).

## 1.4. Batasan Masalah

Adapun batasan masalah dalam penelitian ini antara lain:

- *Fault* pada pengendalian *level separator* terjadi pada sensor.
- Kesalahan yang ditinjau pada sensor berupa kesalahan sensitivitas.
- *Controller* yang terpasang pada *separator* berupa pengendalian PI.
- Pemodelan plant hanya meninjau pada kesetimbangan massa di *separator*.

- Keandalan yang dimaksud dalam penelitian ini adalah nilai keandalan oleh algoritma RP, bukan keandalan yang diperoleh dari data *maintenance*.
- Sistem dikatakan gagal ketika tinggi *level* telah melebihi batas maksimum *level* separator yaitu 2,025 m.
- Simulasi dilakukan pada Matlab R2014b.

## BAB II TEORI PENUNJANG

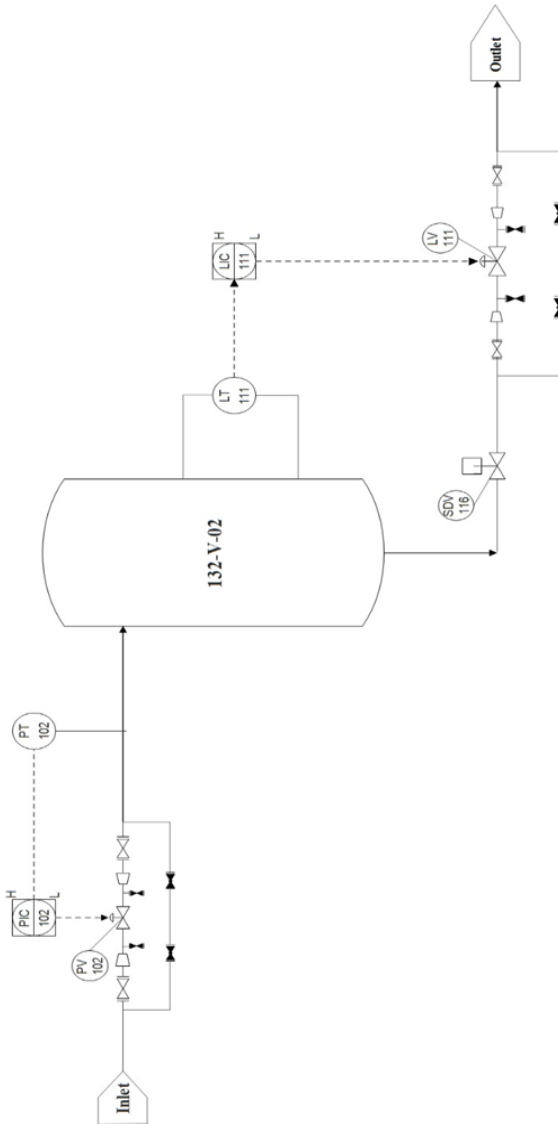
### 2.1 *Separator*

*Separator* adalah bejana bertekanan yang digunakan untuk memisahkan fluida produksi kedalam fasa liquid dan fasa gas. Berdasarkan konfigurasi tangkinya, *separator* dibagi menjadi *vertical separator* dan *horizontal separator*. Berdasarkan tujuan separasinya, *separator* dibagi menjadi *separator* 2 fasa (liquid dan gas), dan *separator* 3 fasa (air,minyak, dan gas) (Prapdito, 2011). *Production separator* 132-V-02 di SAKA merupakan *separator vertical* dan memisahkan 2 fasa fluida (air dan gas).

Adapun prinsip kerja dari *separator tegak/vertical* yaitu fluida yang berasal dari *manifold* akan masuk ke *separator* melalui lubang inlet dan selanjutnya akan menabrak *inlet diverter*. Pengubah-arahan aliran masuk (*inlet diverter*) akan menyebabkan cairan yang masuk menyinggung dinding *separator* dalam bentuk film, dan pada saat yang bersamaan memberikan gerakan *centrifugal* kepada fluida. Ini memberikan pengurangan momentum yang diinginkan dan mengizinkan gas untuk keluar dari film cairan. Gasnya naik kebagian atas dari bejana, dan cairannya turun ke bawah. (Prasetyo, 2016)

Sistem pengendalian *level liquid* pada *separator* dilakukan untuk menjaga ketinggian *level liquid* didalam *separator* tetap berada daerah yang di izinkan jika *level liquid* pada *separator* memiliki ketinggian yang melebihi ketinggian yang dizinkan maka *liquid* akan tercampur dengan gas,tetapi pada saat *liquid* berada di *level* yang terendah maka gas akan tercampur ke *liquid*.

Berikut ini P&ID *Production Separator* 132-V-02 Saka Indonesia Pangkah Limited.

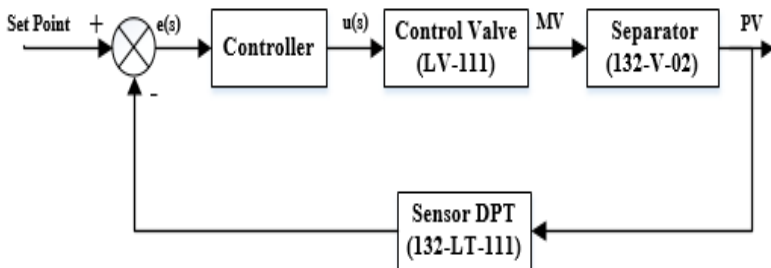


**Gambar 2.1** *Separator* Dua Fasa (P&ID *Production Separator* 132-V-02 Saka Indonesia Pangkah Limited)

## 2.2 Sistem Kontrol Separator

Sistem kontrol pada pengendalian *level pada production separator 132-V-02* Saka Indonesia Pangkah Limited menjaga nilai *level* sebagai variabel proses agar fluida yang tercampur dapat di pisahkan. Pada sistem pengendalian *level* pada tangki *separator*, menggunakan sistem kontrol *proportional integral*. PI (*Proportional Integral*) merupakan gabungan dari dua unit pengendali yaitu P dan I, maka kelebihan dan kekurangan yang ada pada pengendali P dan pengendali I juga ada pada pengendali PI.

Pengendalian *level* yang dimaksudkan disini adalah menjaga menjaga *level* atau ketinggian fluida pada tangki *separator* tetap pada *set point* yaitu pada ketinggian 1,425 meter. Jika *level* fluida yang terukur lebih besar dari *set point* maka *control valve* akan membuka lebih besar untuk menjaga *level* pada tangki kembali ke *setpoint* dan sebaliknya jika *level* yang terukur kurang dari *set point* maka *control valve* akan mengurangi bukaan atau menutup. Kondisi *control valve open* atau *close* terjadi sampai fluida kembali stabil sehingga kondisi *level* tetap terjaga pada nilai *set point*. Berikut adalah diagram blok sistem pengendalian *separator*.



**Gambar 2.2** Diagram Blok Sistem Pengendalian *Level Separator*

Berdasarkan gambar 2.2 pada sistem pengendalian *level* pada *production separator (132-V-02)* terdiri dari beberapa instrument yaitu *Level Indicator Controller (LIC-111)* dan *Level Transmitter*



(LT-111), dan *Level Valve* (LV-111). LIC (*Level Indicator Controller*) berfungsi sebagai media penunjukan *level* yang masuk ke dalam *separator* dan juga berfungsi sebagai pemberi sinyal ke aktuator (*control valve*) untuk melakukan *open* atau *close*. Pada *controller*, sinyal elektrik pada *transmitter* atau yang biasa disebut juga dengan *process variable value* dibandingkan dengan nilai set point yang sudah ditetapkan. Selisih antara *process variable value* dengan set point disebut juga dengan *error*. *Error* inilah yang dijadikan sebagai input bagi *controller*. Sedangkan output yang dihasilkan dari aktuator (*control valve*) disebut *manipulated value* (mv).

### 2.3 Pemodelan Matematis *Separator*

Pada *separator*, pemodelan matematis dicari menggunakan pendekatan hukum kesetimbangan massa seperti di bawah ini :

$$\begin{bmatrix} \text{laju} \\ \text{perubahan} \\ \text{massa} \end{bmatrix} = \begin{bmatrix} \text{laju} \\ \text{massa} \\ \text{input} \end{bmatrix} - \begin{bmatrix} \text{laju} \\ \text{massa} \\ \text{output} \end{bmatrix} \quad (2.1)$$

Hukum kesetimbangan massa menyatakan bahwa jumlah massa yang masuk ke dalam sistem sebanding dengan jumlah massa yang keluar dari sistem serta massa yang terakumulasi dalam sistem itu sendiri. Dengan menggunakan hukum kesetimbangan massa maka model matematis *separator* dapat ditulis sebagai berikut :

$$\rho \frac{dv}{dt} = \sum \dot{m}_{in} - \sum \dot{m}_{out} \quad (2.2)$$

Dengan :

$$\rho \frac{dv}{dt} : \text{laju perubahan massa } \textit{separator}$$

$$\sum \dot{m}_{in} : \text{total laju massa masukan}$$

$$\sum \dot{m}_{out} \quad : \text{ total laju massa keluaran}$$

Pada input *separator* berupa fasa campuran yang terdiri dari fasa gas dan fasa air. Kemudian kedua fasa tersebut dipisah berdasarkan densitasnya. Untuk menjaga *level* air dilakukan pengendalian agar air tidak tercampur dengan gas.

$$\text{Apabila } V = A \times h \quad (2.3)$$

Dan

$$\dot{m} = \rho_{liquid} \times Q \quad (2.4)$$

Maka

$$\rho_{liquid} A \frac{dh}{dt} = \sum \rho_{liquid} \cdot Q_{in} - \sum \rho_{liquid} \cdot Q_{out} \quad (2.5)$$

Karena nilai  $\rho_{liquid}$  memiliki nilai yang sama sehingga dapat dihilangkan, persamaan menjadi :

$$A \frac{dh}{dt} = (Q_{in} - Q_{out}) \quad (2.6)$$

Dengan A, H,  $Q_{in}$  dan  $Q_{out}$  berturut-turut adalah luas penampang separator ( $m^2$ ), ketinggian/*level* (m), *flowrate liquid input* ( $m^3/s$ ) dan *flowrate liquid output* ( $m^3/s$ ).

## 2.4 Level Transmitter

*Level Transmitter* berfungsi untuk mendeteksi fluida yang mengalir, jenis level transmitter yang banyak digunakan adalah DPT yaitu *Differential Pressure Transmitter*. *Differential pressure transmitter* bekerja berdasarkan prinsip beda tekanan, dimana *transmitter* bekerja berdasarkan keseimbangan gaya dua masukkan yang berbeda tekanan atau biasa disebut *hidrostatic head*. Prinsip pembacaan tekanan didalam tangki bergantung pada nilai *high*

*pressure* (HP) dan *lowpressure* (LP) pada *tapping point* yang telah ditentukan, sehingga dapat disimpulkan bahwa DP *transmitter* mengukur tekanan hydrostatic dari fluida yang ada di dalam tangki dan *hydrostatichead* dinyatakan dalam tekanan dengan persamaan (2.7) dan (2.8).



**Gambar 2.3** LDT (*Differential Presure Transmitter*)  
(Emerson, 2017)

$$\Delta P = P_{\text{high}} - P_{\text{low}} \quad (2.7)$$

$$\Delta P = \rho_1 \cdot g_1 \cdot h_1 - \rho_2 \cdot g_2 \cdot h_2 \quad (2.8)$$

Dimana :

$\Delta P$  : Tekanan Hidrostatik (mmH<sub>2</sub>O)

$\rho$  : Densitas fluida (kg/m<sup>3</sup>)

$g$  : Konstanta gravitasi (m/s<sup>2</sup>)

$h$  : *Level* fluida (m)

*Differential Presure Transmitter* bekerja berdasarkan prinsip beda tekanan antara sisi *high* dan *low*. Karena tekanan merupakan parameter utama dari *transmitter*, tentu saja density yang bervariasi akan mempengaruhi pembacaan *transmitter* yang bekerja berdasarkan prinsip beda tekanan. Kesalahan sensitivitas pada

*level transmitter* biasanya disebabkan karena keausan pada DP *transmitter* sehingga menurunkan *sensitivitas transmitter* tersebut.

## 2.5 Control Valve

*Control valve* dalam suatu *loop* pengendalian berfungsi untuk mengubah sinyal keluaran dari *controller* menjadi aksi yang dapat mengendalikan kondisi proses ke nilai yang telah ditentukan bila terjadi penyimpangan. Dasar pemilihan *control valve* pada dasarnya lebih di titik beratkan dari segi keselamatan prosesnya.

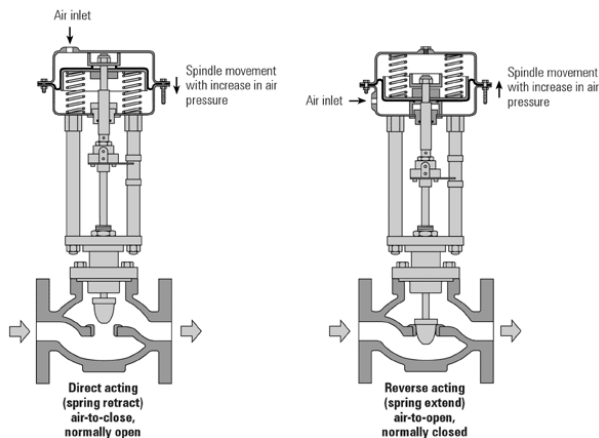
Aksi *control valve* terdiri dari dua macam yaitu:

a. ATO (*Air to Open*)

Bila diberikan masukan sinyal kendali *pneumatic* atau *electric* maka *control valve* tersebut akan membuka dan bila sinyal masukan tidak ada maka *control valve* akan menutup kembali.

b. ATC (*Air to Closed*)

Bila diberikan sinyal kendali *pneumatic* atau *electric* maka *control valve* tersebut akan menutup dan apabila sinyal masukan tidak ada maka *control valve* tersebut akan membuka kembali.



**Gambar 2.4** ATC dan ATO *Control Valve* (Goenka, 2009)

## 2.6 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 5 kategori dasar (Balaban, Saxena, Bansal, Goebel, & Curran, 2009).

- Bias

*Offset* atau penyimpangan konstan pada kondisi normal sinyal sensor. Bias juga dapat dideskripsikan 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  $Y_f = X + \beta + noise$ , dimana  $\beta$  adalah besar penyimpangan konstan.  $\beta$  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  $Y_f = X + \delta(t) + noise$ , 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

$Y_f = a(t) * X + noise$ , dimana  $0 < a(t) < \infty$  adalah parameter *scaling* yang dapat bernilai konstan ataupun berubah terhadap waktu.

- *Noise*  
*Noise* adalah adalah sinyal acak atau sinyal tak pasti yang memngaruhi sinyal keluaran sensor
- *Hard Fault*  
*Hard Fault* terjadi ketika keluaran sensor tertahan pada satu nilai dimana dapat direpresentasikan dengan  $Y_f = C + noise$ , dimana  $C$  adalah konstanta.

## 2.7 Observer

*Observer* adalah suatu algoritma yang digunakan untuk mengestimasi keadaan (*state*) dari variabel yang tidak terukur berdasarkan model matematis sistem, sehingga *observer* juga dapat digunakan untuk mengestimasi kesalahan yang terjadi pada suatu sistem jika kesalahan tersebut masuk dalam pemodelan. Suatu sistem umumnya dimodelkan dalam bentuk *state space*, yang dapat didefinisikan secara umum seperti persamaan sistem dalam bentuk diskrit dibawah ini (Ogata, 1987) :

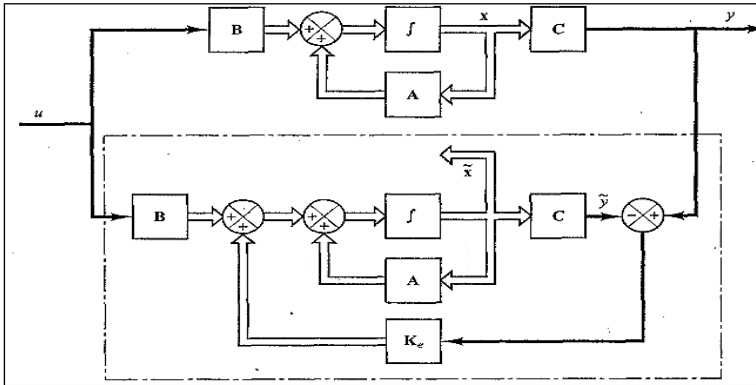
$$\dot{x}(t) = Ax(t) + Bu(t) \quad (2.9)$$

$$y(t) = Cx(t) + Du(t) \quad (2.10)$$

dengan:

A	: Matriks keadaan ( <i>state</i> )
B	:Matriks masukan
C	:Matriks keluaran
D	:Matriks gangguan
$u(t)$	:Masukan kendali
$y(t)$	:Vektor keluaran
$\dot{x}(t)$	:Vektor keadaan ( <i>state</i> )

Berikut merupakan diagram blok *observer* :



**Gambar 2.5** *Observer State* (K. Ogata, 1997)

Jika di dalam sistem atau *plant* terjadi kesalahan, maka persamaan 2.9 dan 2.10 akan menjadi persamaan berikut (K. Indriawati, 2015) :

$$\dot{x}(t) = Ax(t) + Bu(t) + F_a f_a(t) \quad (2.11)$$

$$y(t) = Cx(t) + Du(t) + F_s f_s(t) \quad (2.12)$$

Dimana:

$F_a$  = matriks kesalahan aktuator

$F_s$  = matriks kesalahan sensor

$f_a$  = vektor kesalahan aktuator

$f_s$  = vector kesalahan sensor

Kemudian *state* baru  $z(t)$  ditambahkan untuk memindahkan kesalahan sensor dari persamaan keluaran ke persamaan keadaan, persamaan tersebut dapat didefinisikan persamaan baru sebagai berikut:

$$\dot{z}(t) = Az(y(t) - z(t)) \quad (2.13)$$

$$\dot{z}(t) = Az(Cx(t) + F_s f_s(t) - z(t)) \quad (2.14)$$

$$\dot{z}(t) = Az.Cx(t) + Az.F_s.f_s(t) - Az.z(t) \quad (2.15)$$

Untuk  $Az$  merupakan matriks identitas ( $I$ ) untuk  $Az \in I^q$ , sedangkan matriks dari kesalahan sensor dan aktuator didefinisikan sebagai  $f = [f_s]$

Dengan menjadikan  $z$  sebagai *state* baru pada persamaan keadaan maka didapatkan *augmented state space* seperti yang ditunjukkan persamaan 2.16

$$\begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ Az.C & -Az \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} F_a & 0 \\ 0 & Az.F_s \end{bmatrix} \begin{bmatrix} f_a(t) \\ f_s(t) \end{bmatrix} \quad (2.16)$$

$$\dot{y}(t) = z(t) = [0 \quad 1] \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} \quad (2.17)$$

Dari persamaan *state space* (2.16) dan (2.17), persamaan tersebut dapat didefinisikan persamaan baru yang sudah di ekspansi sebagai berikut.

$$\begin{cases} \dot{X}_a = A_a x_a(t) + B_a u(t) + E_a f(t) \\ \dot{y}(t) = C_a x_a(t) \end{cases} \quad (2.18)$$

Dengan,  $X_a = \begin{bmatrix} x \\ z \end{bmatrix}$

Dimana:

$$A_a = \begin{bmatrix} A & 0 \\ Az.C & -Az \end{bmatrix}, B_a = \begin{bmatrix} B \\ 0 \end{bmatrix}, C_a = [0 \quad 1], E_a = \begin{bmatrix} F_a & 0 \\ 0 & Az.F_s \end{bmatrix}$$

Untuk memperoleh nilai dari *gain observer* maka dipakai ekspansi matrik melalui persamaan ruang keadaan baru yang ditunjukkan pada persamaan berikut.

$$\begin{cases} \dot{\hat{X}}_a(t) = A_a \cdot \hat{X}_a + B_a \cdot u + E_a \hat{f}_s + L_x (y - \hat{y}) \\ \hat{f}_s = L_s (y - \hat{y}) \\ \dot{\hat{Y}} = C_a \cdot \hat{X}_a \end{cases} \quad (2.19)$$



Dimana  $\hat{X}$ ,  $\hat{f}$ ,  $\hat{y}$  merupakan *state estimate*, *fault estimate* dan *output estimate*.  $L_x$  adalah *gain propotional* dari *observer* dan  $L_f$  adalah *gain integral* dari *observer*. Sehingga dari persamaan diatas akan menjadi sebagai berikut.

$$\begin{bmatrix} \dot{\hat{X}}_a(t) \\ \dot{\hat{f}}_s(t) \end{bmatrix} = \begin{bmatrix} A_a & E_a \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{X}_a(t) \\ \hat{f}_s(t) \end{bmatrix} + \begin{bmatrix} B_a \\ 0 \end{bmatrix} u + \begin{bmatrix} L_x \\ L_f \end{bmatrix} [C_a \quad 0](y - \hat{y}) \quad (2.20)$$

$$\hat{y}(t) = [C_a \quad 0] \begin{bmatrix} \hat{X}_a \\ \hat{f} \end{bmatrix} \quad (2.21)$$

Dari bentuk (2.20) dan (2.21) dapat diubah menjadi persamaan keadaan dan persamaan keluaran *observer*, sebagai berikut.

$$\dot{\tilde{x}}(t) = \tilde{A}_a \tilde{x}(t) + \tilde{B}_a u(t) + \tilde{K}_e (\tilde{y}(t) - \hat{y}(t)) \quad (2.22)$$

$$\tilde{y}(t) = \tilde{C}_a \tilde{x}(t) \quad (2.23)$$

Dengan,

$$\tilde{x} = \begin{bmatrix} \hat{X}_a \\ \hat{f}_s \end{bmatrix}$$

Sehingga matriks yang sudah diekspansi ditunjukkan seperti berikut ini:

$$\tilde{A}_a = \begin{bmatrix} A_a & E_a \\ 0 & 0 \end{bmatrix}, \tilde{C}_a = [C_a \quad 0]$$

Kemudian untuk mendapatkan nilai gain matriks observer dengan menggunakan solusi *pole placement* dimana *pole placement* adalah metode uji coba peletakan nilai *pole* yang sesuai.

## 2.8 Algoritma *Exponential Smoothing* untuk Prediksi Kesalahan

Kesalahan yang dimodelkan sebagai *output* aditif dari sistem separator merupakan kesalahan sensor. Dengan merancang *observer* maka dapat menghasilkan estimasi kesalahan sensor. Estimasi kesalahan sensor diasumsikan linier dari indeks waktu diskrit  $n$ . Metode *Holt's* diadopsi untuk memproses prediksi kesalahan. Algoritma prediksi kesalahan dapat dihitung melalui *exponential smoothing algorithm* sebagai berikut (Xu et al., 2009):

$$u_n = a\hat{a}_n + (1 - a)(u_{n-1} + s_{n-1}) \quad (2.24)$$

$$s_n = b(u_n - u_{n-1}) + (1 - b)s_{n-1} \quad (2.25)$$

Misalkan  $u_n$  adalah hasil estimasi yang diperhalus dan  $s_n$  adalah faktor pertumbuhan linear yang diestimasi. Diberikan hasil estimasi kesalahan  $\hat{a}_n (n = 1, 2, \dots)$  yang diperoleh dari *observer*. Dimana  $0 < a < 1$  &  $0 < b < 1$  adalah parameter *Holt's method* (Xu et al., 2009). Kondisi awal untuk *Holt's method* sebagai berikut:

$$u_1 = \hat{a}_1 = \hat{f}(k) \quad (2.26)$$

$$u_2 = \hat{a}_2 = \hat{f}(k + 1) \quad (2.27)$$

$$s_2 = u_2 - u_1 \quad (2.28)$$

Setelah menghitung estimasi kesalahan pada saat  $nT$  dan estimasi faktor pertumbuhan, maka prediksi kesalahan dapat dihitung menggunakan algoritma berikut (Xu et al., 2009):

$$\bar{\alpha}_{n+l} = u_n + s_n lT \quad (2.29)$$

Dengan

$l$  : *step* prediksi

$T$  : *time interval* untuk prediksi kesalahan

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 (2.26), atau (2.27). Jika  $n \geq 3$  menghitung estimasi kesalahan yang diperhalus  $u_n$  menggunakan persamaan (2.24).
- 2) *Step* estimasi faktor pertumbuhan: Menghitung  $n$ -*step* prediksi faktor pertumbuhan  $s_n$  dengan menggunakan (2.28) jika  $n = 2$ , dan menghitung  $n$ -*step* estimasi faktor pertumbuhan  $s_n$  dengan menggunakan (2.25) jika  $n \geq 3$ .
- 3) *Step* prediksi kesalahan: Menghitung prediksi kesalahan  $\bar{\alpha}_{n+t}$  dengan menggunakan persamaan (2.29).

## 2.9 Keandalan

Keandalan merupakan kemungkinan dari suatu komponen atau sistem untuk dapat beroperasi atau melaksanakan fungsinya (Ebeling, 1997). Fungsi tersebut telah ditetapkan pada kondisi pengoperasian tertentu dan dalam lingkungan tertentu untuk periode waktu yang telah ditentukan pula. Jadi, keandalan adalah kemungkinan untuk tidak mengalami kegagalan atau dapat melaksanakan fungsinya selama periode waktu ( $t$ ) atau lebih. Keandalan komponen, subsistem atau sistem produksi merupakan salah satu aspek yang dapat mempengaruhi kesuksesan proses produksi. Evaluasi keandalan dapat membantu dalam memperkirakan peluang sebuah sistem atau komponen untuk dapat melaksanakan fungsinya dalam jangka waktu tertentu dan dalam kondisi tertentu pula. Sistem atau komponen masih dapat beroperasi walaupun tidak dapat memenuhi fungsinya secara maksimal (Dhillon, 2005).

Nilai fungsi *reliability* merupakan nilai probabilitas, maka nilai fungsi *reliability* ( $R$ ) bernilai  $0 \leq R \leq 1$  (Ebeling, 1997). Kegagalan pada separator terjadi ketika fungsi yang dimaksudkan dari komponen separator tidak dapat bekerja dengan baik. Keandalan sistem pengendalian level dipengaruhi oleh keandalan sub-sistemnya, termasuk perangkat keras dan perangkat lunak, dan bagaimana komponen-komponen tersebut terhubung.

## 2.10 Prediksi Keandalan *Real-Time*

Penelitian mengenai keandalan *real-time* atau yang biasanya disebut juga dengan keandalan *on-line* ini mulai berkembang sejak tahun 1990-an. Pada tahun 1992, Kim dan rekannya meneliti mengenai prediksi kendalan *real-time* kondisional berdasarkan data kinerja alat. Sedangkan pada 1999, Chinnman melakukan penelitian mengenai estimasi keandalan *on-line* dari masing-masing komponen menggunakan sinyal degradasi.

Penelitian terus berkembang pada tahun 2000-an. Pada 2002, Chinnman kembali melanjutkan penelitiannya dengan mengambil topik berupa estimasi keandalan *on-line* dari sistem fisik menggunakan jaringan saraf. Penelitian tersebut terus berkembang hingga pada tahun 2009, Xu dan rekannya memperkenalkan penelitiannya mengenai sebuah metode baru dalam memonitoring keandalan *real-time* untuk sistem dinamik yang digabungkan dengan algoritma estimasi kesalahan

Setelah kesalahan diprediksi, keandalan pada sistem dinamik untuk beberapa waktu kedepan dapat dimonitoring. 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  $\mathcal{M}_1$ .

Secara umum, proses kesalahan yang sama memiliki dampak yang berbeda pada sifat sistem untuk sistem yang berbeda. Keadaan sistem merupakan hal yang krusial, dikarenakan berhubungan langsung dengan performansi atau keamanan dari sebuah sistem. Kedua, keadaan sistem tidak dapat diukur secara langsung, harus melalui estimasi dikarenakan kegagalan sistem didefinisikan dalam sebuah keadaan. Keadaan diestimasi menggunakan algoritma OTK-SIR. Pada tugas akhir ini, penulis tidak menggunakan algoritma OTK-SIR, namun menggunakan algoritma *observer*. Ketiga, kunci dari prediksi keandalan adalah prediksi kesalahan (2.30), sehingga algoritma RP (*Reliability Prediction*) yang berdasar pada algoritma *observer* dan prediksi kesalahan.

$$R(nT, (n+r)T) = \Pr(x_{n+r} \in \mathcal{M}_0 | x_n \in \mathcal{M}_0) \quad (2.30)$$

Dimana :

$R(nT, (n+r)T)$  : RP (*Reliability Prediction*)  
 $T$  : *Sampling Interval*  
 $\mathcal{M}_0$  : Batas nilai *variable*

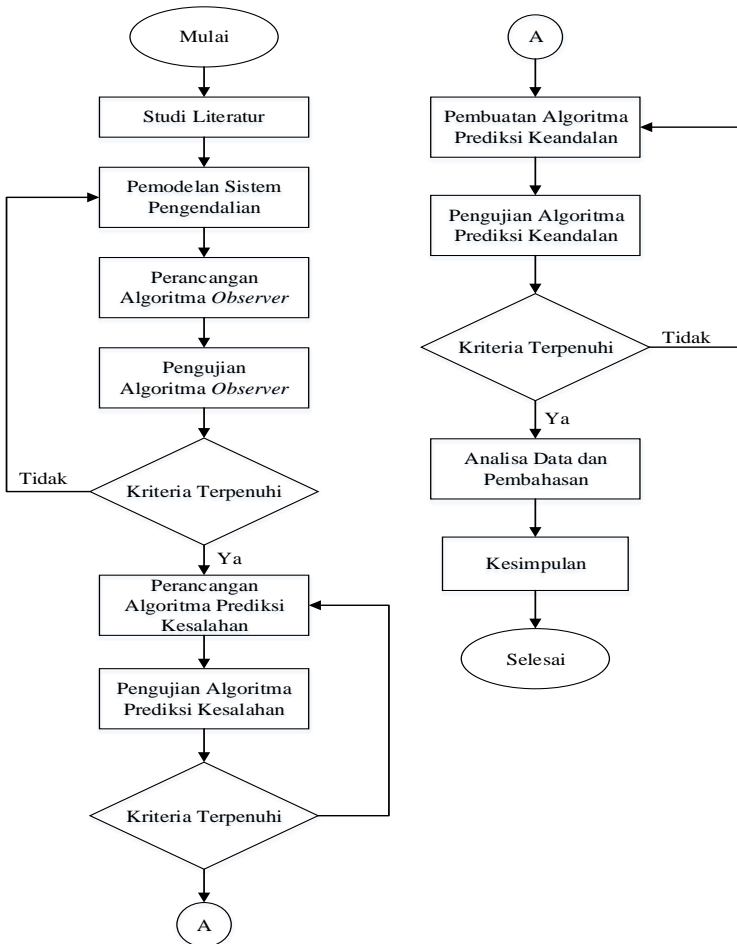
Penjelasan mengenai algoritma RP (*Reability Prediction*) adalah sebagai berikut :

- 1) *Step* inialisasi :  $l$  yang telah dinotasikan pada algoritma prediksi kesalahan dijadikan  $l = 1$ . Inialisasi 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  $\hat{a}_1$  dan  $\hat{a}_2$  pada persamaan (2.26) dan (2.27).
- 3) *Step* prediksi kesalahan: Jalankan algoritma prediksi kesalahan dengan  $l$  langkah untuk menghitung prediksi kesalahan  $\tilde{\alpha}_{n+l}$ .
- 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.

*Step* perhitungan prediksi keandalan : hitung jumlah partikel dari himpunan  $\{\bar{X}_{n+r|n}^{(i)}\} (i = 1, \dots, N)$  yang jatuh pada  $\mathcal{M}_0$  dan dinotasikan oleh  $N_0$ . Maka sistem prediksi keandalan selama  $(nT, (n+r)T)$  dihitung oleh  $N_0/N$

### BAB III METODOLOGI PENELITIAN

Metodologi yang digunakan untuk mencapai tujuan dari tugas akhir ini dijelaskan melalui diagram alir yang ditunjukkan pada gambar di bawah ini :



**Gambar 3.1** Diagram Alir Penelitian Tugas Akhir

### 3.1. Studi Literatur

Studi literatur merupakan tahapan untuk membangun pemahaman awal hingga detail mengenai topik tugas akhir yang diselesaikan. Sehingga, diperlukan studi literatur yang berhubungan erat dengan topik prediksi keandalan yang digunakan untuk sistem pengendalian *level* pada *separator*. Teori yang dipelajari antara lain pemahaman mengenai algoritma prediksi kesalahan, algoritma prediksi keandalan dan juga mengenai mekanisme perancangan sistem prediksi keandalan pada sistem pengendalian *level separator* secara *real-time* berbasis prediksi kesalahan sensor.

### 3.2. Pemodelan Sistem

Pemodelan sistem meliputi pemodelan *separator*, sensor dan aktuatur yang didapatkan dari data-data spesifikasi yang telah diambil. Data *plant* didapatkan dari ruang *maintenance* di SIPL (Saka Indonesia Pangkah Limited).

**Tabel 3.1** Data Spesifikasi *Separator*

Notasi	Keterangan	Nilai dan satuan
$H_{\max}$	Level maksimum	2,025 m
$H_{\min}$	Level minimum	0,6 m
$Q_{\text{in}}$	Flowrate input liquid	0,001835 m <sup>3</sup> /s
$\rho_{\text{liquid}}$	Massa jenis liquid	631,87 Kg/m <sup>3</sup>
D	Diameter Separator	1,22 m
L	Tinggi Tangki	4,978m

Untuk mengetahui respon suatu sistem melalui simulasi, dilakukan pemodelan matematis dari sistem tersebut meliputi pemodelan matematis *plant*, sensor atau transmitter, pemodelan unit kontrol, dan pemodelan matematis aktuatur.

#### 3.2.1. Pemodelan Matematis *Separator*

Pemodelan matematis *separator* dicari menggunakan pendekatan hukum kesetimbangan massa. Hukum ini menyatakan bahwa jumlah massa yang masuk ke dalam sistem sebanding dengan jumlah massa yang keluar dari sistem serta massa yang

terakumulasi dalam sistem itu sendiri. Dengan menggunakan persamaan hukum kesetimbangan massa, maka proses pada Separator dapat dimodelkan sebagai berikut:

$$\begin{bmatrix} \text{laju} \\ \text{perubahan} \\ \text{massa} \end{bmatrix} = \begin{bmatrix} \text{Laju} \\ \text{massa} \\ \text{input} \end{bmatrix} - \begin{bmatrix} \text{Laju} \\ \text{massa} \\ \text{output} \end{bmatrix}$$

Dengan menggunakan hukum kesetimbangan massa maka model matematis *separator* dapat ditulis sebagai berikut:

$$\rho \frac{dv}{dt} = \sum \dot{m}_{in} - \sum \dot{m}_{out} \quad (3.1)$$

Dengan :

$\rho \frac{dv}{dt}$  : laju perubahan massa separator

$\sum \dot{m}_{in}$  : total laju massa masukan

$\sum \dot{m}_{out}$  : total laju massa keluaran

Apabila  $V = A \times h$  dan  $\dot{m} = \rho_{liquid} \times Q$  maka :

$$\rho_{liquid} A \frac{dh}{dt} = \sum \rho_{liquid} \cdot Q_{in} - \sum \rho_{liquid} \cdot Q_{out} \quad (3.2)$$

Karena nilai  $\rho_{liquid}$  memiliki nilai yang sama sehingga dapat dihilangkan, persamaan menjadi :

$$A \frac{dh}{dt} = (Q_{in} - Q_{out}) \quad (3.3)$$

Dari persamaan (3.3) didapatkan model matematis *separator* sebagai berikut:

$$\frac{dV_L}{dt} = Q_{Lin} - Q_{Lout} \quad (3.4)$$



Untuk mendapatkan *state level* dari sistem *separator*, maka nilai spesifikasi terukur dari diameter separator dimasukkan kedalam persamaan sebagai berikut :

$$A \frac{dh}{dt} = Q_{in} - Q_{out} \quad (3.7)$$

$$\frac{dh}{dt} = \frac{4}{\pi d^2} (Q_{in} - Q_{out}) \quad (3.8)$$

$$\frac{dh}{dt} = \frac{4}{\pi 1,22^2} (Q_{in} - Q_{out}) \quad (3.9)$$

$$\frac{dh}{dt} = 0,856 (Q_{in} - Q_{out}) \quad (3.10)$$

### 3.2.2. Pemodelan Matematis Aktuator

Aktuator yang digunakan adalah *level control valve*. *Level control valve* yang digunakan dilengkapi dengan *I/P converter* yang berfungsi untuk mengubah sinyal input 4-20 mA menjadi sinyal *pneumatic* 3-15 psi. Dalam hal ini, *level control valve* memiliki masukan sinyal berupa arus listrik kemudian diubah menjadi tekanan untuk menggerakkan *control valve*.

Pemodelan aktuator yang digunakan untuk mengubah arus 4-20 mA menjadi tekanan 3-15 psi yang akan menggerakkan aktuator dengan nilai H maksimum sebesar 2,025 m dan H minimum 0,6 m adalah sebagai berikut :

$$\frac{\Delta P_v}{\Delta I} = \frac{P_{vmax} - P_{vmin}}{Arus_{max} - Arus_{min}} \quad (3.11)$$

$$\frac{\Delta P_v}{\Delta I} = \frac{15 - 3}{20 - 4} = 0,75 \quad (3.12)$$

Transfer function dari *pneumatic line delay* adalah sebagai berikut:

$$\frac{\Delta p_c}{\Delta p_v} = \frac{1}{\tau_0 s + 1} \quad (3.13)$$

$$\frac{\Delta p_c}{\Delta p_v} = \frac{1}{0,2s + 1} \quad (3.14)$$

$\tau_0$  adalah konstanta waktu aktuator , dengan nilai sebesar 0,2 sekon.(Wang, 2000)

Untuk mendapatkan *state pneumatic line* dari sistem separator, maka nilai *pline* 0,75 dari persamaan (3.12) dikalikan dengan persamaan (3.14) sehingga persamaannya menjadi sebagai berikut :

$$\frac{Pline(s)}{u(s)} = \frac{0,75}{0,2s + 1} \quad (3.15)$$

$$(0,2 s + 1)Pline = 0,75 u(s) \quad (3.16)$$

$$0,2 Pline(s) = 0,75 u(s) - Pline \quad (3.17)$$

$$Pline(s) = \frac{0,75}{0,2} u(s) - \frac{1}{0,2} Pline \quad (3.18)$$

$$Pline(s) = 3,75 u(s) - 5 Pline \quad (3.19)$$

*Transfer function* dari respon *control valve* adalah sebagai berikut:

$$\frac{\Delta x(s)}{\Delta P(s)} = \frac{100}{\frac{P_{max} - P_{min}}{c_0 s + 1}} \quad (3.20)$$

$$\begin{aligned}\frac{\Delta x(s)}{\Delta P(s)} &= \frac{100}{\frac{15-3}{2s+1}} \\ &= \frac{25}{\frac{3}{2s+1}}\end{aligned}$$

$c_0$  adalah konstanta waktu *control valve*, dengan nilai 2 sekon (Wang, 2000).

Dari persamaan (3.20) maka didapatkan *state respon control valve* sebagai berikut :

$$\frac{\Delta x}{Pline} = \frac{8,33}{2s + 1} \quad (3.21)$$

$$2\Delta x(s) + \Delta x = 8,33 Pline \quad (3.22)$$

$$\Delta x(s) = \frac{8,33}{2} Pline - \frac{1}{2} \Delta x \quad (3.23)$$

$$\Delta x(s) = 4,165 Pline - 0,5 \Delta x \quad (3.24)$$

### 3.2.3. Pemodelan Matematis Level Transmitter

Sensor *level* yang digunakan adalah *Differential Pressure Transmitter*, yaitu alat yang berfungsi untuk mengukur tekanan didalam separator dan memanfaatkan perbedaan tekanan untuk mengukur *level* didalam separator. *Transmitter* ini mengubah input berupa *level* menjadi arus dengan persamaan sebagai berikut :

$$\frac{I(s)}{H(s)} = \frac{I_{\max} - I_{\min}}{H_{\max} - H_{\min}} \quad (3.25)$$

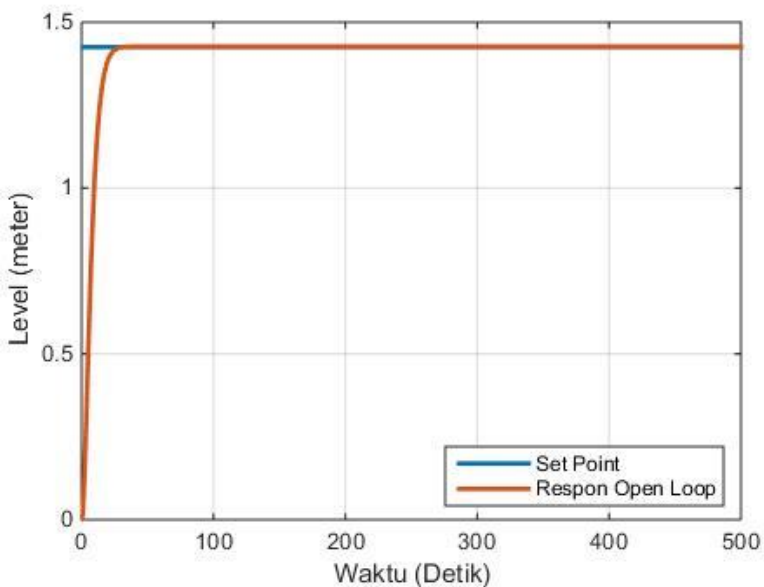
Dimana nilai H maksimum sebesar 2,025 m dan H minimum sebesar 0,6 m, I maksimum sebesar 20 mA dan I minimum sebesar 4 mA sehingga :

$$\frac{I(s)}{H(s)} = \frac{20 - 4}{2,025 - 0,6}$$

$$\frac{I(s)}{H(s)} = 11,22$$

### 3.3 Perancangan Sistem Kendali PI

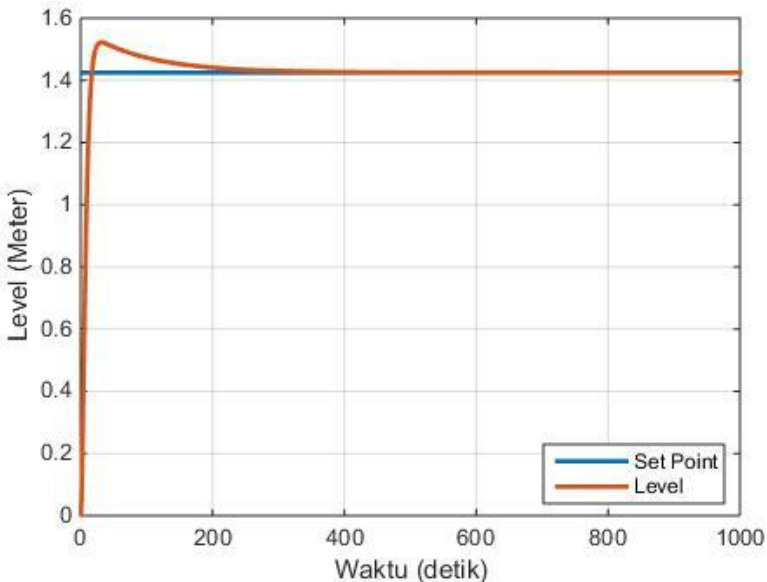
Setelah pemodelan matematis didapatkan, selanjutnya model disimulasikan dengan bantuan *software* Matlab (R2014a) untuk dilakukan validasi dengan data hasil pengukuran yang dimiliki oleh SPIL. Berikut respon uji *open loop* plant :



**Gambar 3.2** Respon *Open Loop*

Grafik respon *openloop* sistem ditunjukkan pada Gambar 3.2. Pada grafik terlihat bahwa sistem berorde satu karena pada saat terjadi penambahan input, maka secara otomatis *level* juga akan bertambah dan tidak konstan pada satu nilai *level*.

Selanjutnya dilakukan perancangan sistem kendali pada sistem separator. Dalam sistem pengendalian *level* pada *separator* menggunakan pengendalian *feedback* dengan algoritma PI. Untuk menentukan gain pengendali PI menggunakan metode *trial and error* hingga memenuhi kriteria *error steady state*  $\leq 2\%$ . Pada sistem ini didapatkan bahwa nilai  $K_p$  sebesar 1 dan  $T_i$  sebesar 0.01. Setelah itu sistem disimulasikan dengan menggunakan *software* Matlab (2014b) dengan respon seperti gambar 3.3. Berdasarkan respon sistem tersebut dapat diketahui karakteristik respon sistem yakni *rise time* sebesar 10,9 detik, *settling time* sebesar 147 detik dan *maximum overshoot* sebesar 6,71%, dan *error steady state* sebesar 0,14%



**Gambar 3.3** Respon *Closed Loop*

### 3.4 Perancangan *Observer*

Perancangan *observer* dilakukan untuk mengestimasi kesalahan yang ada dengan cara mengembangkan persamaan ruang yang ada, dengan cara mengembangkan persamaan ruang keadaan

sistem. Model *state space* dari sistem pengendalian *level separator* terdiri dari 3 *state* yaitu *pneumatic line delay*, respon *control valve* ( $\Delta x$ ) dan *level*. Persamaan (3.19), (3.24) dan (3.10) merupakan persamaan dari *state pneumatic line delay*, respon *control valve* ( $\Delta x$ ) dan *level*, kemudian ketiga persamaan tersebut diubah kedalam domain waktu.

$$\frac{dPneumatic_{line}(t)}{dt} = -5Pneumatic_{line}(t) + 3,75u(t) \quad (3.26)$$

$$\frac{d\Delta x(t)}{dt} = 4,165 Pneumatic_{line}(t) - 0,5\Delta x(t) \quad (3.27)$$

$$\frac{dh(t)}{dt} = 0,856Q_{in} - 0,856Q_{out} \quad (3.28)$$

Ketiga persamaan diatas kemudian dibuat kedalam bentuk matriks ruang keadaan ruang keadaan dengan mengacu pada persamaan seperti berikut ini :

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (3.29)$$

$$y(k) = Cx(t) + Du(t) \quad (3.30)$$

Sehingga,

$$\begin{bmatrix} \frac{dPneumatic_{line}}{dt} \\ \frac{d\Delta x}{dt} \\ \frac{dh}{dt} \end{bmatrix} = \begin{bmatrix} -5 & 0 & 0 \\ 4,165 & -0,5 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_{line} \\ \Delta x \\ h \end{bmatrix} \quad (3.31)$$

$$+ \begin{bmatrix} 3,75 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0,856 & -0,856 \end{bmatrix} \begin{bmatrix} u \\ Q_{in} \\ Q_{out} \end{bmatrix}$$

$$\begin{bmatrix} \frac{d\Delta x}{dt} \\ \frac{dh}{dt} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{line} \\ \Delta x \\ h \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ Q_{in} \\ Q_{out} \end{bmatrix} \quad (3.32)$$

Dari persamaan diatas (3.29) dan (3.30) dapat diketahui nilai matriks ruang keadaan untuk perancangan *observer* sebagai berikut:

$$A = \begin{bmatrix} -5 & 0 & 0 \\ 4.165 & -0.5 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 3.75 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0.856 & -0.856 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$F_s = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Persamaan ruang keadaan yang mengandung kesalahan sensor dituliskan sebagai berikut :

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (3.33)$$

$$y(t) = Cx(t) + F_s f_s(t) \quad (3.34)$$

Dengan A, B, C,  $F_s$ ,  $u(t)$ ,  $x(t)$ ,  $y(t)$  dan  $f_s(t)$  masing-masing adalah matriks keadaan (state), matriks masukan, matriks keluaran, matriks kesalahan sensor, masukan kendali, vektor keadaan (state), vektor keluaran, vektor kesalahan sensor.

Kemudian *state* baru  $\dot{z}(t)$  ditambahkan untuk memindahkan kesalahan sensor dari persamaan keluaran ke persamaan keadaan.

Persamaan tersebut dapat didefinisikan persamaan baru sebagai berikut:

$$\dot{z}(t) = A_z(y(t) - z(t)) \quad (3.35)$$

$$\dot{z}(t) = A_z(Cx(t) + Fsfs(t) - z(t)) \quad (3.36)$$

$$\dot{z}(t) = A_z(Cx(t) + Fsfs(t) - A_z.z(t)) \quad (3.37)$$

Untuk  $A_z$  merupakan matriks identitas ( $I$ ) untuk  $A_z \in I^q$ , sedangkan matriks dari kesalahan sensor didefinisikan sebagai  $f = [f_s]$

Dengan menjadikan  $z$  sebagai *state* baru pada persamaan keadaan maka didapatkan *augmented state space* seperti yang ditunjukkan persamaan 3.38

$$\begin{aligned} \begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} &= \begin{bmatrix} A & 0 \\ A_z.C & -A_z \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(t) \\ &\quad + \begin{bmatrix} 0 \\ A_z.F_s \end{bmatrix} f_s(t) \end{aligned} \quad (3.38)$$

$$\dot{y}(t) = [0 \quad I] \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} \quad (3.39)$$

Sehingga didapatkan,

$$\begin{aligned} A_a &= \begin{bmatrix} A & 0 \\ A_z.C & -A_z \end{bmatrix} = \begin{bmatrix} -5 & 0 & 0 & 0 & 0 \\ 4.166 & -0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix} \\ B_a &= \begin{bmatrix} B \\ 0 \end{bmatrix} = \begin{bmatrix} 3.75 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0.0155 & -0.0155 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned}$$



$$C_a = \begin{bmatrix} 0 \\ C \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$E_a = \begin{bmatrix} 0 \\ A_z \cdot F_s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Untuk memperoleh struktur *observer* maka diperlukan ekspansi matriks melalui persamaan ruang keadaan baru.

$$\begin{cases} \dot{\hat{X}}_a(t) = A_a \hat{X}_a(t) + B_a u(t) + E_a f_s(t) \\ \dot{y}(t) = C_a \hat{X}_a(t) \end{cases} \quad (3.40)$$

Untuk memperoleh nilai *gain observer*, maka ditambahkan *state* yang mengandung *gain* agar dapat mengestimasi kesalahan sensor, sehingga ekspansi matriks ruang keadaan baru yang ditunjukkan pada persamaan berikut.

$$\begin{cases} \dot{\hat{X}}_a(t) = A_a \cdot \hat{X}_a + B_a \cdot u + E_a \hat{f}_s + L_s(Y - \hat{Y}) \\ \hat{f}_s = L_s(Y - \hat{Y}) \\ \dot{\hat{Y}} = C_a \cdot \hat{X}_a \end{cases} \quad (3.41)$$

Sehingga dari persamaan diatas jika diubah kedalam bentuk yang lebih ringkas akan menjadi sebagai berikut.

$$\begin{bmatrix} \dot{\hat{X}}_a(t) \\ \dot{\hat{f}}_s(t) \end{bmatrix} = \begin{bmatrix} A_a & E_a \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{X}_a(t) \\ \hat{f}_s(t) \end{bmatrix} + \begin{bmatrix} B_a \\ 0 \end{bmatrix} u + \begin{bmatrix} L_x \\ L_f \end{bmatrix} [C_a \quad 0](Y - \hat{Y}) \quad (3.42)$$

$$\dot{\hat{Y}}(t) = [C_a \quad 0] \begin{bmatrix} \hat{X}_a \\ \hat{f}_s \end{bmatrix} \quad (3.43)$$

Dari bentuk (3.42) dan (3.43) dapat diubah menjadi persamaan keadaan dan persamaan keluaran *observer*, sebagai berikut.

$$\hat{\tilde{x}}(t) = \tilde{A}_a \tilde{x}(t) + \tilde{B}_a u(t) + \tilde{K}_e (\hat{y}(t) - \hat{\tilde{y}}(t)) \quad (3.44)$$

$$\hat{\tilde{y}}(t) = \tilde{C}_a \tilde{x}(t) \quad (3.45)$$

Dengan mensubstitusikan nilai matriks  $A_a$ ,  $C_a$  dan  $E_a$ , diperoleh nilai matriks  $\tilde{A}_a$  dan  $\tilde{C}_a$  sebagai berikut.

$$\tilde{A}_a = \begin{bmatrix} A_a & E_a \end{bmatrix} = \begin{bmatrix} -5 & 0 & 0 & 0 & 0 & 0 \\ 4.166 & -0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\tilde{C}_a = \begin{bmatrix} C_a & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Untuk dapat menyelesaikan *Observer* maka diperlukan pemilihan dari nilai *pole placement*, nilai *pole placement* yang dipakai adalah :

Pole ke 1= -4,5340 + 0,0000i

Pole ke 2= -0,7607 + 0,9908i

Pole ke 3= -0,7607 - 0,9908i

Pole ke 4= 0,0000 + 0,0000i

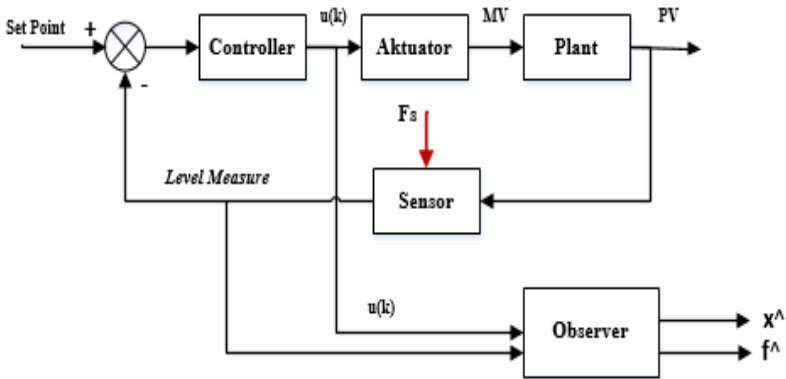
Pole ke 5= -0,6139 + 1,9347i

Pole ke 6= -0,6139 - 1,9347i

Dari nilai *pole placement* diatas secara otomatis dengan menggunakan perangkat lunak *Matlab R2014a* dengan *syntax* “*place*” didapatkan nilai *gain observer* bernilai sebagai berikut.

$$Ke = \begin{bmatrix} -2,1200 & 0 \\ 2,9036 & 0 \\ 0 & -1,167 \cdot 10^{-13} \\ -0,4446 & 0 \\ 0 & 0,2278 \end{bmatrix}$$

Berikut ini adalah gambar diagram blok sistem pengendalian yang telah dintegrasikan observer.



**Gambar 3.4** Diagram Blok Sistem Pengendalian Yang Telah Terintegrasi Dengan *Observer*

### 3.5 Pengujian Algoritma *Observer*

Pengujian *observer* dilakukan untuk memastikan apakah *observer* berupa hasil estimasi kesalahan sensor ( $f^{\wedge}(k)$ ) yang telah dibuat telah sesuai dengan kesalahan rill yang telah dirancang. Pengujian ini dilakukan dengan menggunakan sinyal kesalahan berupa *ramp*.

### 3.6 Perancangan Algoritma Prediksi Kesalahan

Setelah perancangan algoritma *observer*, langkah berikutnya perancangan algoritma prediksi kesalahan yang mengacu pada (Xu et al., 2009) yaitu algoritma *exponential smoothing*. Estimasi kesalahan yang diperhalus dapat dihitung dengan persamaan sebagai berikut:

$$u_n = a\hat{\alpha}_n + (1 - a)(u_{n-1} + s_{n-1}) \quad (3.44)$$

$$s_n = b(u_n - u_{n-1}) + (1 - b)s_{n-1} \quad (3.45)$$

Misalkan  $u_n$  adalah hasil estimasi yang diperhalus dan  $s_n$  adalah faktor pertumbuhan linear yang diestimasi. Diberikan hasil estimasi kesalahan  $\hat{a}_n (n = 1, 2, \dots)$  yang diperoleh dari *observer*. Dimana  $0 < a < 1$  &  $0 < b < 1$  adalah parameter *Holt's method* (Xu et al., 2009). Kondisi awal untuk *Holt's method* sebagai berikut:

$$u_1 = \hat{a}_1 = \hat{f}(k) \quad (3.46)$$

$$u_2 = \hat{a}_2 = \hat{f}(k + 1) \quad (3.47)$$

$$s_2 = u_2 - u_1 \quad (3.48)$$

Setelah menghitung estimasi kesalahan yang diperhalus dan faktor pertumbuhan pada saat  $nT$  prediksi kesalahan kemudian dapat dihitung menggunakan algoritma berikut (Xu et al., 2009):

$$\bar{a}_{n+l} = u_n + s_n l T \quad (3.49)$$

dengan

$l$ : *step* prediksi untuk prediksi kesalahan

$T$ : *time interval*

Untuk  $n = 1$  atau  $n = 2$  menggunakan persamaan (3.46) atau (3.47) untuk  $n > 2$  menggunakan persamaan (3.44).

### 3.7 Pengujian Algoritma Prediksi Kesalahan Sensor

Pengujian algoritma prediksi kesalahan sensor dilakukan untuk memastikan apakah hasil estimasi kesalahan oleh *observer* dapat diterapkan pada algoritma prediksi kesalahan *exponential smoothing*. Hasil estimasi *observer* digunakan untuk menghitung estimasi kesalahan diperhalus  $u_n$  untuk  $n \geq 2$  dan menghitung faktor pertumbuhan linear  $s_n$  untuk  $n \geq 2$ . Dimana nilai  $u_n$  dan  $s_n$  dipengaruhi oleh parameter *Holt's method* ( $a$  dan  $b$ ) dan *Time Interval*. Pengujian algoritma prediksi kesalahan akan dilakukan dengan pemberian variasi *time interval* pada algoritma untuk mendapatkan nilai *time interval* yang dapat memberikan hasil

prediksi  $u_n$   $n \geq 2$  yang optimal. Selain itu pengujian akan dilakukan dengan variasi jumlah step prediksi kedepan untuk mendapatkan nilai step kedepan terbesar dimana algoritma masih memberikan hasil prediksi yang optimal. Hasil prediksi dikatakan optimal jika saat  $t_{U_{n+1}} = t_{U_n} + T$  nilai  $u_{n+1} = u_n$ .

### 3.8 Perancangan Algoritma Prediksi Keandalan *Real-Time*

Langkah terakhir yaitu perancangan algoritma prediksi keandalan *real-time* yang mengacu pada (Xu et al., 2009).

- 1) Inisialisasi partikel dimulai dari  $l = 1$ .

$$\bar{X}_{n+l-1|n}^{(i)} = \bar{X}_{n+0|n}^{(i)} = \hat{X}_{n|n}^{(i)}$$

- 2) Estimasi kesalahan menggunakan algoritma *observer* yang akan menghasilkan *state* estimasi mengacu pada (Indriawati et al., 2015).

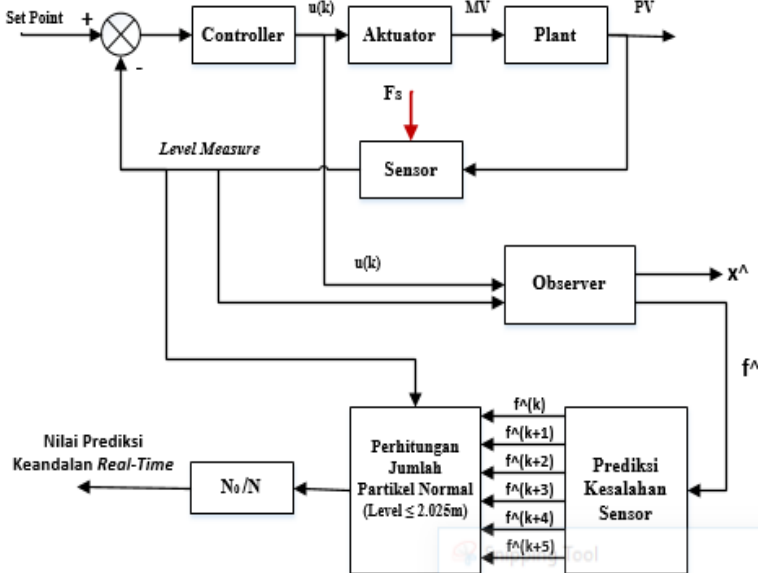
$$u_1 = \hat{\alpha}_1 = \hat{f}(k)$$

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

- 3) Prediksi kesalahan menggunakan algoritma *exponential smoothing* pada persamaan (3.30), (3.31), dan (3.35).
- 4) Menghitung partikel baru menggunakan persamaan :

$$\bar{X}_{n+l-1|n}^{(i)} = f(\bar{X}_{n+l-1|n}^{(i)}) + \Gamma \tilde{\alpha}_{n+l} + W_{n+l}^{(i)} \text{ dimana } W_{n+l}^{(i)} \text{ adalah realisasi dari } w_{n+l} \text{ untuk partikel ke-}i.$$

- 5) Pemilihan partikel jika  $l < r$ , maka  $l = l + 1$ , dan kembali ke step 2, jika sebaliknya lanjut pada step 6.
- 6) Menghitung jumlah partikel dari himpunan  $\{\bar{X}_{n+r|n}^{(i)}\} (i = 1, \dots, N)$  yang jatuh pada  $\mathcal{M}_0$  (area yang diperbolehkan) dan dinotasikan oleh  $N_0$ . Maka sistem prediksi keandalan selama  $(nT, (n + r)T|$  dihitung oleh  $N_0/N$ .



**Gambar 3.5** Diagram Blok Sistem Prediksi Keandalan *Real-Time*

Gambar 3.5 merupakan diagram blok sistem prediksi keandalan *real-time* yang telah diintegrasikan dengan sistem pengendalian level pada *separator*, *observer*, dan prediksi kesalahan sensor.  $\hat{f}(k)$  merupakan nilai estimasi kesalahan dari *observer* kemudian digunakan dalam prediksi kesalahan sensor. Hasil prediksi kesalahan sensor digunakan matriks untuk mengambil *state* prediksi kesalahan sensor. Selanjutnya akan dibandingkan nilai *state* prediksi kesalahan sensor dengan nilai level maksimum sebesar 2,025 m, jika *state* prediksi kesalahan sensor melebihi nilai 2,025 m maka sistem dikatakan gagal. Lalu pada blok  $N_0/N$  merupakan hasil jumlah partikel yang memasuki area yang diperbolehkan  $\mathcal{M}_0$  yang dinotasikan  $N_0$  (*Level* tidak melebihi  $\leq 2,025$  m) kemudian dibandingkan jumlah tersebut dengan jumlah seluruh partikel ( $N$ ).

### 3.9 Pengujian Algoritma Prediksi Keandalan *Real-Time*

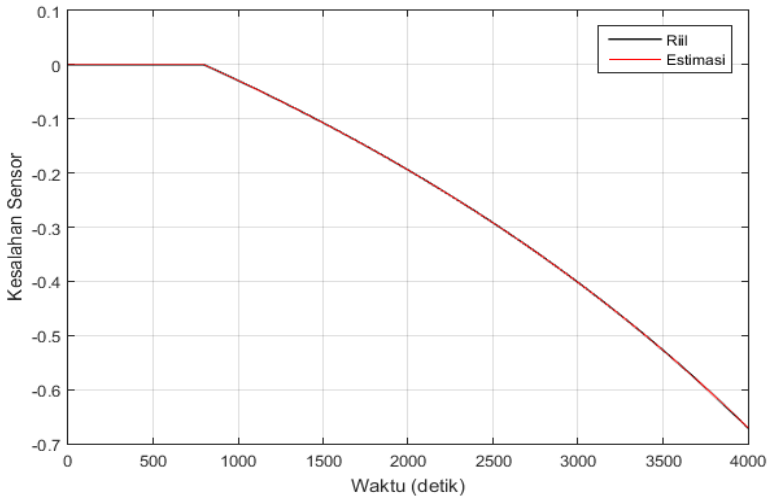
Pengujian algoritma prediksi keandalan *real-time* dilakukan untuk menentukan nilai parameter desain yang mempengaruhi prediksi keandalan *real-time* pada sistem pengendalian *level separator*. Hasil prediksi kesalahan sensor digunakan dalam memprediksi keandalan *real-time*. Pengujian ini dilakukan dengan memberikan variasi nilai  $T$  (*time interval*) 1 detik, 2 detik dan 5 detik. Setiap nilai *time interval* diberikan variasi jumlah prediksi tiga prediksi, lima prediksi, tujuh prediksi dan sepuluh prediksi. Pengujian hasil prediksi keandalan *real-time* dilakukan dengan prosedur sebagai berikut:

- a. Prediksi kesalahan sensor yang dihasilkan melalui algoritma *exponential smoothing* selanjutnya digunakan matriks untuk mengambil *state* prediksi kesalahan sensor.
- b. Dibandingkan nilai *state* prediksi kesalahan sensor dengan nilai maksimum kecepatan sebesar 2,025 m, jika *state* prediksi kesalahan sensor melebihi nilai 2,025 m maka sistem dikatakan gagal.
- c. Lalu pada  $N_0/N$  merupakan hasil jumlah partikel yang memasuki area yang diperbolehkan  $\mathcal{M}_0$  yang dinotasikan  $N_0$  ( $\text{Level} \leq 2,025 \text{ m}$ ) kemudian dibandingkan jumlah tersebut dengan jumlah seluruh partikel ( $N$ ).

## BAB IV ANALISIS DATA DAN PEMBAHASAN

### 4.1. Hasil Uji *Observer*

Pengujian *observer* dilakukan dengan tujuan untuk mengetahui kemampuan *observer* dalam mengestimasi kesalahan sensor. Pada pengujian ini sensor diasumsikan mengalami *loss of sensitivity* atau penurunan sensitivitas. Kesalahan sensor yang disimulasikan berupa sinyal *ramp* sebesar  $-0.0001/t$  dan kesalahan diberikan pada detik ke 800. Berikut grafik hasil pengujian *observer* :

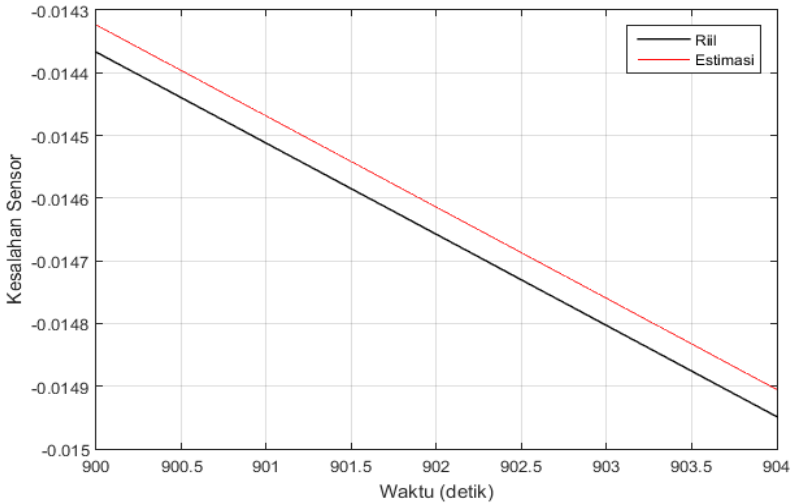


**Gambar 4.1** Perbandingan Hasil Estimasi dengan Kesalahan Sensor (Ril)

Gambar 4.1 menunjukkan *observer* mampu mengestimasi kesalahan sensor dengan baik dapat dilihat pada saat terjadi perubahan nilai estimasi di awal maupun pada saat detik ke-800, tidak ada *overshoot* dan waktu yang dibutuhkan untuk melakukan estimasi memiliki nilai yang sama dengan nilai riil kesalahan



sensor. Dapat diketahui bahwa hasil estimasi kesalahan sensor mengikuti dinamika kesalahan sensor atau riil yang terjadi.



**Gambar 4.2** Perbandingan Hasil Estimasi dengan Kesalahan Sensor (perbesaran gambar 4.1)

**Tabel 4.1** Perbandingan Hasil Estimasi dengan Kesalahan Sensor

Waktu	Riil	Estimasi	Error (%)
900	-0,01432	-0,01437	0,302436
901	-0,01447	-0,01451	0,299475
902	-0,01461	-0,01466	0,296572
903	-0,01476	-0,0148	0,293726
904	-0,01491	-0,01495	0,290934

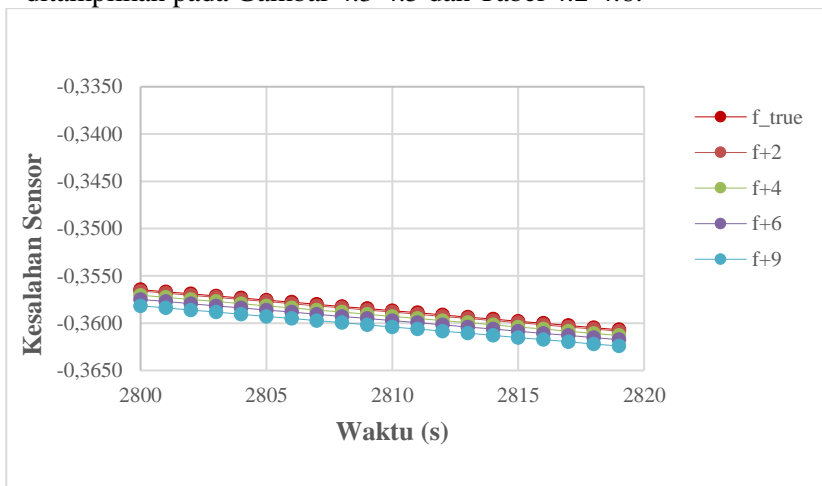
Gambar 4.2 merupakan grafik perbesaran hasil estimasi dan kesalahan sensor, terlihat bahwa hasil estimasi kesalahan dan nilai riilnya memiliki perbedaan yang relatif kecil. Untuk nilai hasil estimasi dari nilai kesalahan riil pada detik ke 900-904 ditunjukkan pada Tabel 4.1 terlihat bahwa rata-rata *error* hasil estimasi dari nilai riil yaitu sekitar 0,297%. Dari hasil pengujian yang diperoleh

menunjukkan bahwa *observer* yang telah dirancang telah mampu mengikuti dinamika kesalahan sensor atau riil yang terjadi.

#### 4.2 Hasil Uji Algoritma Prediksi Kesalahan Sensor

Prediksi kesalahan sensor dirancang menggunakan hasil estimasi kesalahan sensor dari *observer*. Algoritma prediksi kesalahan sensor mengacu pada algoritma *exponential smoothing*. Nilai parameter *Holt's method* pada algoritma *exponential smoothing* diberikan sebesar  $a=0,9$ , 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 dimana untuk setiap nilai *time interval* diberikan variasi jumlah prediksi. Pengujian algoritma prediksi kesalahan sensor ini dilakukan dengan memberikan kesalahan pada sensor dalam bentuk sinyal *ramp* dengan slope  $-0.0001/t$  dan kesalahan dimulai pada detik ke-800.

Berikut hubungan prediksi kesalahan sensor terhadap waktu ditampilkan pada Gambar 4.3-4.5 dan Tabel 4.2-4.6.



**Gambar 4.3** Grafik Perbandingan Nilai Hasil Prediksi Kesalahan Sensor dengan *Time Interval* 1s

**Tabel 4.2** Hasil Prediksi dengan *Time Interval* 1s

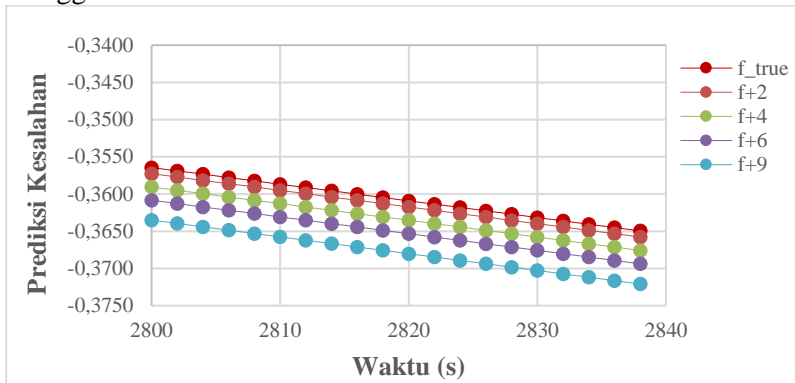
t	<i>True Fault</i>	f(k+2)	f(k+4)	f(k+6)	f(k+9)
2800	-0,3564	-0,3566	-0,3571	-0,3575	-0,3582
2801	-0,3567	-0,3568	-0,3573	-0,3577	-0,3584
2802	-0,3569	-0,3571	-0,3575	-0,3579	-0,3586
2803	-0,3571	-0,3573	-0,3577	-0,3582	-0,3588
2804	-0,3573	-0,3575	-0,3579	-0,3584	-0,3591
2805	-0,3576	-0,3577	-0,3582	-0,3586	-0,3593
2806	-0,3578	-0,3579	-0,3584	-0,3588	-0,3595
2807	-0,3580	-0,3582	-0,3586	-0,3591	-0,3597
2808	-0,3582	-0,3584	-0,3588	-0,3593	-0,3600
2809	-0,3585	-0,3586	-0,3591	-0,3595	-0,3602
2810	-0,3587	-0,3588	-0,3593	-0,3597	-0,3604
2811	-0,3589	-0,3591	-0,3595	-0,3600	-0,3606
2812	-0,3591	-0,3593	-0,3597	-0,3602	-0,3608
2813	-0,3593	-0,3595	-0,3600	-0,3604	-0,3611
2814	-0,3596	-0,3597	-0,3602	-0,3606	-0,3613
2815	-0,3598	-0,3600	-0,3604	-0,3608	-0,3615
2816	-0,3600	-0,3602	-0,3606	-0,3611	-0,3617
2817	-0,3602	-0,3604	-0,3608	-0,3613	-0,3620
2818	-0,3605	-0,3606	-0,3611	-0,3615	-0,3622
2819	-0,3607	-0,3608	-0,3613	-0,3617	-0,3624

Pada gambar 4.3 dan tabel 4.2 menunjukkan hasil prediksi kesalahan sensor dengan *step* prediksi  $f(k+2)$ ,  $f(k+4)$ ,  $f(k+6)$  dan  $f(k+9)$  dengan menggunakan *time interval* sebesar 1 detik.

Dimana nilai prediksi  $f(k+2)$  menunjukkan nilai kesalahan 1 kali *time interval* lebih cepat,  $f(k+4)$  menunjukkan nilai kesalahan 3 kali *time interval* lebih cepat,  $f(k+6)$  menunjukkan nilai kesalahan 5 kali *time interval* lebih cepat,  $f(k+9)$  menunjukkan nilai kesalahan 8 kali *time interval* lebih cepat dari nilai kesalahan yang sebenarnya (*true fault*) sensor. Validasi kebenaran hasil prediksi kesalahan dilakukan dengan menganalisa nilai dan waktu pada setiap *step* prediksi  $f(k+2)$ ,  $f(k+4)$ ,  $f(k+6)$  dan  $f(k+9)$  terhadap waktu kesalahan yang sebenarnya (*true fault*).

Pada pengujian ini, *true fault* berfungsi sebagai *validator* kebenaran hasil algoritma prediksi kesalahan sensor. Pada gambar 4.3 menunjukkan bahwa prediksi kesalahan *step* prediksi  $f(k+2)$ ,  $f(k+4)$ ,  $f(k+6)$  dan  $f(k+9)$  memiliki *error* yang relatif kecil dari nilai kesalahan sebenarnya yaitu pada *step* prediksi  $f(k+2)$  terdapat *error* sebesar 0,04%, pada *step* prediksi  $f(k+4)$  terdapat *error* sebesar 0,17%, pada *step* prediksi  $f(k+6)$  terdapat *error* sebesar 0,29%, pada *step* prediksi  $f(k+9)$  terdapat *error* sebesar 0,48%.

Berikut data hasil prediksi kesalahan sensor dengan menggunakan *time interval* 2s.



**Gambar 4.4** Grafik Hasil Algoritma Prediksi Kesalahan Sensor dengan *Time Interval* 2s

**Tabel 4.3** Hasil Prediksi dengan *Time Interval* 2s

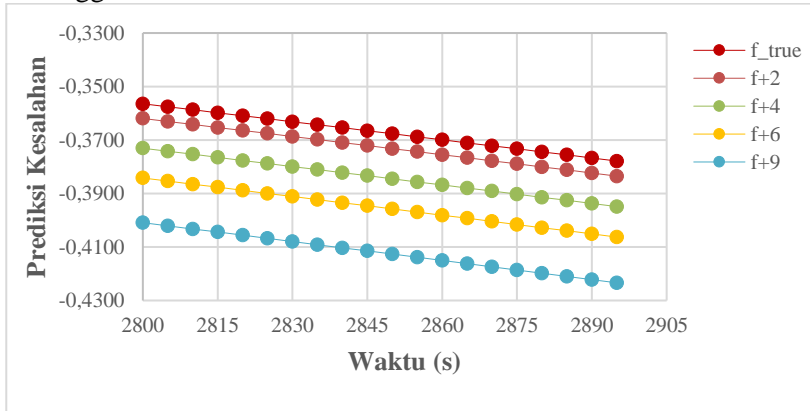
t	<i>True Fault</i>	$f(k+2)$	$f(k+4)$	$f(k+6)$	$f(k+9)$
2800	-0,3564	-0,3573	-0,3591	-0,3608	-0,3635
2802	-0,3569	-0,3577	-0,3595	-0,3613	-0,3640
2804	-0,3573	-0,3582	-0,3599	-0,3617	-0,3644
2806	-0,3578	-0,3586	-0,3604	-0,3622	-0,3649
2808	-0,3582	-0,3591	-0,3608	-0,3626	-0,3653
2810	-0,3587	-0,3595	-0,3613	-0,3631	-0,3658
2812	-0,3591	-0,3600	-0,3617	-0,3635	-0,3662
2814	-0,3596	-0,3604	-0,3622	-0,3640	-0,3667

**Tabel 4.4** Hasil Prediksi dengan *Time Interval* 2s (Lanjutan)

<b>t</b>	<b>True Fault</b>	<b>f(k+2)</b>	<b>f(k+4)</b>	<b>f(k+6)</b>	<b>f(k+9)</b>
2816	-0,3600	-0,3608	-0,3626	-0,3644	-0,3671
2818	-0,3605	-0,3613	-0,3631	-0,3649	-0,3676
2818	-0,3605	-0,3613	-0,3631	-0,3649	-0,3676
2820	-0,3609	-0,3617	-0,3635	-0,3653	-0,3680
2822	-0,3614	-0,3622	-0,3640	-0,3658	-0,3685
2824	-0,3618	-0,3626	-0,3644	-0,3662	-0,3689
2826	-0,3623	-0,3631	-0,3649	-0,3667	-0,3694
2828	-0,3627	-0,3635	-0,3653	-0,3671	-0,3698
2830	-0,3632	-0,3640	-0,3658	-0,3676	-0,3703
2832	-0,3636	-0,3644	-0,3662	-0,3680	-0,3707
2834	-0,3641	-0,3649	-0,3667	-0,3685	-0,3712
2836	-0,3645	-0,3653	-0,3671	-0,3689	-0,3716
2838	-0,3650	-0,3658	-0,3676	-0,3694	-0,3721

Pada gambar 4.4 dan tabel 4.3 - 4.4 menunjukkan hasil prediksi kesalahan sensor dengan *step* prediksi  $f(k+2)$ ,  $f(k+4)$ ,  $f(k+6)$  dan  $f(k+9)$  dengan menggunakan *time interval* sebesar 2 detik. Diperoleh *error* untuk setiap *step* prediksi  $f(k+2)$ ,  $f(k+4)$ ,  $f(k+6)$  dan  $f(k+9)$  dari nilai kesalahan sebenarnya (*true fault*) yaitu pada *step* prediksi  $f(k+2)$  terdapat *error* sebesar 0,23%, pada *step* prediksi  $f(k+4)$  terdapat *error* sebesar 0,73%, pada *step* prediksi  $f(k+6)$  terdapat *error* sebesar 1,22%, pada *step* prediksi  $f(k+9)$  terdapat *error* sebesar 1,97%. Meskipun *error* prediksi kesalahan dengan menggunakan *time interval* 2s pada setiap *step* prediksi masih terbilang kecil namun jika dibandingkan dengan hasil prediksi dengan *time interval* 1s, *error* prediksi dengan *time interval* 2s relatif jauh lebih besar. Hal ini dapat dilihat pada grafik gambar 4.4, dimana deviasi yang dimiliki setiap *step* prediksi mulai cenderung membesar. Selanjutnya dilakukan pengujian dengan menggunakan *time interval* lebih besar lagi yaitu 5s untuk membuktikan bahwa penambahan *time interval* mempengaruhi penambahan deviasi untuk setiap *step* prediksi.

Berikut data hasil prediksi kesalahan sensor dengan menggunakan *time interval* 5s.



**Gambar 4.5** Grafik Hasil Algoritma Prediksi Kesalahan Sensor dengan *Time Interval* 5s

**Tabel 4.5** Hasil Prediksi dengan *Time Interval* 5s

t	True Fault	f(k+2)	f(k+4)	f(k+6)	f(k+9)
2800	-0,3564	-0,3619	-0,3731	-0,3842	-0,4009
2805	-0,3576	-0,3631	-0,3742	-0,3854	-0,4021
2810	-0,3587	-0,3642	-0,3753	-0,3865	-0,4032
2815	-0,3598	-0,3653	-0,3765	-0,3877	-0,4044
2820	-0,3609	-0,3664	-0,3776	-0,3888	-0,4056
2825	-0,3620	-0,3676	-0,3788	-0,3900	-0,4068
2830	-0,3632	-0,3687	-0,3799	-0,3911	-0,4079
2835	-0,3643	-0,3698	-0,3811	-0,3923	-0,4091
2840	-0,3654	-0,3710	-0,3822	-0,3934	-0,4103
2845	-0,3665	-0,3721	-0,3833	-0,3946	-0,4115
2850	-0,3677	-0,3732	-0,3845	-0,3958	-0,4127
2855	-0,3688	-0,3744	-0,3856	-0,3969	-0,4139
2860	-0,3699	-0,3755	-0,3868	-0,3981	-0,4150
2865	-0,3710	-0,3766	-0,3879	-0,3993	-0,4162
2870	-0,3722	-0,3778	-0,3891	-0,4004	-0,4174
2875	-0,3733	-0,3789	-0,3903	-0,4016	-0,4186
2880	-0,3744	-0,3801	-0,3914	-0,4028	-0,4198

**Tabel 4.6** Hasil Prediksi dengan *Time Interval* 5s (Lanjutan)

t	<i>True Fault</i>	f(k+2)	f(k+4)	f(k+6)	f(k+9)
2885	-0,3756	-0,3812	-0,3926	-0,4039	-0,4210
2890	-0,3767	-0,3823	-0,3937	-0,4051	-0,4222
2895	-0,3779	-0,3835	-0,3949	-0,4063	-0,4234

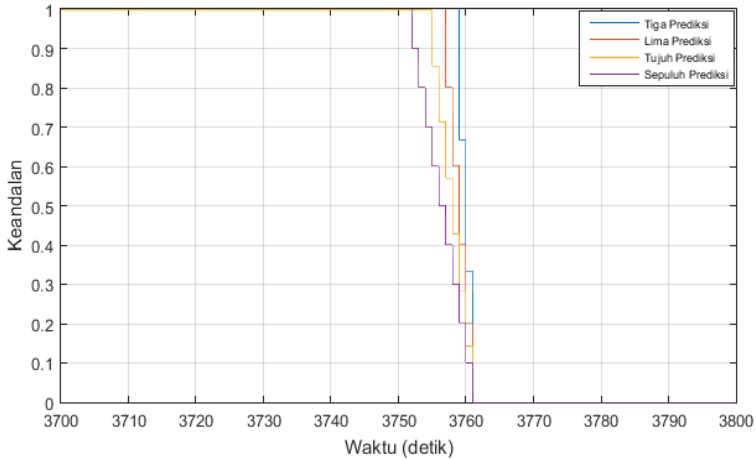
Pada gambar 4.5 dan tabel 4.5 - 4.6 menunjukkan hasil prediksi kesalahan sensor dengan *step* prediksi  $f(k+2)$ ,  $f(k+4)$ ,  $f(k+6)$  dan  $f(k+9)$  dengan menggunakan *time interval* sebesar 5 detik. Diperoleh *error* untuk setiap *step* prediksi  $f(k+2)$ ,  $f(k+4)$ ,  $f(k+6)$  dan  $f(k+9)$  dari nilai kesalahan sebenarnya (*true fault*) yaitu pada *step* prediksi  $f(k+2)$  terdapat *error* sebesar 1,5%, pada *step* prediksi  $f(k+4)$  terdapat *error* sebesar 4,6%, pada *step* prediksi  $f(k+6)$  terdapat *error* sebesar 7,7%, pada *step* prediksi  $f(k+9)$  terdapat *error* sebesar 12,3%. Jika dibandingkan dengan hasil prediksi menggunakan *time interval* 1s dan 2s, hasil prediksi menggunakan *time interval* 5s menghasilkan *error* hasil prediksi kesalahan yang jauh lebih besar.

Dari hasil pengujian yang diperoleh, membuktikan bahwa hasil algoritma prediksi kesalahan sensor dipengaruhi oleh nilai *time interval* yang diberikan. Semakin kecil nilai *time interval* maka semakin baik peforma algoritma prediksi kesalahan.

### 4.3 Hasil Uji Algoritma Prediksi Keandalan *Real-Time*

Prediksi keandalan *real-time* diterapkan menggunakan hasil prediksi kesalahan sensor yang telah diuji sebelumnya. Berdasarkan hasil pengujian algoritma prediksi kesalahan yaitu algoritma prediksi kesalahan sensor bekerja secara optimal pada *time interval* 1s maka pada pengujian prediksi keandalan *real-time* ini juga menggunakan *time interval* 1s. Hasil uji dari algoritma prediksi keandalan *real-time* dapat digunakan sebagai acuan tingkat “kesehatan” sistem pengendalian level pada *separator* ketika terdapat kesalahan pada sensor. Pengujian algoritma prediksi keandalan ini akan dilakukan untuk prediksi 3 *step*, 5 *step*, 7 *step* dan 10 *step* kedepan. Kesalahan dimulai pada detik ke-800

dengan batas kesalahan  $\mathcal{M}_0$  sebesar 2,025 m, yang berarti jika nilai *level* melebihi 2,025 m, maka sistem dikatakan gagal.



**Gambar 4.6** Grafik Hasil Uji Prediksi Keandalan *Real-Time* dengan *Time Interval* 1s

**Tabel 4.7** Hasil Uji Prediksi Keandalan *Real-Time*

Waktu (detik)	Keandalan Tiga Prediksi	Waktu (detik)	Keandalan Lima Prediksi	Waktu (detik)	Keandalan Tujuh Prediksi	Waktu (detik)	Keandalan Sepuluh Prediksi
0-3759	1	0 - 3757	1	0-3755	1	0 - 3752	1
3759-3760	0,7	3757-3758	0,8	3755 - 3756	0,9	3752 - 3753	0,9
3760-3761	0,3	3758-3759	0,6	3756 - 3757	0,7	3753 - 3754	0,8
	3761	3759-3760	0,4	3757 - 3758	0,6	3754 - 3755	0,7
		3760 - 3761	0,2	3758 - 3759	0,4	3755 - 3756	0,6
		3761	0	3759 - 3760	0,3	3756 - 3757	0,5
				3760 - 3761	0,1	3757 - 3758	0,4
				3761	0	3758 - 3759	0,3
						3759 - 3760	0,2
						3760 - 3761	0,1
						3761	0



Pada gambar 4.6 ditunjukkan grafik keandalan hasil uji algoritma prediksi keandalan *real-time* untuk 3,5,7 dan 10 *step* kedepan. Kesalahan dimulai pada detik ke- 800 pada simulasi. Pada gambar 4.6 menunjukkan bahwa jumlah *step* prediksi yang lebih banyak akan terlebih dahulu mendeteksi terjadinya kegagalan hal ini dapat dilihat dari penurunan nilai keandalan yang terjadi. Kurva keandalan dengan 10 *step* prediksi lebih dahulu mengalami penurunan keandalan pada detik 3752 diikuti oleh keandalan dengan 7 *step* prediksi pada detik 3755 selanjutnya diikuti oleh keandalan dengan 5 *step* prediksi pada detik 3757 dan keandalan dengan 3 *step* prediksi pada detik 3759.

Berdasarkan hasil yang diperoleh dapat diketahui bahwa prediksi keandalan *real-time* untuk 10 *step* prediksi mempunyai performansi terbaik jika dibandingkan dengan *step* prediksi lainnya. Hal ini didasarkan pada waktu keandalan mulai turun untuk sepuluh prediksi lebih teliti, tampak bahwa pada detik ke 3755 sudah terdeteksi terjadi kesalahan sensor. Hasil tersebut sesuai dengan penelitian sebelumnya yang diaplikasikan pada *three-vessel water tank system* bahwa semakin besar jumlah prediksi maka tingkat prediksi keandalan akan semakin teliti (Xu et al., 2009).

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan hasil analisis data yang telah dilakukan, maka kesimpulan yang diperoleh dari tugas akhir ini antara lain :

- *Observer* sebagai *estimator* kesalahan sensor yang telah dirancang mampu mengikuti dinamika kesalahan residual atau riil yang terjadi dengan nilai *error observer* sebesar 0.297%.
- Algoritma prediksi kesalahan *exponential smoothing* dipengaruhi oleh variasi *time interval*. Algoritma mampu memberikan hasil prediksi kesalahan yang optimal pada *time interval* 1s dengan nilai parameter *holt's method*  $a = 0.9$  dan  $b = 0.9$ .
- Algoritma prediksi keandalan *real-time* mampu mendeteksi kegagalan terlebih dahulu. Jumlah prediksi 10 menghasilkan prediksi keandalan *real-time* lebih baik dibandingkan jumlah prediksi sebesar dari 3, 5 dan 7.

#### **5.2 Saran**

Pada Tugas Akhir ini, metode algoritma prediksi kesalahan yang dirancang menggunakan algoritma *exponential smoothing*. Pada pengembangan berikutnya untuk perancangan prediksi kesalahan dapat menggunakan metode lain untuk mendapatkan hasil yang lebih optimal.

*“Halaman ini sengaja dikosongkan”*

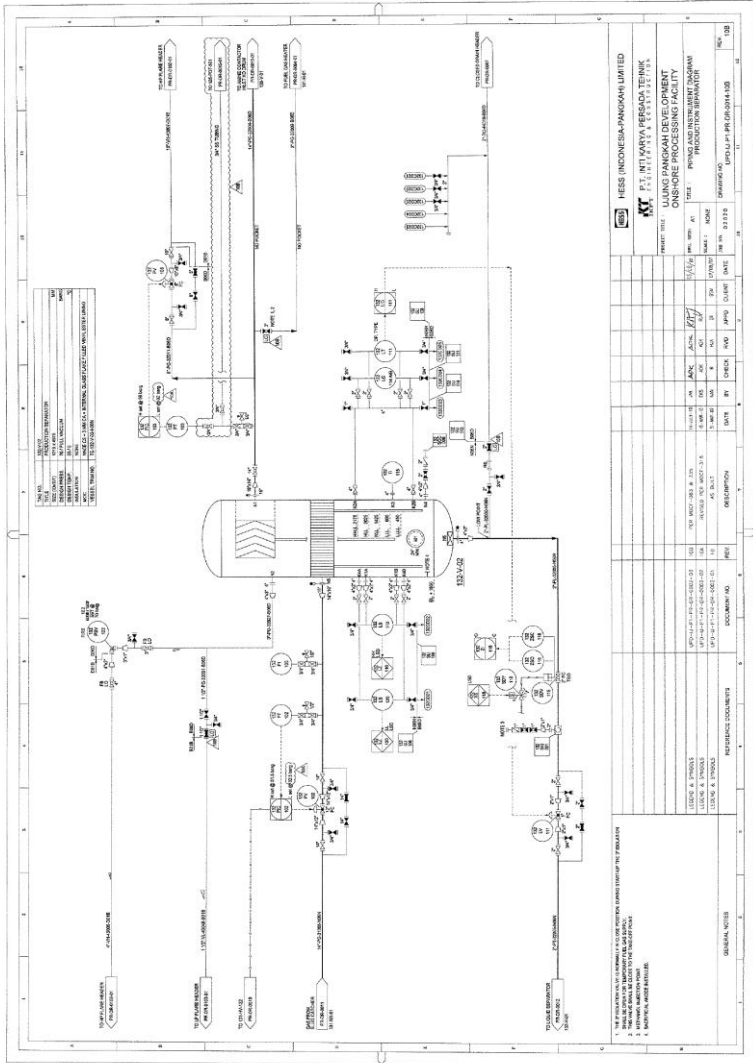
## DAFTAR PUSTAKA

- 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.
- Chen, B. (2016). *Digital Control System*. Singapore: The National University of Singapore.
- Chinnam, R. B. (1999). *On-line Reliability Estimation of Individual Components, Using Degradation Signals*, 48
- Dhillon, B. S. (2005). *Reliability, Quality, and Safety for Engineers*.
- Ebeling, C. (1997). *An Introduction To Reliability and Maintainability Engineering*. Boston: McGraw-Hill.
- Emerson. (2017, Mei). Dipetik Mei 16, 2019, dari <http://www.emerson.com/en-us/catalog/rosemount-3051-coplanar-pressure-transmitter>
- Fitria, M. A (2018) Monitoring Keandalan *Real-Time* pada Sistem Pengendalian Level Separator Berdasarkan Estimasi Kesalahan
- Isermann, R. (2006). *Fault-Diagnosis Systems*. Darmstadt: Springer.
- 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.
- Lu, H., Kolarik, W., & Lu, S. (2001). Real Time Performance Reliability Prediction. *IEEE Transactions On Reliability*, 353 - 357.
- Ogata, K. (1970). *Modern Control Engineering*. New Jersey: Prentice-Hall.
- Ogata, K. (1987). *Discrete-Time Control Systems*. USA: Prentice-Hall, Inc.
- Ogata, K. (2004). *System Dynamics*. New Jersey: Prentice-Hall.

- Ogata, K. (2010). *Modern Control Engineering 5th Edition*. New Jersey: Prentice Hall.
- Prapdito, R. R (2015) Analisis Reliability Dari Basic Process Control System Dan Safety Instrumented System Pada Separator PV-9700 Di Job Pertamina-Petrocina Tuban
- Rubiandani, R. (2001). *Peralatan Eksplorasi Minyak* : Bandung.
- Walijah, C. S. (2017). Perancangan Active Fault Tolerant Control Pada Sistem Pengendalian Level Separator Di Petronas Muriah LTD Dengan Kesalahan Pada Sensor dan Aktuator. *ITS : Surabaya*.
- Wang, S. (2000). Dynamic Simulation Experimental Investigation And Control System Design Of Gas Liquid Cylindrical Cyclone Separators. *Tulsa, Oklahoma, Amerika Serikat*.
- Zhang, Y., & Jiang, J. (2008). Bibliographical review on reconfigurable fault-tolerant control systems. *Science Direct* , 229-252.
- Zhegguo Xu, Y. j. (2008). Real-time Reliability Prediction for a Dynamic System Based on the Hidden Degradation Process Identification.
- Zhengguo Xu, Y. J. (2009). A New Real-Time Reliability Prediction Method for Dynamic Systems Based on On-line Fault Prediction. *IEEE Transaction On Reliability*, Vol 58. 3.

# LAMPIRAN A

## P&id Separator 132-V-02 Saka Indonesia Pangkah Limited (SIPL)



NO. REV.	REVISION
01	ISSUE FOR CONSTRUCTION
02	FOR REVIEW
03	FOR APPROVAL
04	FOR CONSTRUCTION
05	FOR REVIEW
06	FOR APPROVAL
07	FOR CONSTRUCTION
08	FOR REVIEW
09	FOR APPROVAL
10	FOR CONSTRUCTION

<b>PT. INTI PANGKAH SEPTA TUNAS</b> <b>LIANG PANGKAH DEVELOPMENT</b> <b>ON-SHORE PROCESSING FACILITY</b>	
PERMIT NO.	132-V-02
SCALE	1:1
DATE	13/01/2010
DESIGNER	Y. S. S.
CHECKER	Y. S. S.
APPROVER	Y. S. S.
DATE	13/01/2010
REVISION	
NO.	DESCRIPTION
1	ISSUE FOR CONSTRUCTION
2	FOR REVIEW
3	FOR APPROVAL
4	FOR CONSTRUCTION
5	FOR REVIEW
6	FOR APPROVAL
7	FOR CONSTRUCTION
8	FOR REVIEW
9	FOR APPROVAL
10	FOR CONSTRUCTION

1. ALL P&ID SHALL BE APPROVED BY THE OPERATIONAL SUPERVISOR OF THE FACILITY
2. ALL P&ID SHALL BE APPROVED BY THE DESIGNER
3. ALL P&ID SHALL BE APPROVED BY THE PROJECT MANAGER
4. ALL P&ID SHALL BE APPROVED BY THE CLIENT

DESIGNED BY: Y. S. S.

CHECKED BY: Y. S. S.

APPROVED BY: Y. S. S.

DATE: 13/01/2010

SCALE: 1:1

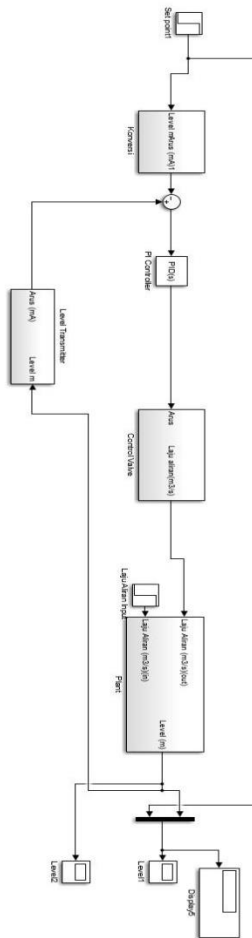
NO. REV.

REVISION

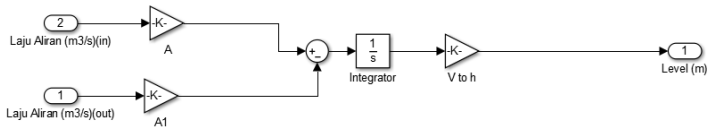
## LAMPIRAN B

### SIMULINK SISTEM PREDIKSI KEANDALAN *REAL-TIME* PADA SISTEM PENGENDALIAN *LEVEL SEPARATOR*

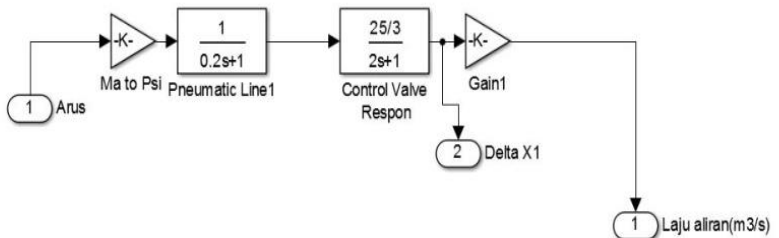
#### B.1 Skema Perancangan Sistem Pengendalian *Level* pada *Separator*.



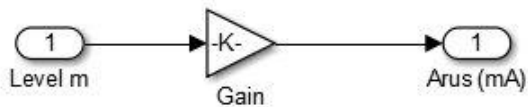
## B.2 Pemodelan Plant (*Separator*)



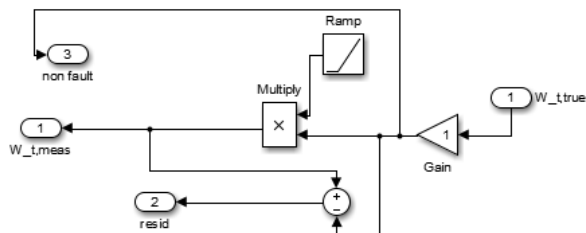
## B.3 Pemodelan Aktuator



## B.4 Pemodelan *Level Transmitter*

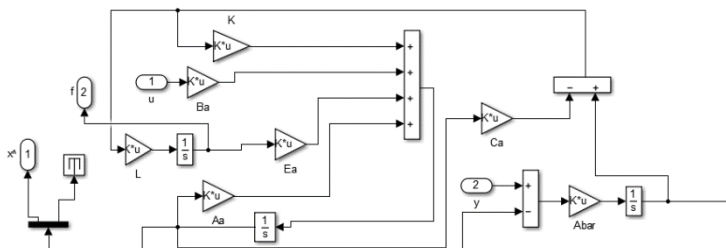


## B.5 Pemodelan Kesalahan *Level Transmitter*

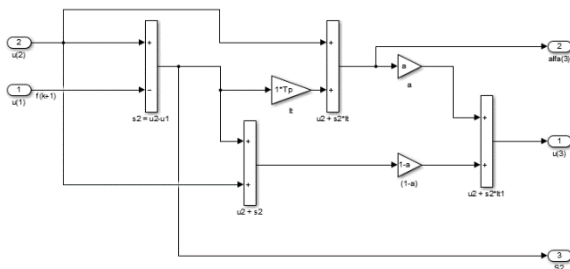
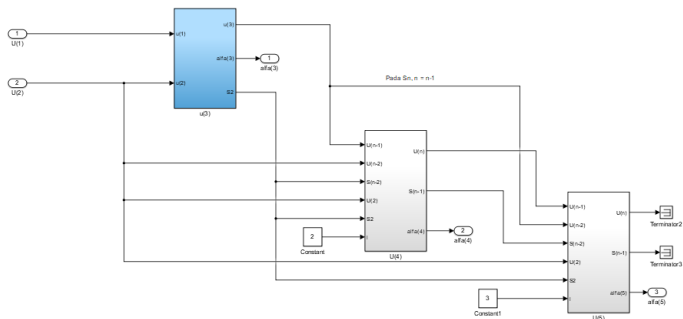




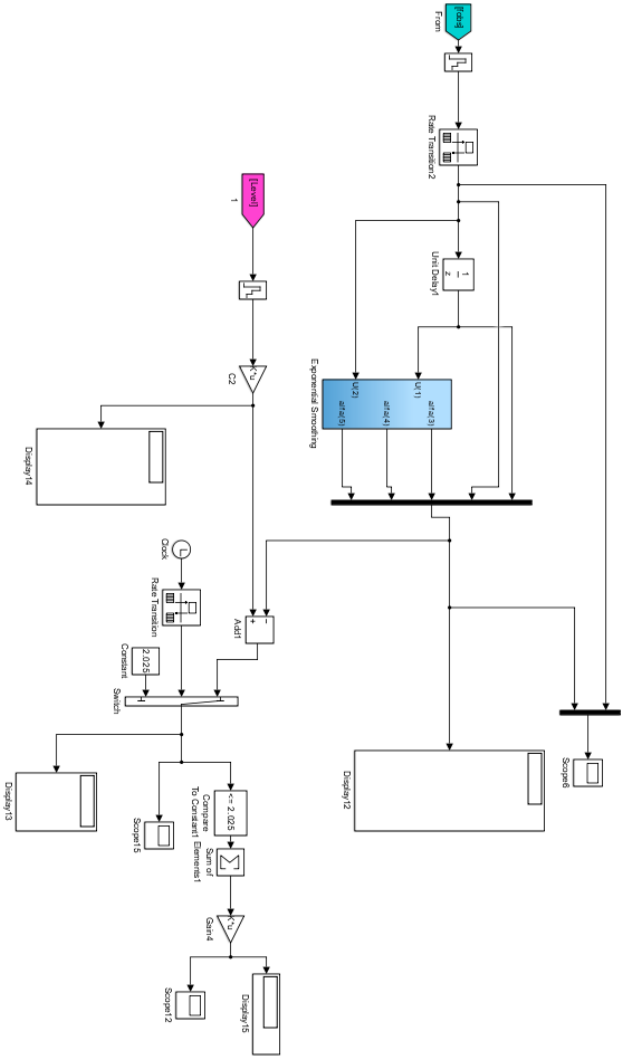
## B.6 Observer



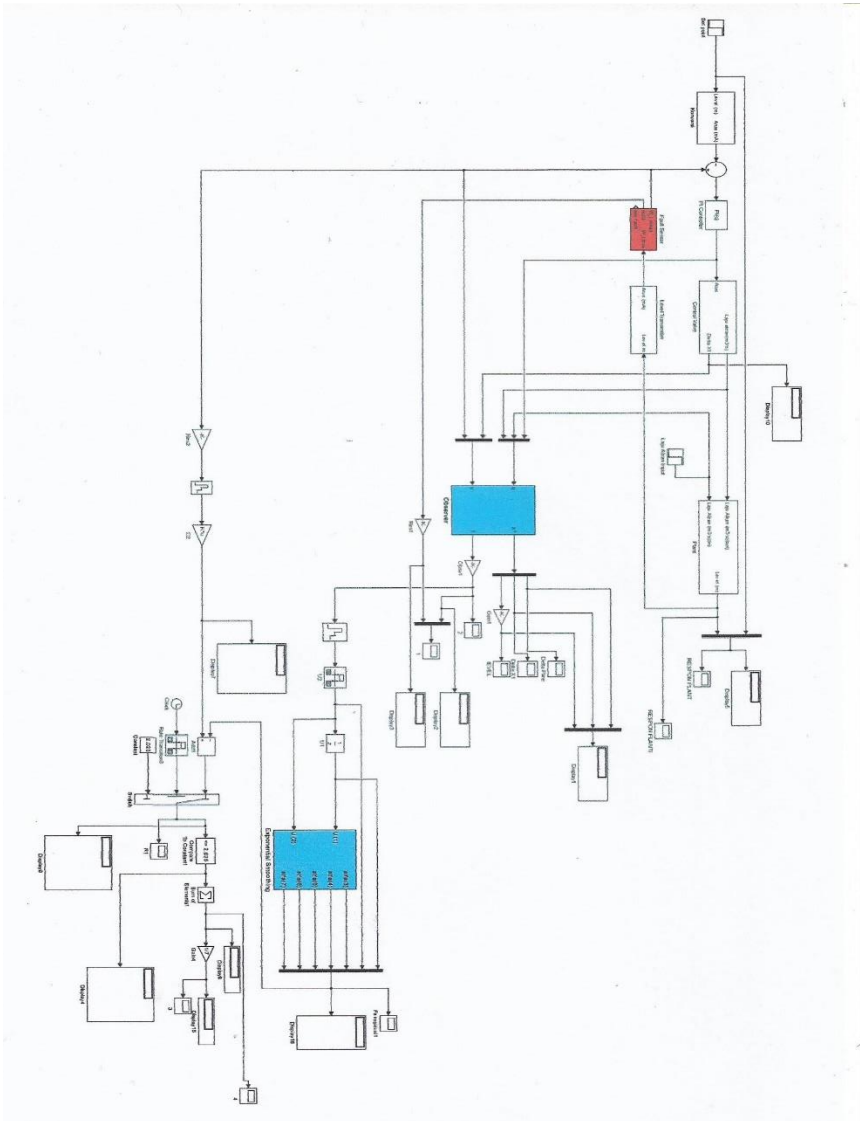
## B.7 Algoritma Exponential Smoothing



# B.8 Algoritma Reliabilty Prediction



**B.9 Pemodelan Sistem Pengendalian *Level Separator, Observer*,  
Prediksi Kesalahan Sensor dan Prediksi Keandalan  
secara *Real-Time*.**



**LAMPIRAN C**  
**LISTING PROGRAM MATLAB R2014a m.file**

```
clear
clc
A=[-5 0 0;4.165 -0.5 0;0 0 0]
B=[3.75 0 0;0 0 0;0 -0.856*11.22*0.0025
0.856*11.22*0.0025]
C=[0 1 0; 0 0 1]
D=zeros(2,3);
Ts = 0.01;
Fs =[0;1];
Abar=eye(2)
n = length(A)
p = size(C,1)
m = size(B,2)
fs = size(Fs,2)P
Aa = [A zeros(3,2);C -eye(2)]
Ba = [B;zeros(2,3)]
Ca = [zeros(2,3) eye(2)]
Ea = [zeros(n,fs); Abar*Fs]
Atild = [Aa Ea; zeros(1,5) zeros
(1,1)];
Ctild=[Ca zeros(2,1)]
Bo = [zeros(n+p,fs); eye(fs)];
Qf = eye(n+p+fs);
Qe = eye(fs);

setlmis([])
P = lmivar(1,[n+p+fs 1]);
G = lmivar(2,[n+p+fs p]);
gamma = lmivar(1,[1 0]);

lmitem([1 1 1 P],1,Atild,'s')
lmitem([1 1 1 G],-1,Ctild,'s')
lmitem([1 1 1 0],Qf)
lmitem([1 1 2 P],1,Bo)
lmitem([1 2 2 gamma],-1,1)
lmitem([-2 1 1 P],1,1)
```

```
LMI_s = getlmis;  
[t,xdec]=feasp(LMI_s);  
P = dec2mat(LMI_s,xdec,1);  
G = dec2mat(LMI_s,xdec,2);  
gamma = dec2mat(LMI_s,xdec,3);  
Ktild = inv(P)*G;  
K = Ktild(1:n+p, :);  
L = Ktild(n+p+1:n+p+fs, :);  
ars=eig(Atild-Ktild*Ctild)
```

```
%Exponential Smoothing
```

```
a = 0.9;  
b = 0.9;  
Tp= 1;  
T = 0.01;  
end
```

## **BIODATA PENULIS**



Nama Aloys Yantri Martina, dilahirkan di Sidikalang, tanggal 28 Januari 1996. Alamat penulis yaitu di Jalan Ampera No.36.. Penulis telah menyelesaikan pendidikan di SD ST Yosef Sidikalang, pendidikan di SMP ST Petrus Sidikalang, pendidikan di SMAK Yos Sudarso Kepanjen, Diploma 3 Metrologi dan Instrumentasi Fakultas Vokasi di Institut Teknologi Sepuluh Nopember Surabaya dan sedang menempuh pendidikan S1 Teknik Fisika FTI di Institut Teknologi Sepuluh Nopember Surabaya hingga sekarang. Bagi pembaca yang memiliki kritik, saran atau ingin berdiskusi lebih lanjut mengenai tugas akhir ini maka dapat menghubungi penulis melalui email: [alloysyantrimartina@gmail.com](mailto:alloysyantrimartina@gmail.com).