

**PERANCANGAN DAN APLIKASI ALGORITMA
ADAPTIF LEVEL SUPERVISI PADA PENGENDALIAN
PRESSURE PROCESS RIG FEEDBACK 38-714 UNTUK
MENGATASI FENOMENA *BURSTING* PADA *SET-POINT*
KURANG TEREKSITASI**

SKRIPSI

Oleh :

IYUNG

04 04 03 052 Y



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

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

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul :

**PERANCANGAN DAN APLIKASI ALGORITMA ADAPTIF LEVEL
SUPERVISI PADA PENGENDALIAN *PRESSURE PRICESS RIG FEEDBACK*
38-714 UNTUK MENGATASI FENOMENA *BURSTING* PADA *SET-POINT*
KURANG TEREKSITASI**

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

Depok, Juni 2008

Iyung

NPM 04 04 03 052 Y

PENGESAHAN

Skripsi dengan judul :

**PERANCANGAN DAN APLIKASI ALGORITMA ADAPTIF LEVEL
SUPERVISI PADA PENGENDALIAN *PRESSURE PRICESS RIG FEEDBACK*
38-714 UNTUK MENGATASI FENOMENA *BURSTING* PADA *SET-POINT*
KURANG TEREKSITASI**

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

Depok, Juni 2008
Dosen Pembimbing

Ir. Aries Subiantoro, M.Sc.

NIP 132 137 887

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada :

Ir. Aries Subiantoro, M.Sc.

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



Iyung
NPM 04 04 03 052 Y
Departemen Teknik Elektro

Dosen Pembimbing
Aries Subiantoro, S.T., M.Sc,

PERANCANGAN DAN APLIKASI ALGORITMA ADAPTIF LEVEL SUPERVISI PADA PENGENDALIAN *PRESSURE PROCESS RIG FEEDBACK 38-714* UNTUK MENGATASI FENOMENA BURSTING PADA SET-POINT KURANG TEREKSITASI

ABSTRAK

Pengendali adaptif pada *Pressure Process Rig Feedback 38-714* dengan mekanisme adaptasi, yaitu faktor pelupa lebih kecil dari 1, menunjukkan performa yang baik jika *set-point* yang diberikan cukup tereksitasi. Pada sistem dengan *set-point* kurang tereksitasi, pengendalian adaptif dengan mekanisme adaptasi menghasilkan fenomena Bursting, yaitu fenomena di mana sistem tidak terkontrol akibat gagalnya kerja estimator. Untuk mengatasi hal tersebut dirancanglah suatu algoritma supervisi. Algoritma supervisi ini berfungsi untuk menata kinerja estimator dan sintesa pengendali dan untuk memastikan lup pengendalian selalu stabil.

Pada skripsi ini, algoritma supervisi memantau besaran sinyal rata-rata dan variansi kesalahan prediksi, autokorelasi sinyal kendali, variansi parameter model, dan letak kutub parameter model terestimasi. Besaran – besaran ini dihitung secara rekursif (setiap pencuplikan) dari besaran – besaran yang dihasilkan oleh pengendali adaptif. Algoritma supervisi ini diaplikasikan pada pengendalian adaptif *Pressure Process Rig (Feedback 38-714)*.

Dari uji eksperimen terbukti bahwa pengendalian adaptif dengan supervisi memberikan hasil pengendalian yang lebih baik dibandingkan dengan pengendalian adaptif tanpa supervisi. Hal tersebut dapat terlihat dari tidak adanya fenomena bursting yang terjadi pada pengendali adaptif dengan supervisi yang mempunyai mekanisme adaptasi dan set-point kurang tereksitasi.

Kata Kunci : pengendali adaptif, bursting, supervisi, rekursif.

Iyung
NPM 04 04 03 052 Y
Departemen Teknik Elektro

Dosen Pembimbing
Aries Subiantoro, S.T., M.Sc,

**DESIGN AND APPLICATION ADAPTIVE CONTROL WITH
SUPERVISION LEVEL ALGORITHM ON *PRESSURE PROCESS RIG
FEEDBACK 38-714* CONTROLLING TO COPE WITH BURSTING
PHENOMENON ON LESS EXCITATION SET-POINT**

ABSTRACT

Adaptive controlling on *Pressure Process Rig Feedback 38-714* with adaptation mechanism, that has forgetting factor less than 1, shows good performance if the set-point given excite enough. In the system with less excitation, adaptive control with adaptation mechanism results Bursting phenomenon, that is phenomenon where system can't be controlled anymore because of the estimator failure. Supervision algorithm is designed to cope with that problem. This supervision algorithm rule is organizing estimator's work and controller design to make sure that control closed-loop always stable.

In this bachelor thesis, supervision algorithm monitors some parameters, there are mean and variance of prediction error signal, autocorrelation of control signal, variance of model parameter, and place of estimated model poles. These parameters are recursive calculated (every sample time) from adaptive control parameter yielded. This supervision algorithm is implemented on *Pressure Process Rig (Feedback 38-714)* with adaptive control.

From experiment test, it is proved that adaptive control with supervision gives better control result than adaptive control without supervision. It can be seen from no Bursting phenomenon that happened in adaptive control with supervision level that has adaptation mechanism and less excitation set-point.

Kata Kunci : adaptive control, bursting, supervision, recursive.

DAFTAR ISI

<u>PERNYATAAN KEASLIAN SKRIPSI</u>	ii
<u>PENGESAHAN</u>	iii
<u>UCAPAN TERIMA KASIH</u>	iv
<u>ABSTRAK</u>	v
<u>ABSTRACT</u>	vi
<u>DAFTAR ISI</u>	vii
<u>DAFTAR GAMBAR</u>	x
<u>DAFTAR TABEL</u>	xiii
BAB 1 PENDAHULUAN.....	1
1.1 LATAR BELAKANG.....	1
1.2 TUJUAN [2]	1
1.3 PEMBatasan MASALAH	2
1.4 SISTEMATIKA PENULISAN	2
BAB 2 LANDASAN TEORI	3
2.1 VOLTAGE-TO-CURRENT CONVERTER (V/I CONVERTER).....	3
2.2 CURRENT-TO-VOLTAGE CONVERTER (I/V CONVERTER).....	4
2.3 METODE KUADRAT TERKECIL REKURSIF (<i>RECURSIVE LEAST SQUARE</i>).....	5
2.4 METODE KUADRAT TERKECIL REKURSIF DENGAN FAKTOR PELUPAAN	9
2.5 METODE PENEMPATAN KUTUB (<i>POLE PLACEMENT</i>)	11
2.6 ATURAN KENDALI KENAIKAN (<i>INCREMENTAL CONTROL</i>)	13
2.7 LEVEL SUPERVISI	14
BAB 3 PERANCANGAN SISTEM.....	18
3.1 DESKRIPSI SISTEM <i>PRESSURE PROCESS RIG</i> [1].....	18
3.1.1 Peralatan Pendukung	20
3.1.1.1 Process Interface (Feedback 38-200)	20
3.1.1.2 V/I Converter.....	21
3.1.1.3 Air Compressor	22
3.1.1.4 Data Acquisition.....	22
3.1.1.5 Personal Computer	22

3.1.2	Interkoneksi Alat	22
3.2	SPEKIFIKASI PERANGKAT LUNAK YANG DIGUNAKAN.....	23
3.3	SISTEM KENDALI SWATALA PADA <i>PRESSURE PROCESS RIG</i>	23
3.4	DISAIN FUNGSI SUPERVISI ESTIMATOR DAN PENGENDALI	24
3.4.1	Pemantauan Besaran-Besaran dalam Pengendali Adaptif.....	24
3.4.1.1	Pemantauan Eksitasi Sinyal Masukan Sistem (Sinyal Kendali).....	24
3.4.1.2	Pemantauan Parameter Model Terestimasi	25
3.4.1.3	Pemantauan Sinyal Kesalahan Prediksi (<i>Prediction Error</i>)	25
3.4.2	DISAIN LOGIKA SUPERVISI.....	25
3.4.3	ALGORITMA REKURSIF FUNGSI SUPERVISI	29
3.4.3.1	Rata – rata Rekursif (<i>mean</i>).....	29
3.4.3.2	Variansi Rekursif.....	30
3.4.3.3	Matrix Informasi H.....	31
3.4.3.4	Perhitungan Kutub Model	33
3.4.4	VALIDASI ALGORITMA REKURSIF	33
3.4.4.1	Validasi Algoritma Rata-rata dan Variansi Rekursif.....	33
3.4.4.2	Validasi Algoritma Determinan Matriks $H_{2 \times 2}$ Rekursif	34
3.4.4.3	Validasi Algoritma Perhitungan Kutub Model.....	35
BAB 4	PEMBAHASAN DAN ANALISA	37
4.1	SIMULASI PENERAPAN SIMULASI PENERAPAN LEVEL SUPERVISI UNTUK MENGATASI FENOMENA BURSTING.....	37
4.1.1	Simulasi Penerapan Level Supervisi pada <i>Set-Point</i> Tereksitasi	37
4.1.2	Simulasi Penerapan Level Supervisi pada <i>Set-Point</i> Kurang Tereksitasi	43
4.2	FENOMENA <i>BURSTING</i> PADA UJI EKSPERIMEN PENGENDALI SWATALA <i>PRESSURE PROCESS RIG</i>	49
4.3	UJI EKSPERIMEN LEVEL SUPERVISI PADA <i>SET-POINT</i> KURANG TEREKSITASI.....	55
BAB 5	KESIMPULAN	61
	DAFTAR ACUAN	62
	DAFTAR PUSTAKA.....	63
	LAMPIRAN	64

DAFTAR GAMBAR

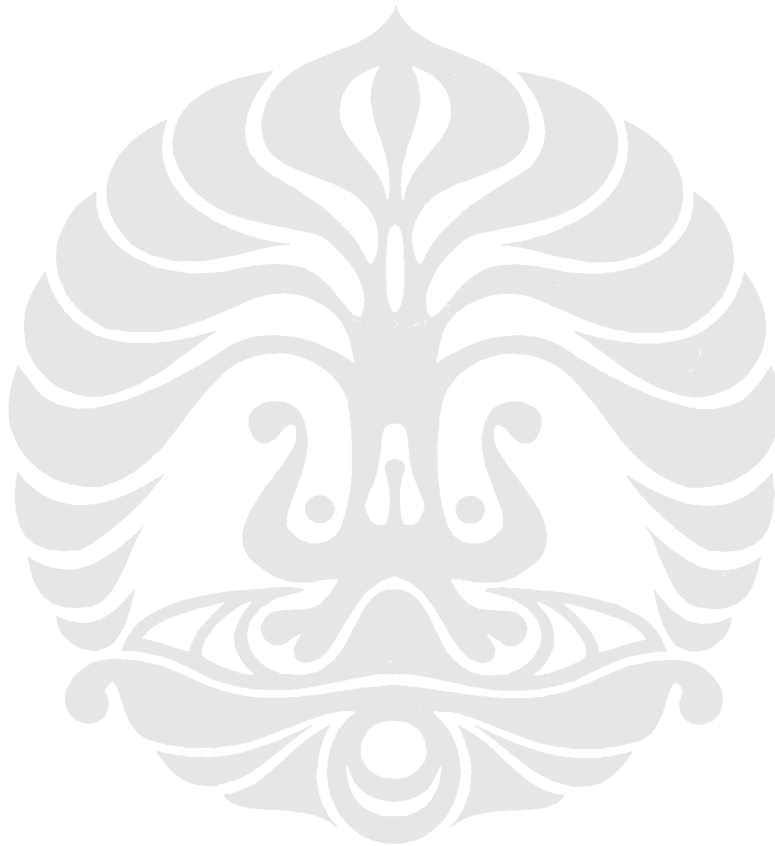
Gambar 2.1	<i>Voltage-to-current converter</i>	3
Gambar 2.2	<i>Current-to-voltage converter</i>	4
Gambar 2.3	Diagram blok Sistem Lingkar Tertutup Pengendali Penempatan Kutub 11	
Gambar 2.4	Konsep Aksi Kendali Kenaikan	13
Gambar 2.5	Sistem kendali adaptif berbasis supervisi.....	15
Gambar 3.1	Pressure Process Rig (38-714, Feedback Instruments Limited).....	18
Gambar 3.2	Koneksi pada Process Interface (Feedback 38-200).....	20
Gambar 3.3	<i>Current-to-voltage converter</i>	20
Gambar 3.4	<i>Voltage-to-Current Converter</i>	21
Gambar 3.5	Interkoneksi komponen-komponen perangkat keras	23
Gambar 3.6	Diagram alir level supervisi.....	26
Gambar 3.7	Pantauan determinan matriks H_{22}	27
Gambar 3.8	Pantauan rata-rata dan variansi <i>prediction error</i>	27
Gambar 3.9	Variansi norm parameter yang sesuai untuk penalaan pengendali.....	28
Gambar 3.10	Letak kutub stabil dalam bidang z	29
Gambar 3.11	Perubahan variansi karena perubahan rata-rata.....	30
Gambar 3.12	Diagram blok Simulink untuk validasi mean dan variansi rekursif	33
Gambar 3.13	Diagram blok Simulink untuk validasi determinan $H_{2 \times 2}$	34
Gambar 3.14	Diagram blok Simulink untuk validasi perhitungan pole.....	35
Gambar 4.1	Simulasi estimasi parameter model tanpa supervisi pada <i>set-point</i> tereksitasi.....	37
Gambar 4.2	Simulasi penalaan parameter pengendali tanpa supervisi pada <i>set-point</i> tereksitasi.....	38
Gambar 4.3	Simulasi penalaan perbandingan keluaran dan <i>set-point</i> tereksitasi tanpa supervisi.....	38
Gambar 4.4	Simulasi perbandingan keluaran dan sinyal kendali tanpa supervisi pada <i>set-point</i> tereksitasi.....	39
Gambar 4.5	Simulasi estimasi parameter model dengan supervisi pada <i>set-point</i> tereksitasi.....	40
Gambar 4.6	Simulasi penalaan parameter pengendali dengan supervisi pada <i>set-point</i> tereksitasi.....	40

Gambar 4.7	Simulasi determinan H_{22} dengan supervisi pada <i>set-point</i>	41
Gambar 4.8	Simulasi <i>norm</i> variansi parameter model dengan supervisi pada <i>set-point</i>	41
Gambar 4.9	Letak kutub model pada simulasi dengan supervisi pada <i>set-point</i> tereksitasi	42
Gambar 4.10	Simulasi perbandingan keluaran dengan <i>set-point</i> tereksitasi dengan supervisi.....	42
Gambar 4.11	Simulasi estimasi parameter model tanpa supervisi pada <i>set-point</i> kurang tereksitasi	43
Gambar 4.12	Simulasi parameter pengendali tanpa supervisi pada <i>set-point</i> kurang tereksitasi	44
Gambar 4.13	Sinyal kendali tanpa supervisi pada simulasi dengan <i>set-point</i> kurang tereksitasi	44
Gambar 4.14	Simulasi perbandingan keluaran dengan <i>set-point</i> kurang tereksitasi tanpa supervisi	45
Gambar 4.15	Simulasi estimasi parameter model dengan supervisi pada <i>set-point</i> kurang tereksitasi	45
Gambar 4.16	Simulasi parameter pengendali dengan supervisi pada <i>set-point</i> kurang tereksitasi	46
Gambar 4.17	Simulasi determinan matriks H_{22} dengan supervisi pada <i>set-point</i> kurang tereksitasi	46
Gambar 4.18	Sinyal kendali dengan supervisi pada <i>set-point</i> kurang tereksitasi	47
Gambar 4.19	Simulasi perbandingan keluaran dan <i>set-point</i> kurang tereksitasi dengan supervisi.....	47
Gambar 4.20	Simulasi <i>norm</i> variasi parameter model dengan supervisi pada <i>set-point</i> kurang tereksitasi	48
Gambar 4.21	Simulasi <i>mean</i> dan variansi prediction error dengan supervisi pada <i>set-</i> <i>point</i> kurang tereksitasi.....	49
Gambar 4.22	Perbandingan keluaran dan <i>set-point</i> pada kondisi <i>bursting</i>	49
Gambar 4.23	Perbandingan keluaran dan <i>set-point</i> pada kondisi <i>bursting</i>	50
Gambar 4.24	Estimasi parameter pada kondisi <i>bursting</i>	51
Gambar 4.25	Letak kutub hasil estimasi parameter pada kondisi <i>bursting</i>	51

Gambar 4.26	Estimasi parameter pada kondisi <i>bursting</i>	52
Gambar 4.27	Perbandingan sinyal kendali dan keluaran pada kondisi <i>bursting</i>	53
Gambar 4.28	Nilai <i>mean</i> dan variansi <i>prediction error</i> pada kondisi <i>bursting</i>	53
Gambar 4.29	Nilai Determinan matriks H_{22} pada kondisi <i>bursting</i>	54
Gambar 4.30	Norm variansi parameter model pada kondisi <i>bursting</i>	54
Gambar 4.31	Hasil eksperimen perbandingan keluaran dan <i>set-point</i> kurang tereksitasi dengan algoritma supervisi.....	55
Gambar 4.32	Hasil eksperimen parameter model dengan supervisi pada <i>set-point</i> kurang tereksitasi.....	56
Gambar 4.33	Hasil eksperimen determinan H_{22} dengan supervisi pada <i>set-point</i> kurang tereksitasi.....	57
Gambar 4.34	Hasil eksperimen determinan H_{22} pada saat perubahan <i>set-point</i> dengan supervisi pada <i>set-point</i> kurang tereksitasi.....	57
Gambar 4.35	Hasil perhitungan parameter pengendali dengan supervisi pada <i>set-point</i> kurang tereksitasi.....	58
Gambar 4.36	Sinyal kendali dengan supervisi pada <i>set-point</i> kurang tereksitasi	58
Gambar 4.37	Hasil eksperimen norm variansi parameter model dengan supervisi pada <i>set-point</i> kurang tereksitasi.....	59
Gambar 4.38	Hasil eksperimen <i>mean</i> dan variansi <i>prediction error</i> dengan supervisi pada <i>set-point</i> kurang tereksitasi	60

DAFTAR TABEL

Tabel 3.1 Perbandingan mean dan variansi rekursif dengan MATLAB	34
Tabel 3.2 Perbandingan perhitungan determinan matriks H_{22} rekursif dengan MATLAB	35
Tabel 3.3 Perbandingan perhitungan letak pole rekursif dengan MATLAB.....	36



BAB 1

PENDAHULUAN

1.1 LATAR BELAKANG

Pengendali konvensional dengan parameter yang tetap tidak dapat mengatasi sistem yang mengalami perubahan parameter seiring berjalannya waktu. Hal ini menyebabkan unjuk kerja sistem lingkaran tertutup akan mengalami penurunan bahkan dapat mengakibatkan ketidakstabilan.

Pengendali adaptif yang terdiri dari kombinasi estimasi parameter model dan penalaan parameter pengendali dapat mengatasi masalah tersebut. *Forgetting factor* pada estimator parameter model dapat menghasilkan mekanisme adaptasi yang terus – menerus. Hal ini dapat dilihat pada skripsi Kumar Simanjuntak [1] yang menerapkan pengendali adaptif dengan *forgetting factor* pada *Pressure Process Rig (Feedback 38-714)*. Pengendali adaptif bekerja baik jika set-point yang diinginkan mengalami perubahan pada waktu yang cepat atau disebut juga kondisi cukup tereksitasi.

Sedangkan pada kondisi kurang tereksitasi, yaitu kondisi di mana *set-point* cenderung konstan, pengendali adaptif dengan *forgetting factor* yang cukup kecil akan menyebabkan ketidakstabilan. Hal ini disebut juga kondisi *bursting*. Pada kondisi ini, keluaran tidak akan mencapai suatu nilai tunak.

Untuk itu diperlukan level supervisi pada sistem kendali adaptif yang dapat mengatasi masalah *Bursting* [4]. Level ini berfungsi untuk mengambil keputusan perlu tidaknya perhitungan suatu algoritma, estimasi atau penalaan pengendali, dilakukan berdasarkan pemantauan sinyal – sinyal kendali adaptif yang ada.

1.2 TUJUAN

Skripsi ini merupakan suatu riset yang dilakukan untuk mengaplikasikan dan membuktikan teori kinerja level supervisi pada pengendalian adaptif pada suatu sistem nyata. Tujuan dari skripsi ini adalah:

- Merancang algoritma level supervisi pada pengendali adaptif yang berfungsi untuk menghindari terjadinya *bursting* dengan cara membuat fungsi manajemen estimator dan manajemen pengendali
- Mengimplementasikan pengendali adaptif berbasis level supervisi pada pengendalian sistem *Pressure Process Rig (Feedback 38-714)*.

1.3 PEMBATASAN MASALAH

Dalam skripsi ini, sistem yang dikendalikan adalah sistem *Single Input Single Output (SISO)*, yaitu *Pressure Process Rig (38-714, Feedback Instruments Limited)* yang sudah dilengkapi sistem kendali swatara dengan metode RLS (*Recursive Least Square*) orde 2 dengan factor pelupa sebagai metode estimator parameter model dan metode Penempatan Kutub (*Pole Placement*) sebagai metode penalaan parameter pengendali. Algoritma supervisi yang dikerjakan dalam skripsi ini tanpa proses pemfilteran dari sinyal – sinyal pengendali adaptif dan tanpa model substitusi. Algoritma ini hanya diaplikasikan dengan faktor pelupa lebih kecil dari 1 dan *set-point* yang cenderung konstan.

1.4 SISTEMATIKA PENULISAN

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

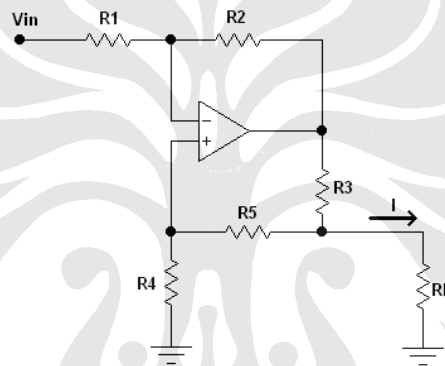
Bab satu merupakan pendahuluan yang berisi latar belakang, tujuan, pembatasan masalah dan sistematika penulisan skripsi. Bab dua membahas mengenai landasan teori mengenai penambahan algoritma supervisi pada suatu pengendali adaptif. Bab tiga merupakan pembahasan tentang penjelasan proses dan perancangan level supervisi. Bab empat berisi simulasi, uji eksperimen, dan analisa hasil penambahan level supervisi pada suatu pengendali adaptif. Bab lima merupakan kesimpulan dari keseluruhan pembahasan dalam skripsi ini.

BAB 2

LANDASAN TEORI

2.1 VOLTAGE-TO-CURRENT CONVERTER (V/I CONVERTER) [1]

Sebagian besar alat – alat kendali proses menggunakan sinyal arus untuk mentransmisikan sinyal, maka dibutuhkan suatu alat pengubah sinyal tegangan menjadi sinyal arus yang linear untuk menghubungkan peralatan kendali proses ini dengan komputer, yang menggunakan sinyal tegangan untuk berinteraksi dengan modul di luarnya. Secara sederhana, rangkaian *V/I Converter* terlihat seperti pada Gambar 2.1.



Gambar 2.1. Voltage-to-current converter

Suatu analisis yang dilakukan pada rangkaian ini menunjukkan hubungan antara besar arus dan tegangan yang dapat ditulis sebagai

$$I = \frac{R_2}{R_1 R_2} V_{in} \quad (2.1)$$

dengan besarnya resistansi dipilih sedemikian sehingga

$$R_1 (R_3 + R_5) = R_2 R_4 \quad (2.2)$$

Akan tetapi, pada konfigurasi rangkaian seperti ini terdapat suatu batas maksimum beban dan maksimum arus yang dapat dialirkan. Hal ini berhubungan dengan adanya tegangan keluaran pada op amp. Suatu analisis yang dilakukan pada rangkaian ini menunjukkan hubungan antara besarnya resistansi beban maksimum dan arus maksimum, yang ditunjukkan dalam Persamaan (2.3)

$$R_{ml} = \frac{(R_4 + R_5) \left[\frac{V_{sat}}{I_m} - R_3 \right]}{R_3 + R_4 + R_5} \quad (2.3)$$

dengan: R_{ml} = Resistansi beban maksimum (ohm)
 V_{sat} = Tegangan saturasi op amp (volt)
 I_m = Arus maksimum yang mengalir (ampere)

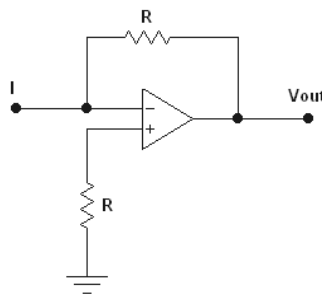
Suatu pendekatan yang lebih jauh pada Persamaan (2.3) menunjukkan bahwa resistansi maksimum beban R_{ml} selalu lebih kecil daripada V_{sat} / I_m . Resistansi minimum beban adalah nol.

2.2 CURRENT-TO-VOLTAGE CONVERTER (I/V CONVERTER) [1]

Standar sinyal yang banyak digunakan dalam perindustrian adalah sinyal arus, sedangkan untuk kemudahan pengolahan sinyal, digunakan komputer yang menggunakan sinyal tegangan dalam prosesnya. Untuk mengakomodasi pengiriman sinyal arus dari sistem menjadi sinyal tegangan ke komputer, diperlukan suatu *Current-to-voltage converter (I/V Converter)*. Secara sederhana, rangkaian *I/V Converter* dapat terlihat seperti Gambar 2.2. Besarnya tegangan keluaran V_{out} sebanding dengan besarnya resistor R dan arus I yang mengalir, sesuai dengan persamaan

$$V_{out} = IR \quad (2.4)$$

Resistor R yang terdapat pada terminal *non-inverting Operational Amplifier* (Op Amp) dimaksudkan untuk menjaga kestabilan temperatur konfigurasi rangkaian.



Gambar 2.2. *Current-to-voltage converter*

2.3 METODE KUADRAT TERKECIL REKURSIF (*RECURSIVE LEAST SQUARE*) [1]

Metode kuadrat terkecil adalah metode identifikasi secara online. Identifikasi model dilakukan setiap pencuplikan data baru. Estimasi ini bersifat iteratif, yaitu data hasil pencuplikan yang terakhir ditambahkan pada kumpulan data lama.

Perumusan metode kuadrat terkecil rekursif dapat dilakukan dengan membandingkan hasil estimasi yang dilakukan atas data dari waktu 1 sampai t , terhadap estimasi yang dilakukan atas data dari waktu 1 sampai $t+1$. Nilai-nilai koefisien parameter yang dinyatakan dalam vektor untuk estimasi parameter dari waktu 1 sampai t :

$$\underline{\theta}(t) = [\underline{X}(t) \cdot \underline{X}^T(t)]^{-1} \cdot \underline{X}(t) \cdot \underline{Y}(t) \quad (2.5)$$

dengan $\underline{X}^T(t) = \begin{pmatrix} x^T(1) \\ x^T(2) \\ \vdots \\ x^T(t) \end{pmatrix}$, dan $\underline{Y}(t) = \begin{pmatrix} y(1) \\ y(2) \\ \vdots \\ y(t) \end{pmatrix}$ (2.6)

Jika komponen-komponen dalam Persamaan (2.5) dinyatakan dengan:

$$\begin{aligned} [\underline{X}(t) \cdot \underline{X}^T(t)]^{-1} &= \underline{P}(t) \\ \underline{X}(t) \cdot \underline{X}^T(t) &= \underline{B}(t) \end{aligned} \quad (2.7)$$

Maka Persamaan (2.5) menjadi:

$$\underline{\theta}(t) = \underline{P}(t) \cdot \underline{B}(t) \quad (2.8)$$

Kemudian untuk data hasil pencuplikan yang baru pada saat $t+1$, maka Persamaan (2.6) menjadi:

$$\underline{X}^T(t+1) = \begin{pmatrix} x^T(1) \\ x^T(2) \\ \vdots \\ x^T(t) \\ \text{-----} \\ x^T(t+1) \end{pmatrix} = \begin{pmatrix} X^T(t) \\ \text{-----} \\ x^T(t+1) \end{pmatrix}$$

$$\underline{y}(t+1) = \begin{pmatrix} y(1) \\ y(2) \\ \vdots \\ y(t) \\ \text{-----} \\ y(t+1) \end{pmatrix} = \begin{pmatrix} Y(t) \\ \text{-----} \\ Y(t+1) \end{pmatrix} \quad (2.9)$$

Sehingga estimasi parameter pada waktu $t+1$ adalah:

$$\underline{\theta}(t+1) = [\underline{X}(t+1)\underline{X}^T(t+1)]^{-1} \cdot \underline{X}(t+1)\underline{Y}(t+1) \quad (2.10)$$

berdasarkan persamaan-persamaan di atas, maka sekarang dapat ditulis:

$$\begin{aligned} [\underline{X}(t+1)\underline{X}^T(t+1)] &= [X(t) \quad x(t+1)] \begin{pmatrix} X^T(t) \\ \text{-----} \\ x^T(t+1) \end{pmatrix} \\ &= \underline{X}(t)\underline{X}^T(t) + \underline{x}(t+1)\underline{x}^T(t+1) \end{aligned} \quad (2.11a)$$

dan

$$\begin{aligned} \underline{X}(t+1)\underline{Y}(t+1) &= [X(t) \quad x(t+1)] \begin{pmatrix} Y(t) \\ \text{-----} \\ y(t+1) \end{pmatrix} \\ &= \underline{X}(t)\underline{Y}(t) + \underline{x}(t+1)y(t+1) \end{aligned} \quad (2.12b)$$

Berdasarkan Persamaan (2.10) dan (2.11), dapat dilihat bahwa $\underline{\theta}$ dapat diperbaharui dengan faktor $x(t+1)$ dan $y(t+1)$, sehingga dengan mengacu pada Persamaan (2.8), maka $\underline{\theta}$ pada waktu $t+1$ dapat dinyatakan dengan:

$$\underline{\theta}(t+1) = \underline{P}(t+1)\underline{B}(t+1) \quad (2.13)$$

Jika Persamaan (2.11a) dinyatakan dalam Persamaan (2.7) maka akan menjadi:

$$\underline{P}(t+1)^{-1} = \underline{P}(t)^{-1} + \underline{x}(t+1)\underline{x}^T(t+1) \quad (2.14)$$

atau

$$\underline{P}(t+1) = [\underline{P}(t)^{-1} + \underline{x}(t+1)\underline{x}^T(t+1)]^{-1} \quad (2.15)$$

dan

$$\underline{B}(t+1) = \underline{B}(t) + \underline{x}(t+1)y(t+1) \quad (2.16)$$

Jika bentuk invers dalam Persamaan (2.14) disederhanakan menurut dilemma invers matriks:

$$\begin{aligned}
(\underline{A} + \underline{B}\underline{C}\underline{D})^{-1} &= \underline{A}^{-1} - \underline{A}^{-1}\underline{B}(\underline{C}^{-1} + \underline{D}\underline{A}^{-1}\underline{B})^{-1}\underline{D}\underline{A}^{-1} \\
&= \underline{A}^{-1}[\underline{I} - \underline{B}(\underline{C}^{-1} + \underline{D}\underline{A}^{-1}\underline{B})^{-1}\underline{D}\underline{A}^{-1}]
\end{aligned} \tag{2.17}$$

dengan: $\underline{A} = \underline{P}(t)^{-1}$

$$\underline{B} = \underline{x}(t+1)$$

$$\underline{C} = 1$$

$$\underline{D} = \underline{x}^T(t+1)$$

maka Persamaan (2.14) menjadi:

$$\underline{P}(t+1) = \underline{P}(t)[\underline{I} - \underline{x}(t+1)\{1 + \underline{x}^T(t+1)\underline{P}(t)\underline{x}(t+1)\}^{-1}\underline{x}^T(t+1)\underline{P}(t)] \tag{2.18}$$

atau

$$\underline{P}(t+1) = \left[\underline{I} - \frac{\underline{P}(t)\underline{x}(t+1)\underline{x}^T(t+1)}{1 + \underline{x}^T(t+1)\underline{P}(t)\underline{x}(t+1)} \right] \underline{P}(t) \tag{2.19}$$

Dari Persamaan (2.18) terlihat bahwa harga $\underline{P}(t+1)$ dapat langsung diperoleh dari $\underline{P}(t)$ dan data baru pada saat $t+1$. Matriks $\underline{P}(t)$ disebut matriks kovarians dan memegang peranan penting dalam proses estimasi parameter.

Seperti halnya matriks kovarians, vektor $\underline{\theta}(t+1)$ juga dapat langsung diperoleh dari $\underline{\theta}(t)$ dengan mendefinisikan variabel kesalahan $e(t+1)$:

$$e(t+1) = y(t+1) - \underline{x}^T(t+1)\underline{\theta}(t) \tag{2.20}$$

$$y(t+1) = e(t+1) + \underline{x}^T(t+1)\underline{\theta}(t) \tag{2.21}$$

Jika Persamaan (2.20) di substitusikan ke dalam Persamaan (2.15) maka menjadi:

$$\begin{aligned}
\underline{B}(t+1) &= \underline{B}(t) + \underline{x}(t+1)[e(t+1) + \underline{x}^T(t+1)\underline{\theta}(t)] \\
&= \underline{B}(t) + \underline{x}(t+1)e(t+1) + \underline{x}(t+1)\underline{x}^T(t+1)\underline{\theta}(t)
\end{aligned} \tag{2.22}$$

Berdasarkan Persamaan (2.13):

$$\underline{x}(t+1)\underline{x}^T(t+1) = \underline{P}(t+1)^{-1} - \underline{P}(t)^{-1} \tag{2.23}$$

dengan mensubstitusi Persamaan (2.8) dan (2.22) ke Persamaan (2.21):

$$\begin{aligned}
\underline{B}(t+1) &= \underline{B}(t) + \underline{x}(t+1)e(t+1) + [\underline{P}(t+1)^{-1} - \underline{P}(t)^{-1}]\underline{P}(t)\underline{B}(t) \\
&= [1 + \underline{P}(t+1)^{-1}\underline{P}(t) - 1]\underline{B}(t) + \underline{x}(t+1)e(t+1) \\
&= \underline{P}(t+1)^{-1}\underline{P}(t)\underline{B}(t) + \underline{x}(t+1)e(t+1)
\end{aligned} \tag{2.24}$$

kedua ruas dikalikan dengan $\underline{P}(t+1)$ sehingga:

$$\underline{P}(t+1)\underline{B}(t+1) = \underline{P}(t)\underline{B}(t) + \underline{P}(t+1)\underline{x}(t+1)e(t+1) \tag{2.25}$$

Jika Persamaan (2.24) dinyatakan dengan Persamaan (2.8) dan (2.12):

$$\underline{\theta}(t) = \underline{\theta}(t-1) + \underline{P}(t)\underline{x}(t)\underline{e}(t) \quad (2.26)$$

Sehingga langkah-langkah memperbaharui vektor dapat dinyatakan sebagai berikut:

1. Menyusun vektor $\underline{x}(t+1)$ berdasarkan data masukan dan keluaran yang baru
2. Menghitung kesalahan $e(t+1)$ dengan persamaan:

$$e(t+1) = y(t+1) - \underline{x}^T(t+1)\underline{\theta}(t) \quad (2.27)$$

atau

$$e(t) = y(t) - \underline{x}^T(t)\underline{\theta}(t-1) \quad (2.28)$$

3. Menghitung $\underline{P}(t+1)$ dengan persamaan:

$$\underline{P}(t+1) = \left[\underline{I} - \frac{\underline{P}(t)\underline{x}(t+1)\underline{x}^T(t+1)}{1 + \underline{x}^T(t+1)\underline{P}(t)\underline{x}(t+1)} \right] \underline{P}(t) \quad (2.29)$$

atau

$$\underline{P}(t) = \left[\underline{I} - \frac{\underline{P}(t-1)\underline{x}(t)\underline{x}^T(t)}{1 + \underline{x}^T(t)\underline{P}(t-1)\underline{x}(t)} \right] \underline{P}(t-1) \quad (2.30)$$

4. Menghitung $\underline{\theta}(t+1)$ dengan persamaan:

$$\underline{\theta}(t+1) = \underline{\theta}(t) + \underline{P}(t+1)\underline{x}(t+1)\underline{e}(t+1) \quad (2.31)$$

atau

$$\underline{\theta}(t) = \underline{\theta}(t-1) + \underline{P}(t)\underline{x}(t)\underline{e}(t) \quad (2.32)$$

5. Menunggu hasil pencuplikan berikutnya dan kembali ke langkah 1

2.4 METODE KUADRAT TERKECIL REKURSIF DENGAN FAKTOR PELUPAAN [1]

Pada metode kuadrat terkecil rekursif, adaptasi perubahan yang terjadi pada sistem agak lambat. Hal ini dikarenakan data lama yang sudah tidak signifikan dengan perubahan sekarang masih diperhitungkan dalam estimasi.

Agar mekanisme adaptasi menjadi lebih baik, maka data lama yang tidak signifikan harus dilupakan dalam perhitungan estimasi parameter yang sekarang. Hal ini menyebabkan estimator hanya menggunakan data yang *up-to-date* dalam mengestimasi parameter model. Banyaknya data yang diperhitungkan dalam

estimasi ditentukan oleh faktor pelupaan (*forgetting factor*), yang bernilai antara 0 sampai 1, dengan simbol λ (faktor pelupa). Banyaknya data yang diperhitungkan dalam estimasi dapat dihitung dengan persamaan:

$$N = \frac{1}{1 - \lambda} \quad (2.33)$$

Persamaan di atas juga dapat diartikan bahwa data akan mulai dilupakan setelah N kali pencuplikan.

Dengan adanya faktor pelupaan, maka persamaan – persamaan dalam langkah-langkah metode kuadrat terkecil rekursif menjadi:

1. Menyusun vektor $\underline{x}(t + 1)$ berdasarkan data masukan dan keluaran yang baru
2. Menghitung kesalahan $e(t + 1)$ dengan persamaan:

$$e(t + 1) = y(t + 1) - \underline{x}^T(t + 1)\underline{\theta}(t) \quad (2.34)$$

atau

$$e(t) = y(t) - \underline{x}^T(t)\underline{\theta}(t - 1) \quad (2.35)$$

3. Menghitung $\underline{P}(t + 1)$ dengan persamaan:

$$\underline{P}(t + 1) = \left[\underline{I} - \frac{\underline{P}(t)\underline{x}(t + 1)\underline{x}^T(t + 1)}{1 + \underline{x}^T(t + 1)\underline{P}(t)\underline{x}(t + 1)} \right] \frac{\underline{P}(t)}{\lambda} \quad (2.36)$$

atau

$$\underline{P}(t) = \left[\underline{I} - \frac{\underline{P}(t - 1)\underline{x}(t)\underline{x}^T(t)}{1 + \underline{x}^T(t)\underline{P}(t - 1)\underline{x}(t)} \right] \frac{\underline{P}(t - 1)}{\lambda} \quad (2.37)$$

4. Menghitung $\underline{\theta}(t + 1)$ dengan persamaan:

$$\underline{\theta}(t + 1) = \underline{\theta}(t) + \underline{P}(t + 1)\underline{x}(t + 1)e(t + 1) \quad (2.38)$$

5. Menunggu hasil pencuplikan berikutnya dan kembali ke langkah 1

Sebelum memulai proses estimasi, ada hal yang perlu dilakukan terlebih dahulu, yaitu pemberian harga awal untuk vektor $\underline{\theta}(t)$ dan $\underline{P}(t)$ yang akan terus diperbaharui. Hal ini dilaukkan karena pada saat awal estimasi, harga $\underline{\theta}(0)$ dan $\underline{P}(0)$ tidak diketahui sehingga perlu diberikan harga awal yang kira-kira sesuai dengan kondisi system.

Ada dua cara untuk memberikan harga awal estimasi, yaitu:

1. Perkiraan

Pada cara ini, harga $\underline{\theta}(0)$ diberikan dengan menebak suatu nilai yang kira-kira mendekati parameter sistem yang dikendalikan. Bila tidak ada harga yang cukup memadai, dapat diberikan nilai awal nol. Sedangkan untuk matriks kovarians $\underline{P}(t)$, harga awal yang diberikan biasanya berupa suatu matriks diagonal:

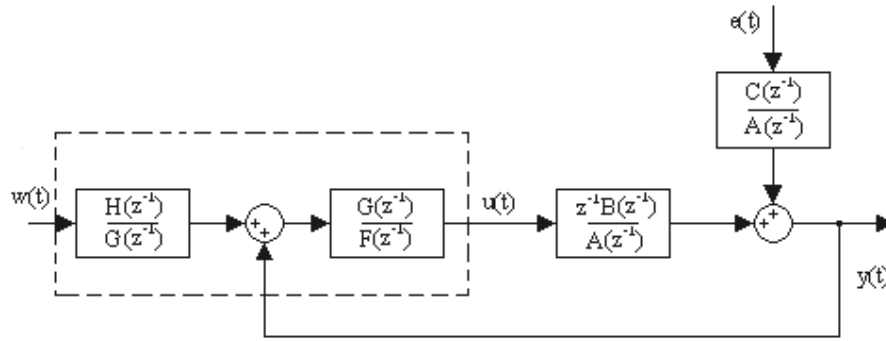
$$\underline{P}(0) = \underline{I} \cdot \alpha \quad (2.39)$$

Dengan merupakan matriks identitas sedangkan α adalah sebuah bilangan yang menyatakan seberapa besar ketidakpastian harga $\underline{\theta}(0)$ yang diberikan terhadap sifat-sifat dari sistem yang diestimasi. Harga α yang besar akan membuat estimasi berjalan cepat, sedangkan jika harga α bernilai kecil maka estimasi akan berjalan lambat. Untuk kondisi yang tidak pasti diberikan harga α yang besar.

2. Berdasarkan data sistem setelah sistem berjalan dalam selang waktu tertentu
 Pada cara ini, harga awal $\underline{\theta}(0)$ ditentukan dengan menggunakan persamaan $\underline{\theta}(t)$ berdasarkan data masukan dan keluaran dalam selang waktu tertentu, demikian pula dengan harga $\underline{P}(0)$. Jika data yang terkumpul sebanyak m , maka $\underline{\theta}(0) = \underline{\theta}(m)$ dan $\underline{P}(0) = \underline{P}(m)$. Jadi estimasi dapat dilakukan pada waktu ke $m+1$. Dengan kata lain, estimasi dilakukan setelah sistem dibiarkan berjalan untuk selang waktu tertentu.

2.5 METODE PENEMPATAN KUTUB (POLE PLACEMENT)

Setelah estimator menemukan model sistem, maka parameter pengendali dihitung dengan Metode Penempatan Kutub. Hasil estimasi dikirimkan ke blok perancangan pengendali untuk menentukan parameter pengendali yang sesuai dengan kinerja yang diinginkan. Parameter pengendali ini diperbarui setiap periode pencuplikan. Blok diagram lingkaran tertutup sistem menjadi:



Gambar 2.3. Diagram blok Sistem Lingkaran Tertutup Pengendali Penempatan Kutub

Dari gambar 2.3 dapat dilihat bahwa pengendali yang digunakan adalah sebuah regulator yang memiliki aturan kendali sebagai berikut:

$$u(t) = \frac{H.w(t) - G.y(t)}{F} \quad (2.40)$$

dengan : $F = 1 + f_1.z^{-1} + \dots + f_{nf}.z^{-nf}$

$G = g_0 + g_1.z^{-1} + \dots + g_{ng}.z^{-ng}$

$u(t)$ = Masukan sistem

$y(t)$ = Keluaran sistem

$w(t)$ = Sinyal Acuan

T adalah polinom dalam z^{-1} yang koefisien – koefisiennya diperoleh dari pole – pole fungsi alih lingkaran tertutup yang ingin dicapai oleh pengendali swatela.

$$T = 1 + t_1.z^{-1} + \dots + t_{nt}.z^{-nt} \quad (2.41)$$

Jika persamaan (2.39) disubstitusikan ke dalam fungsi alih sistem:

$$A.y(t) = B.u(t) \quad (2.42)$$

maka fungsi alih lingkaran tertutup dari blok diagram sistem di atas menjadi:

$$\frac{y(t)}{r(t)} = \frac{B.H}{F.A + B.G} \quad (2.43)$$

Oleh karena itu, polinomial ($F.A + B.G$) dibuat sama dengan polinomial T yang tersiri dari kutub–kutub yang diinginkan. Sehingga persamaan penempatan kutubnya adalah:

$$T = F.A + B.G \quad (2.44)$$

Nilai konstanta H diperoleh dari persamaan lingkaran tertutup sistem pada kondisi tunak. Persamaan lingkaran tertutup sistem dinyatakan sebagai berikut:

$$y(t) = \frac{B.H}{T} w(t) \quad (2.45)$$

Pada kondisi tunak, ($t \rightarrow \infty$) maka $y(t) \approx w(t)$ sehingga konstanta H :

$$H = \frac{T}{B} \Big|_{z=1} \quad (2.46)$$

Agar persamaan (2.45) dapat diselesaikan, maka ada beberapa persyaratan yang harus dipenuhi, yaitu:

$$n_f = n_b$$

$$n_g = n_a - 1$$

A dan B tidak memiliki faktor sekutu

$$n_t \leq n_a + n_b$$

Pola umum dari persamaan penempatan kutub adalah:

$$\begin{bmatrix} 1 & 0 & \cdots & 0 & b_1 & 0 & \cdots & 0 \\ a_1 & 1 & \cdots & \vdots & b_2 & b_1 & \cdots & \vdots \\ a_2 & a_1 & \cdots & 0 & \vdots & b_2 & \cdots & b_1 \\ \vdots & a_2 & & 1 & b_{nb} & \vdots & & b_2 \\ a_{na} & \vdots & & a_1 & 0 & b_{nb} & & \vdots \\ \vdots & a_{na} & & a_2 & \vdots & 0 & & b_{nb} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & a_{na} & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{nf} \\ g_0 \\ g_1 \\ \vdots \\ g_{ng} \end{bmatrix} = \begin{bmatrix} t_1 - a_1 \\ t_2 - a_2 \\ \vdots \\ \vdots \\ -a_{na} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (2.47)$$

Koefisien pengendali $[f_1 \ f_2 \ \dots \ f_{nf} \ g_0 \ g_1 \ g_2 \ \dots \ g_{ng}]$ diperoleh dengan menerapkan aturan eliminasi Gauss pada matriks di atas. Koefisien ini kemudian dimasukkan ke persamaan 2.16 menjadi:

$$(1 + f_1 z^{-1} + \dots + f_{nf} z^{-nf}) u(t) = H.w(t) - (g_0 + g_1 z^{-1} + \dots + g_{ng} z^{-ng}) y(t) \quad (2.48)$$

Karena z^{-1} merupakan operator waktu tunda, maka dalam fungsi waktu persamaan di atas menjadi:

$$[u(t) + f_1 u(t-1) + \dots + f_{nf} u(t-nf)] = H.w(t) - [g_0 y(t) + g_1 y(t-1) + \dots + g_{ng} y(t-ng)] \quad (2.49)$$

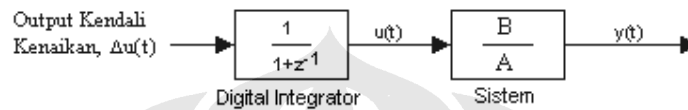
Sehingga sinyal kendali, $u(t)$, yang dikirimkan sebagai masukan sistem adalah:

$$u(t) = H.w(t) - u(t) - f_1 u(t-1) - \dots - f_{nf} u(t-nf) - g_0 y(t) - g_1 y(t-1) - \dots - g_{ng} y(t-ng) \quad (2.50)$$

2.6 ATURAN KENDALI KENAIKAN (INCREMENTAL CONTROL)

Pengendali ini merupakan pengendali yang dapat menghilangkan galat tunak tanpa adanya pre-compensator dan dapat digunakan pada sistem dengan orde sembarang.

Algoritma aturan kendali kenaikan menghasilkan keluaran berupa $u(t) = \Delta u(t) + u(t-1)$. Sinyal kendali $u(t)$ ini akan menjadi masukan sistem. Konsep aksi kendali kenaikan dalam blok diagram adalah sebagai berikut:



Gambar 2.4. Konsep Aksi Kendali Kenaikan

Dengan asumsi bahwa keluaran sistem yang dikendalikan diganggu oleh gangguan yang terukur $v(t)$ dan gangguan yang tidak terukur $s(t)$, maka persamaan model yang digunakan adalah:

$$Ay(t) = Bu(t-1) + Dv(t) + s(t) \quad (2.51)$$

Dalam bentuk kenaikan, persamaan ini dapat diekspresikan menjadi:

$$\bar{A}y(t) = B\Delta u(t-1) + \bar{D}v(t) + \bar{s}(t) \quad (2.52)$$

dengan $\bar{A} = (1 - z^{-1})A = \Delta A$ dan seterusnya.

Diasumsikan pengendali mempunyai bentuk

$$\Delta u(t) = -\frac{G}{F}(y(t) - r(t)) \quad (2.53)$$

dengan G dan F diperoleh dengan memenuhi persamaan identitas

$$FA + z^{-1}BG = T_d \quad (2.54)$$

Derajat polinomial G harus dinaikkan satu untuk memastikan bahwa persamaan (2.53) mempunyai solusi yang unik.

Substitusi persamaan (3.53) dan (3.52) kedalam persamaan persamaan (3.51) menghasilkan respon sistem lingkaran tertutup

$$y(t) = \frac{GB}{T_d} w(t-1) + \frac{F\bar{D}}{T_d} v(t) + \frac{F}{T_d} s(t) \quad (2.55)$$

yang dapat ditulis kembali menjadi

$$y(t) = w(t) + \frac{(1 - z^{-1})}{T_d} \{FDv(t) + FAw(t) + Fs(t)\} \quad (2.56)$$

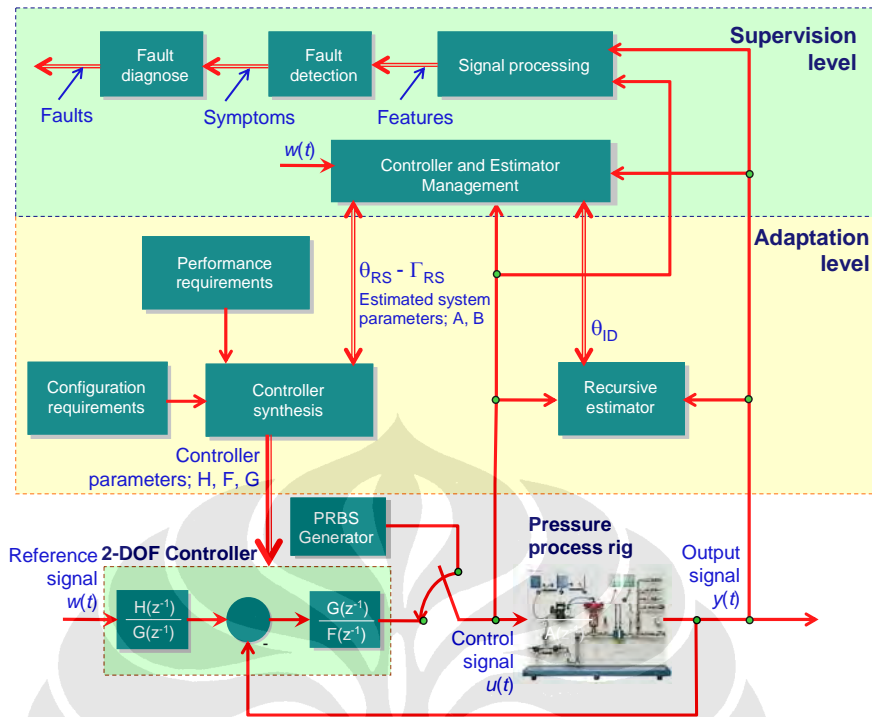
Suku kedua dari bagian ruas kanan dapat direpresentasikan sebagai multi gangguan lingkaran tertutup. Gangguan ini mempunyai kutub-kutub yang terletak pada akar-akar yang ditentukan pada T_d . Adanya suku $(1 - z^{-1})$ menyebabkan pada frekuensi rendah ($\omega \rightarrow 0$ atau $z \rightarrow 1$), pengaruh semua gangguan dapat dihilangkan walaupun parameter sistem berubah.

2.7 LEVEL SUPERVISI

Penetapan mekanisme adaptasi (faktor pelupa < 1) pada pengendali adaptif menghasilkan permasalahan "bursting" estimasi parameter. Hal ini baru dapat dikenali setelah proses berjalan cukup lama.

Untuk mengatasi permasalahan tersebut, diperlukan level tambahan pada skema sistem kendali swa-tala, seperti terlihat pada gambar 2.5. Disamping mempunyai kombinasi algoritma estimasi parameter rekursif dan algoritma disain parameter pengendali, pengendalian swa-tala juga dilengkapi tingkat yang lebih tinggi yang disebut level supervisi.

Level supervisi memantau kerusakan dan kinerja estimasi parameter, kinerja pengendalian, dan kestabilan lup tertutup. Berdasarkan kriteria – kriteria yang ditetapkan, level supervisi akan memutuskan apakah estimator akan melakukan proses estimasi baru jika data yang diperoleh tidak membawa informasi baru. Selain estimator, level supervisi juga memantau kinerja blok disain pengendali.



Gambar 2.5. Sistem kendali adaptif berbasis supervisi

Tujuan manajemen *estimator* adalah untuk memastikan bahwa model sistem teridentifikasi cukup sesuai dengan dinamika kelakuan masukan – keluaran sistem secara *online* atau *real time*. Pada skripsi ini, level supervisi yang dirancang dan diaplikasikan hanya Manajemen *estimator* dan *controller*. Level supervisi bagian ini akan memantau besaran – besaran dari *estimator* dan *controller* berikut ini:

- a. Besaran sinyal:
 - i. a priori error $e(k)$
 - ii. rata – rata kesalahan prediksi $\bar{e}(k)$
 - iii. variansi kesalahan prediksi $\sigma_e^2(k)$
 - iv. fungsi autokorelasi $\phi_{ee}(k)$
 - v. fungsi kroskorelasi $\phi_{ue}(k)$
 - vi. loss function $V(k)$
- b. Besaran parameter
 - i. nilai ekspektasi parameter terestimasi $E\{\theta\}$
 - ii. variansi parameter terestimasi $\sigma_\theta^2(k)$

iii. matriks kovarian $\mathbf{P}(k)$

iv. matrix informasi $\mathbf{H}(k)$

Beberapa kuantiti besaran di atas dikombinasikan untuk memantau kemungkinan kesalahan pada estimator, klasifikasi alasan kesalahan, dan mengambil aksi untuk memperbaiki estimasi parameter.

Fenomena *bursting* yang mungkin terjadi pada estimator diakibatkan sinyal masukan sistem tidak cukup tereksitasi terutama jika *fading memory* estimator dipilih dengan $\lambda < 1$. Dalam kasus ini baris yang tidak linier akan muncul pada matriks informasi \mathbf{H} , sehingga masalah identifikasi tidak dapat dipecahkan.

Disamping memantau prosedur estimasi parameter, level supervisi juga memantau perancangan pengendali. Adanya proses monitoring sebelum perhitungan parameter pengendali akan memperbaiki keamanan lup sistem kendali swa-tala, mengurangi pengaruh perubahan parameter pengendali yang cepat. Pengaruh dari cacat estimator yang mengestimasi zero atau kutub sistem yang mendekati unit lingkaran atau berada di luar lingkaran dapat dihindari dengan adanya level supervisi, sehingga disain pengendali dengan pemotongan zero atau kutub proses tidak dilakukan.

Untuk perubahan parameter proses yang kecil, estimator rekursif dapat mengikuti perubahan parameter proses, jika faktor pelupa ditentukan lebih kecil dari satu. Sehingga parameter pengendali dapat dikalkulasi dan diadaptasi secara kontinu. Akan tetapi jika perubahan estimasi parameter cepat, estimator rekursif tidak dapat mengikuti dengan cepat. Dalam kondisi ini, perhitungan parameter pengendali dihentikan untuk sementara waktu dihentikan, dan kendali *on-line* menggunakan nilai parameter pengendali yang lama. Setelah kondisi tunak mulai tercapai, perhitungan parameter pengendali dapat dimulai kembali.

Untuk perubahan parameter proses yang sangat lambat, kecenderungan estimasi parameter dan nilai rata-ratanya harus diperhitungkan. Jika bergerak naik atau berkurang secara monoton, maka disain pengendali dilakukan kembali secara kontinu. Sebelum mengganti nilai parameter pengendali lama dengan nilai yang baru, analisis kestabilan lup kendali dapat dilakukan berdasarkan model proses terestimasi dan pengendali yang sudah dirancang. Secara teoritis perbaikan kestabilan lup kendali dengan menggunakan nilai parameter pengendali baru

diamati ulang dengan menggunakan simulasi sistem kendali lup tertutup secara *on-line*. Simulasi ini dilakukan berdasarkan model terakhir proses terestimasi, harga awal riil untuk nilai masukan proses saat ini dan sebelumnya, serta nilai keluaran proses sebelumnya. Jika kesalahan antara hasil prediksi dengan keluaran proses sebenarnya kecil untuk memberikan respon sistem yang lebih baik, maka nilai parameter pengendali baru ini dapat digunakan untuk mengendalikan proses riil.



BAB 3

PERANCANGAN SISTEM

3.1 DESKRIPSI SISTEM PRESSURE PROCESS RIG [1]

Pada skripsi ini, algoritma level supervisi diaplikasikan pada sistem *Pressure Process Rig* (Feedback 38-714) yang sudah dikendalikan oleh pengendali adaptif. Alat ini dapat digunakan untuk mendemonstrasikan prinsip-prinsip dari proses pengukuran dan pengendalian.



Gambar 3.1. Pressure Process Rig (38-714, Feedback Instruments Limited).

Alat/sistem ini terdiri dari:

1. Sebuah *Pneumatic Control Valve*,
2. Sebuah *I/P converter*,
3. Sebuah *Blok Orifice*,
4. Sebuah *Flowmeter*,
5. Enam buah pengukur tekanan (*Gauge*),
6. Dua buah regulator tekanan,

7. Sebuah sensor tekanan,
8. Sebuah sensor perbedaan tekanan,
9. Sebuah *Air Receiver*,
10. Tujuh buah *Valve*, dan
11. Beberapa buah saluran pipa yang menghubungkan komponen –komponen di atas.

Sistem ini merupakan *Single Input Single Output* (SISO) dengan masukan berupa *Air Compressor*. Komponen-komponen sistem yang berpengaruh pada pembuatan skripsi ini adalah 2 buah regulator (R1 dan R2), 6 buah indikator tekanan (G1, G2, ..., G6), dan 7 buah *valve* (V1, V2, ..., V7). Regulator R1 berfungsi untuk mengendalikan tekanan yang diukur pada G1. Regulator R2 berfungsi untuk mengendalikan tekanan yang diukur pada G3 atau G4 atau G5.

Keluaran yang dapat dikendalikan pada sistem ini ada 2 buah, yaitu tekanan pada G5 dan perbedaan tekanan pada G4 dan G5. *Pressure transmitter* (Feedback 38-461) berfungsi untuk mengubah keluaran *pressure sensor* menjadi besaran arus standar (4-20mA). *Differential Pressure Transmitter* (Feedback 38-462) berfungsi untuk mengubah keluaran *differential pressure sensor* menjadi besaran arus standar (4-20mA). Keluaran yang dikendalikan dalam skripsi ini adalah tekanan pada G5, sehingga hanya *pressure transmitter* dan *pressure sensor* yang digunakan.

Sebelum digunakan, *pressure transmitter* harus dikalibrasi terlebih dahulu. Keterangan terperinci pengalibrasian komponen ini sapat dilihat pada Laporan Kerja Praktek [3].

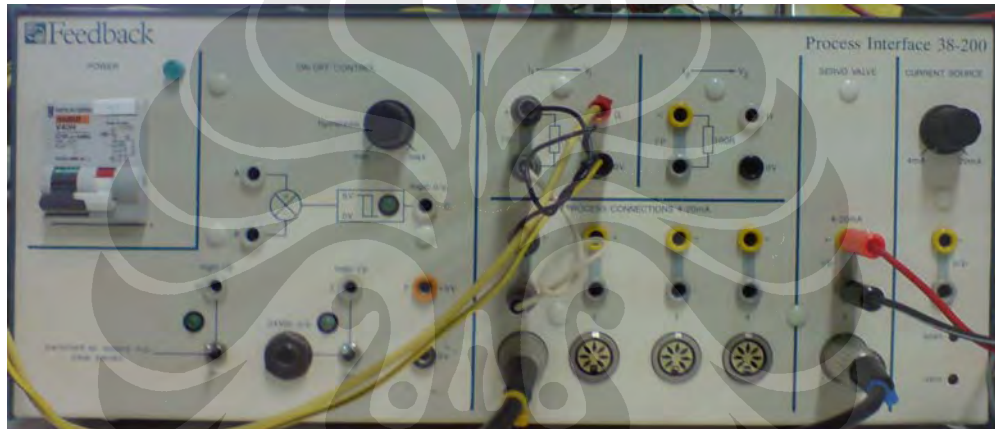
Terdapat sejumlah variasi konfigurasi sistem yang dapat diterapkan pada *Pressure Process Rig*. Variasi ini diperoleh dengan cara membeda-bedakan (membuka atau menutup) katup (*valve*). Untuk keperluan skripsi ini, konfigurasi sistem yang digunakan adalah konfigurasi dengan membuka katup V2, V4, dan V7 (sebagian) dan menutup katup lainnya. Dan keluaran pada G5 (*process variable*) memiliki jangkauan 0–10 psi yang dengan *Process Transmitter* diubah menjadi sinyal arus standar (4-20mA).

3.1.1 Peralatan Pendukung

3.1.1.1 *Process Interface* (Feedback 38-200)

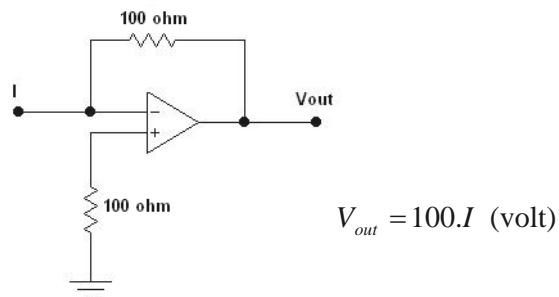
Process Interface digunakan sebagai jembatan penghubung antara *Pressure Process Rig*, yang menggunakan sistem pentransmision sinyal menggunakan 7 kawat, dengan menggunakan perangkat listrik lain yang menggunakan sistem pentransmision dengan 2 kawat (1 pasang) untuk menciptakan kondisi lup tertutup.

Pada *Process Interface* telah tertanam dua *I/V Converter*, yang salah satunya digunakan dalam perancangan perangkat keras untuk skripsi ini. Secara keseluruhan, sistem pengabelan pada *Process Interface* dapat diilustrasikan seperti gambar di bawah ini.



Gambar 3.2.Koneksi pada *Process Interface* (Feedback 38-200).

Pengubah sinyal arus menjadi sinyal tegangan (*I/V Converter*) pada *Process Interface* ini berfungsi untuk mengubah sinyal yang dikenal oleh sistem (sinyal arus) ke sinyal yang dikenal oleh komputer (sinyal tegangan). Pada *Process Interface* ini, besarnya R pada *I/V Converter* adalah 100 ohm. Secara sederhana, rangkaian *I/V Converter* dapat dilihat pada gambar berikut:



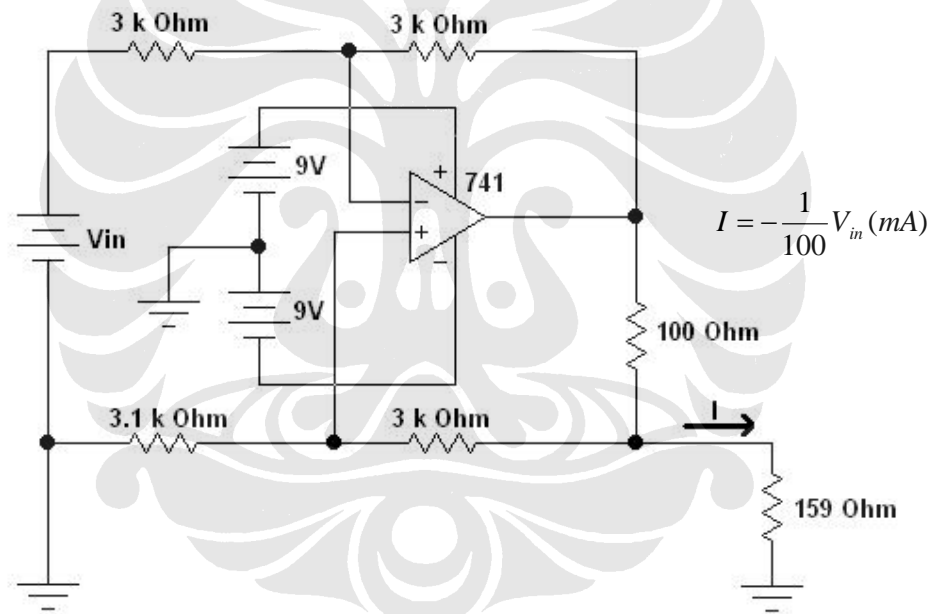
Gambar 3.3.*Current-to-voltage converter*.

Jangkauan sinyal arus yang ditransmisikan oleh *Pressure Transmitter* adalah 4-20 mA. Dengan resistor sebesar 100 ohm pada *I/V Converter*, maka sinyal tegangan yang dihasilkan mempunyai jangkauan antara 0,4-2,0 volt.

3.1.1.2 *V/I Converter*

Dalam pengiriman sinyal kendali untuk *Pressure Process Rig* yang berasal dari komputer, dibutuhkan suatu rangkaian pengubah sinyal tegangan menjadi sinyal arus. Dalam hal ini, *V/I Converter* digunakan untuk mengubah sinyal tegangan 0,4-2,0 volt menjadi sinyal arus 4-20 mA (standar pensinyalan yang digunakan oleh *Pressure Process Rig*).

Konfigurasi rangkaian *V/I Converter* agar memperoleh spesifikasi di atas adalah sebagai berikut:



Gambar 3.4. *Voltage-to-Current Converter*.

Walaupun amplitudo keluaran *V/I Converter* (sinyal arus listrik yang dihasilkan) sesuai dengan yang diinginkan, akan tetapi arah aliran arus yang dihasilkannya terbalik (berharga negatif). Sehingga pada koneksinya dengan *Process Interface*, pengabelan harus dibuat terbalik (Kutub positif keluaran *V/I Converter* terhubung dengan *Ground* pada *Process Interface* dan *Ground* keluaran *V/I Converter* terhubung dengan Kutub positif pada *Process Interface*).

3.1.1.3 *Air Compressor*

Air Compressor yang digunakan pada skripsi ini adalah ***Oil Free Compressor (P50/10L Werther International Inc.)***. Tekanan yang dialirkan oleh *Air Compressor* sebagai masukan keseluruhan *Pressure Process Rig* sebesar 30 psi (batas maksimum tekanan masukan 40 psi).¹

3.1.1.4 *Data Acquisition*

Untuk keperluan *data acquisition*, digunakan card *National Instruments PCI-6024E*. Pada card ini terdapat 2 buah DAC (*Digital-to-Analog Converter*) dan 8 buah ADC (*Analog-to-Digital Converter*). Pada skripsi ini, hanya 1 buah DAC (*channel 0*) dan 1 buah ADC (*channel 0*) yang digunakan.

Untuk keperluan ADC skripsi ini, bagian positif (+) pentransmision sinyal keluaran proses terhubung dengan pin 68 (ACH0) dan *ground* (-) pentransmision terhubung dengan pin 34 (ACH8). *Ground* untuk *channel 0*, yaitu pin 67 (AIGND) dihubungkan langsung dengan bagian negatif *channel 0* (pin34).

Untuk keperluan DAC skripsi ini, bagian positif (+) pentransmision sinyal kendali terhubung dengan pin 22 (DAC0 OUT) dan *ground* (-) pentransmision terhubung dengan pin 55 (AOGND). Penghubungan kabel tambahan yaitu antara pin 67 dan pin 55 ditujukan agar *ground* masukan dan keluaran PCI-6024E memiliki referensi yang sama.

3.1.1.5 *Personal Computer*

Untuk keperluan skripsi ini, komputer yang digunakan untuk melakukan perhitungan sinyal kendali memiliki spesifikasi:

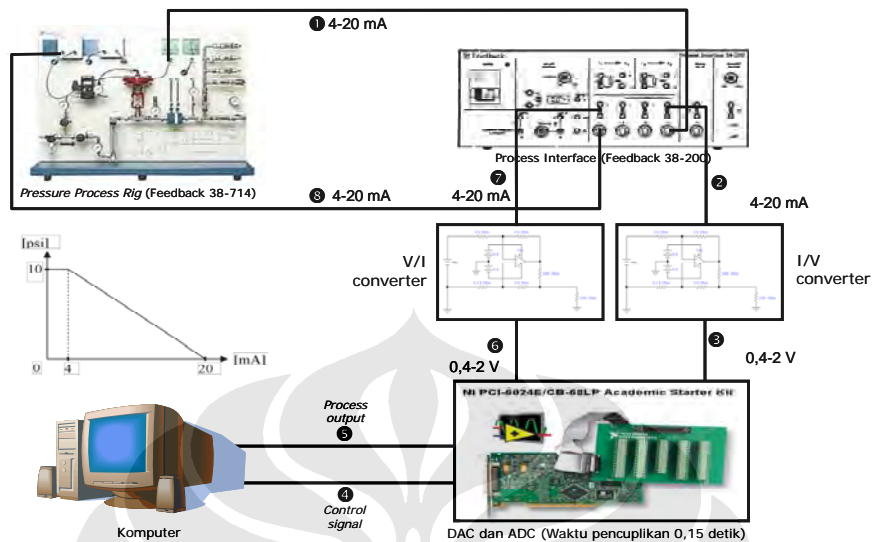
- a. Processor 1,7 GHz
- b. RAM 512 MB

3.1.2 *Interkoneksi Alat*

Interkoneksi keseluruhan perangkat keras, mulai dari keluaran sinyal dari sistem ke komputer dan keluaran sinyal kendali dari komputer ke sistem dalam

¹ Buku petunjuk penggunaan *Pressure Process Rig* (Feedback 38-714)

bentuk pengendalian aktuatur (*Pneumatic Control Valve*) dapat dilihat pada gambar 3.5.



Gambar 3.5. Interkoneksi komponen-komponen perangkat keras.

3.2 SPESIFIKASI PERANGKAT LUNAK YANG DIGUNAKAN

Pada skripsi ini, spesifikasi perangkat lunak yang digunakan adalah:

1. Sistem Operasi Microsoft® Windows® 2000 Professional
2. Perangkat Lunak Perhitungan Teknikal MATLAB® 6.5
3. Dynamic System Simulation Simulink® 5.0

3.3 SISTEM KENDALI SWATALA PADA PRESSURE PROCESS RIG

Pada [1], praidentifikasi model merupakan kesatuan sistem kendali swatala dengan kondisi open loop selama 128 pencuplikan (19.2 detik) pertama. Sinyal yang digunakan dalam praidentifikasi ini adalah sinyal PRBS (*Pseudo Random Binary Signal*). Batas sinyal PRBS berada dalam jangkauan daerah kerja linier *Pressure Process Rig*, yaitu 0.8 Volt sampai 1,4 Volt. *Lambda* yang digunakan dalam skripsi ini adalah 0.995. Penentuan nilai ini dikarenakan *bursting* terjadi pada nilai *lambda* ini. Dan waktu cuplik sistem adalah 0.15 detik.

Penentuan parameter pengendali *Pressure Process Rig* dihitung dengan menggunakan Metode Penempatan Kutub dengan Aturan Kendali Kenaikan, yaitu terdapat unsur integrator (pole pada $z = +1.0$) pada Metode Penempatan Kutub. Aturan kendali kenaikan ini menyebabkan hilangnya galat tunak. Kutub lingkaran

tertutup yang diinginkan adalah polinom orde 2 dengan letak kedua kutub pada $z = +0.9$ atau dapat ditulis:

$$T(z^{-1}) = 1 - 1.8z^{-1} + 0.81z^{-2} \quad (3.1)$$

3.4 DISAIN FUNGSI SUPERVISI ESTIMATOR DAN PENGENDALI

3.4.1 Pemantauan Besaran-Besaran dalam Pengendali Adaptif

3.4.1.1 Pemantauan Eksitasi Sinyal Masukan Sistem (Sinyal Kendali)

Fungsi pemantauan ini adalah untuk mencegah letupan nilai estimasi parameter yang tidak terduga. Hal ini dilakukan dengan memastikan bahwa data yang diterima estimator cukup membawa informasi.

Pemantauan eksitasi sinyal masukan proses dilakukan dengan mengautokorelasi sinyal masukan $u(k)$ yang kemudian membentuk matriks \mathbf{H}_{22} . Matriks ini merupakan bagian dari matriks informasi \mathbf{H} , yang didefinisikan sebagai:

$$\mathbf{H}_{22} = \begin{bmatrix} \sum_{k=n}^{n+N-1} u^2(k) & \sum_{k=n}^{n+N-1} u(k)u(k-1) & \dots & \sum_{k=n}^{n+N-1} u(k)u(k-n+1) \\ \sum_{k=n}^{n+N-1} u(k)u(k+1) & \sum_{k=n}^{n+N-1} u^2(k) & \dots & \sum_{k=n}^{n+N-1} u(k)u(k-n+2) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{k=n}^{n+N-1} u(k)u(k+n-1) & \sum_{k=n}^{n+N-1} u(k)u(k+n-2) & \dots & \sum_{k=n}^{n+N-1} u^2(k) \end{bmatrix}$$

$$= \begin{bmatrix} r_{uu}(0) & r_{uu}(1) & \dots & r_{uu}(n-1) \\ r_{uu}(1) & r_{uu}(0) & \dots & r_{uu}(n-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_{uu}(n-1) & r_{uu}(n-2) & \dots & r_{uu}(0) \end{bmatrix} \quad (3.2)$$

Jika sinyal kendali yang diberikan bernilai konstan untuk jangka waktu yang lama, maka informasi yang diberikan kepada estimator mempunyai kualitas data yang rendah atau tidak ada informasi baru yang dapat diberikan. Kondisi ini sangat tidak baik untuk estimator dan pada kondisi ini nilai autokorelasi dari sinyal masukan sangat tinggi. Akibatnya pada matriks \mathbf{H}_{22} terdapat kemungkinan muncul baris yang tidak bebas linier yang mengakibatkan determinan matriks \mathbf{H}_{22}

ini nol atau mendekati nol. Pada saat ini, proses estimasi parameter model proses tidak akan dieksekusi.

3.4.1.2 Pemantauan Parameter Model Terestimasi

Pemantauan ini berfungsi untuk mencegah penghitungan parameter pengendali ketika kinerja parameter pengendali sebelumnya menunjukkan hasil yang lebih baik. Perhitungan parameter pengendali baru akan dilakukan pada kondisi sebaliknya. Pada skripsi ini, parameter pengendali tidak dihitung pada saat transien perubahan sistem dan pada saat sistem tidak stabil

Indikator yang dapat digunakan adalah nilai variansi parameter sistem $\sigma_o^2(k)$. Jika nilai *norm* variansi sistem berada di bawah nilai toleransi yang ditetapkan ε selama beberapa pencuplikan, maka perhitungan parameter pengendali dilakukan karena hal ini menunjukkan perubahan sistem mulai mencapai kondisi tunak. Perhitungan letak zero (untuk jenis pengendali tertentu) dan kutub model sistem juga akan menentukan pengambilan keputusan tersebut. Perhitungan parameter pengendali baru akan dilakukan jika letak semua kutub dan zero sistem berada di dalam *unit circle* yang mewakili kutub yang stabil pada bidang z.

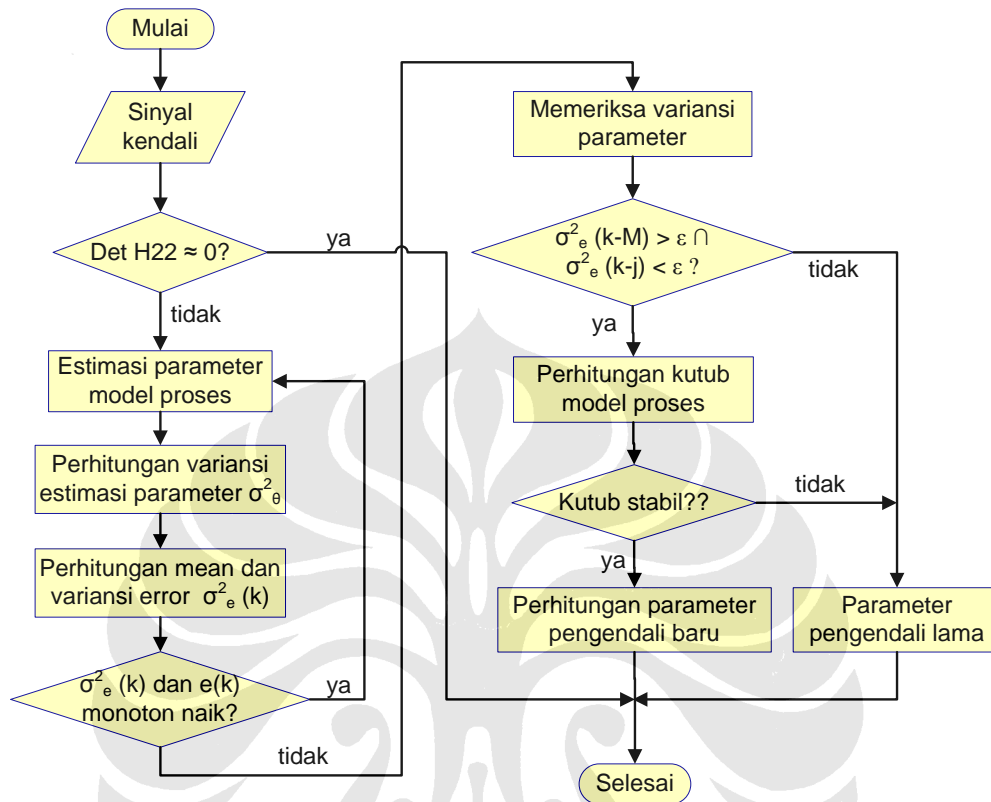
3.4.1.3 Pemantauan sinyal kesalahan prediksi (*prediction error*)

Estimator menghitung parameter-parameter model berdasarkan struktur model yang sudah ditentukan sebelumnya. Besaran yang dievaluasi adalah rata-rata kesalahan prediksi $\bar{e}(k)$ dan variansi kesalahan prediksi $\sigma_e^2(k)$. Jika kedua besaran ini monoton naik, maka fungsi manajemen estimator akan merestrukturisasi model sistem. Tetapi dalam skripsi ini, tindakan yang diambil adalah menghentikan keseluruhan proses pengendalian.

3.4.2 Disain Logika Supervisi

Berdasarkan teori yang ada, besaran – besaran yang diperlukan harus dihitung secara on-line. Dari hasil pemantauan besaran-besaran tersebut, maka keputusan juga dilakukan setiap waktu pencuplikan. Dalam setiap pencuplikan,

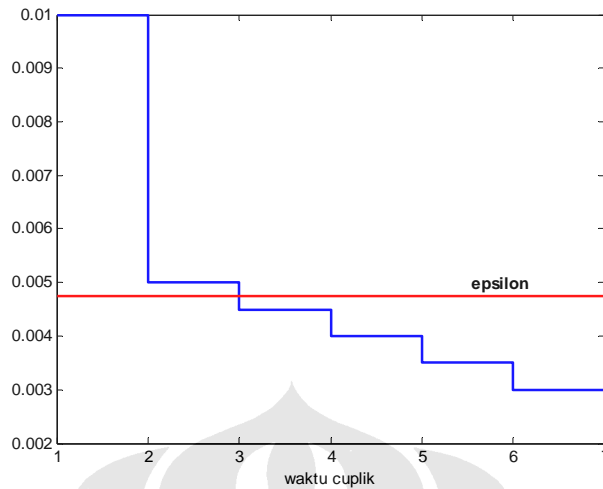
perhitungan besaran – besaran yang diperlukan dan keputusan yang akan diambil dapat diwakilkan dengan *flow chart* berikut:



Gambar 3.6. Diagram alir level supervisi.

Pada *flow chart* terlihat bahwa terdapat 4 buah pengambilan keputusan setiap waktu pencuplikan, yaitu:

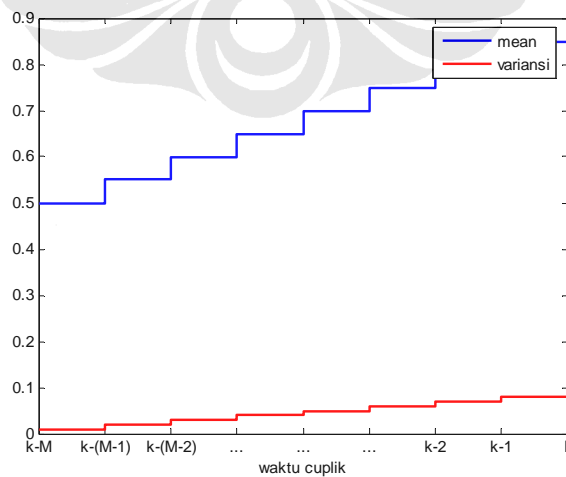
1. Keputusan pertama berfungsi untuk memutuskan apakah perlu melakukan perhitungan estimasi parameter sistem. Keputusan ini diambil berdasarkan pantauan besaran determinan matriks H_{22} . Matriks ini menunjukkan hubungan sinyal kendali pada waktu cuplik k dengan sinyal – sinyal kendali pada waktu cuplik sebelumnya. Nilai komponen – komponen matriks ini akan saling mendekati seiring dengan sinyal kendali yang cenderung konstan sehingga nilai determinan matriks ini mendekati nol. Hal ini menjadi indikasi bahwa estimasi model dan penalaan parameter pengendali sudah bekerja dengan baik. Dalam penerapannya, jika nilai determinan matriks ini berada di bawah suatu nilai ϵ , maka proses identifikasi dan penalaan parameter pengendali tidak perlu dilakukan. Dimana nilai ϵ ini merupakan suatu nilai yang mendekati nol



Gambar 3.7. Pantauan determinan matriks H_{22} .

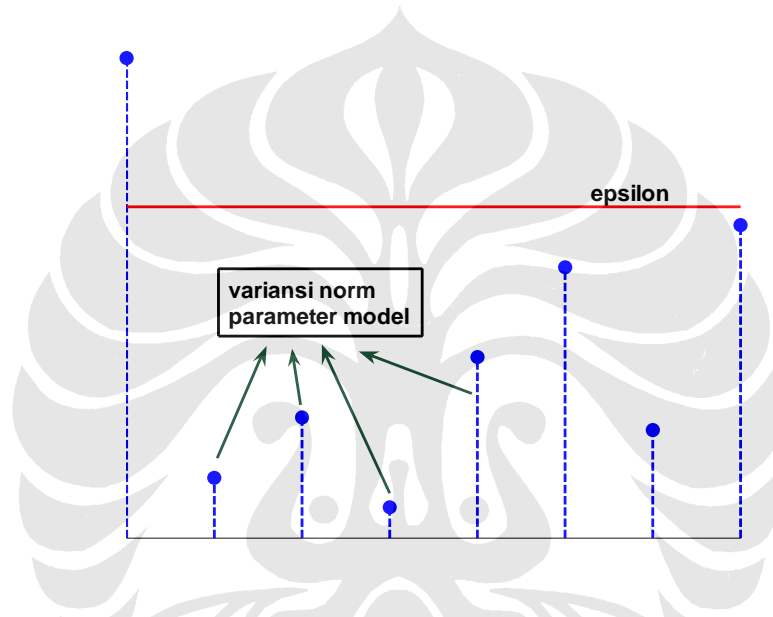
Pada gambar 3.7, perhitungan estimasi parameter sistem dihentikan pada waktu cuplik ke-3.

- Keputusan kedua diambil berdasarkan pantauan besaran *mean* dan variansi *prediction error* dan berfungsi untuk memutuskan apakah model yang dipakai dapat mewakili sistem yang dikendalikan. Jika *mean* dan variansi besaran *prediction error* terus naik (sebanyak M pencuplikan pada gambar 3.8), maka pendekatan model yang digunakan tidak sesuai dan harus diganti dengan jenis model lainnya. Pada skripsi ini, model yang ada tidak diganti. Hanya saja proses identifikasi dan pengendalian tidak dilanjutkan untuk mencegah kejadian yang tidak diinginkan



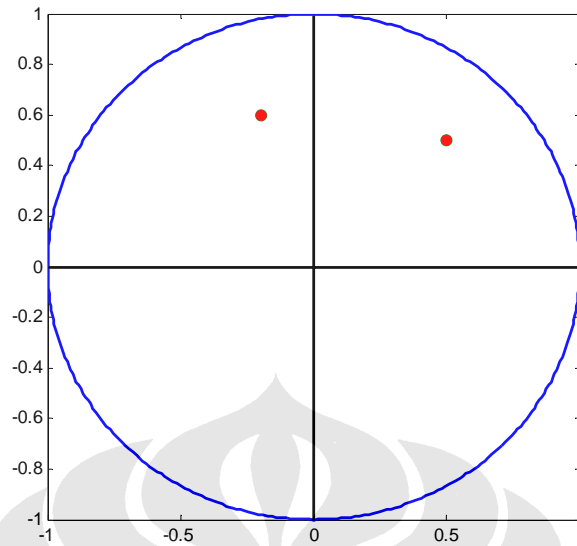
Gambar 3.8. Pantauan rata-rata dan variansi *prediction error*.

- Keputusan ketiga berfungsi untuk memutuskan apakah model sistem sudah cukup baik untuk dapat melakukan perhitungan parameter pengendali. Keputusan ini diambil berdasarkan pantauan besaran *norm* parameter model sistem. Jika *norm* parameter sistem sudah berada di bawah suatu nilai ϵ selama beberapa waktu pencuplikan, maka model sistem sudah dapat dianggap mewakili sistem sebenarnya. Pada saat ini, perhitungan parameter pengendali dapat dilakukan dan diperkirakan dapat bekerja dengan baik. Penggambaran pengambilan keputusan ini secara *visual* adalah:



Gambar 3.9. Variansi norm parameter yang sesuai untuk penalaan pengendali.

- Dan keputusan terakhir diambil berdasarkan stabil atau tidaknya kutub sistem. Ini merupakan syarat terakhir yang diperlukan untuk memutuskan perlu tidaknya perhitungan parameter pengendali dilakukan. Penalaan pengendali pada sistem yang tidak stabil akan menyebabkan buruknya kinerja pengendali. Pada bidang z , letak kutub yang stabil adalah yang berada di dalam lingkaran berjari-jari 1 seperti gambar 3.10.



Gambar 3.10.Letak kutub stabil dalam bidang z.

3.4.3 Algoritma Rekursif Fungsi Supervisi

Besaran – besaran yang dibutuhkan dalam level supervisi perlu dihitung secara *on-line* (setiap waktu pencuplikan). Berikut ini adalah penyesuaian rumus–rumus untuk menghitung besaran – besaran tersebut menjadi rumus *on-line* atau bersifat rekursif.

3.4.3.1 Rata – rata Rekursif (*mean*)

Perhitungan perubahan *mean* pada pencuplikan sekarang berdasarkan *mean* pencuplikan sebelumnya menggunakan rumus:

$$\bar{x}_{(k)} = \frac{(k-1)\bar{x}_{(k-1)} + x(k)}{k} \quad (3.3)$$

dengan:

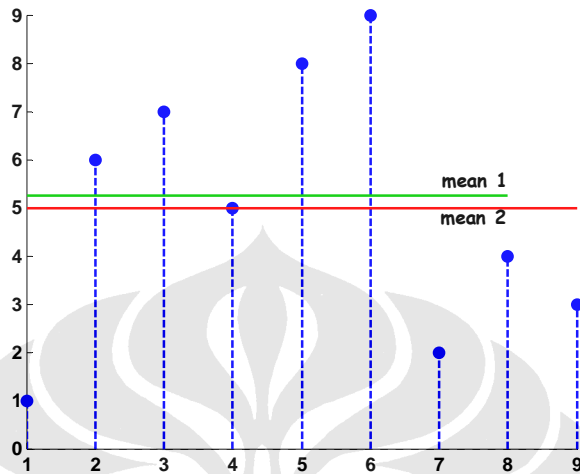
$\bar{x}_{(k)}$ = *mean* pada pencuplikan ke-k

$\bar{x}_{(k-1)}$ = *mean* pada pencuplikan ke-(k-1)

$x(k)$ = data pada pencuplikan ke-k

3.4.3.2 Variansi Rekursif

Perhitungan variansi rekursif didasarkan pada perubahan variansi jika terjadi perubahan *mean* seperti pada gambar 3.11:



Gambar 3.11. Perubahan variansi karena perubahan rata-rata.

Misal variansi pada saan *mean* awal sama dengan \bar{x} adalah

$$\sigma_1^2 = \frac{\sum_{i=1}^n (x_i - \bar{x}_1)^2}{(n-1)} \quad (3.4)$$

- dengan σ_1^2 : variansi
- x_i : data ke- i
- \bar{x}_1 : nilai rata-rata
- n : banyaknya data

Jika nilai *mean* berubah menjadi $\bar{x}_2 = \bar{x}_1 + b$, maka besarnya variansi menjadi

$$(n-1)\sigma_2^2 = \sum_{i=1}^n (x_i - \bar{x}_1 - b)^2 \quad (3.5)$$

$$(n-1)\sigma_2^2 = \sum_{i=1}^n (x_i - \bar{x}_1)^2 - 2b \sum_{i=1}^n (x_i - \bar{x}_1) + \sum_{i=1}^n b^2 \quad (3.6)$$

$$(n-1)\sigma_2^2 = \sum_{i=1}^n (x_i - \bar{x}_1)^2 - 2b \left(\sum_{i=1}^n x_i - \sum_{i=1}^n \bar{x}_1 \right) + \sum_{i=1}^n b^2 \quad (3.7)$$

Pada persamaan 3.7, suku ke-2 dan suku ke-3 mempunyai nilai yang sama. Sehingga persamaan variansi menjadi

$$(n-1)\sigma_2^2 = \sum_{i=1}^n (x_i - \bar{x}_1)^2 + \sum_{i=1}^n b^2 \quad (3.8)$$

$$(n-1)\sigma_2^2 = (n-1)\sigma_1^2 + \sum_{i=1}^n b^2$$

$$\sum_{i=1}^n (x_i - \bar{x}_2) = \sum_{i=1}^n (x_i - \bar{x}_1) + \sum_{i=1}^n b^2 \quad (3.9)$$

Dari persamaan 3.9 terlihat bahwa perubahan variansi hanya tergantung dari *sigma* kuadrat perubahan *mean* (b^2). Karena itu, baik perubahan positif maupun perubahan negatif dari *mean* akan menghasilkan perubahan variansi yang sama.

Pada perhitungan variansi rekursif, data yang baru (x_k) akan mengubah nilai $\bar{x}_{(k-1)}$ menjadi $\bar{x}_{(k)}$. Nilai variansi pada pencuplikan ke-(k-1) adalah

$$\sigma_{(k-1)}^2 = \frac{\sum_{i=1}^{k-1} (x_i - \bar{x}_{(k-1)})^2}{(k-2)}$$

atau

$$(k-2)\sigma_{(k-1)}^2 = \sum_{i=1}^{k-1} (x_i - \bar{x}_{(k-1)})^2$$

Dengan $b = \bar{x}_{(k)} - \bar{x}_{(k-1)}$, maka variansi pada pencuplikan ke-(k-1)

$$\sigma_k^2 = \frac{\sum_{i=1}^k (x_i - \bar{x}_k)^2}{(k-1)}$$

$$(k-1)\sigma_k^2 = \sum_{i=1}^k (x_i - \bar{x}_{(k)})^2$$

$$(k-1)\sigma_k^2 = \sum_{i=1}^{k-1} (x_i - \bar{x}_{(k)})^2 + (x_k - \bar{x}_{(k)})^2$$

$$(k-1)\sigma_k^2 = \sum_{i=1}^{k-1} (x_i - \bar{x}_{(k-1)})^2 + \sum_{i=1}^{k-1} b^2 + (x_k - \bar{x}_{(k)})^2$$

$$(k-1)\sigma_k^2 = (k-2)\sigma_{(k-1)}^2 + (k-1)b^2 + (x_k - \bar{x}_{(k)})^2$$

$$\sigma_{(k)}^2 = \frac{(k-2)}{(k-1)}\sigma_{(k-1)}^2 + b^2 + \frac{(x_k - \bar{x}_{(k)})^2}{(k-1)} \quad (3.10)$$

3.4.3.3 Matrix informasi H

Matrix H adalah matrix yang berisi autokorelasi sinyal kendali.

$$\mathbf{H} = \begin{bmatrix} \sum_{k=n}^{n+N-1} u^2(k) & \sum_{k=n}^{n+N-1} u(k)u(k-1) & \dots & \sum_{k=n}^{n+N-1} u(k)u(k-n+1) \\ \sum_{k=n}^{n+N-1} u(k)u(k+1) & \sum_{k=n}^{n+N-1} u^2(k) & \dots & \sum_{k=n}^{n+N-1} u(k)u(k-n+2) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{k=n}^{n+N-1} u(k)u(k+n-1) & \sum_{k=n}^{n+N-1} u(k)u(k+n-2) & \dots & \sum_{k=n}^{n+N-1} u^2(k) \end{bmatrix} \\
= \begin{bmatrix} r_{uu}(0) & r_{uu}(1) & \dots & r_{uu}(n-1) \\ r_{uu}(1) & r_{uu}(0) & \dots & r_{uu}(n-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_{uu}(n-1) & r_{uu}(n-2) & \dots & r_{uu}(0) \end{bmatrix} \quad (3.11)$$

dengan $u(k)$: sinyal kendali pada waktu cuplik ke-k

$u(k-1)$: sinyal kendali pada waktu cuplik ke-(k-1)

$r_{uu}(0)$: nilai autokorelasi dengan $j = 0$

$r_{uu}(1)$: nilai autokorelasi dengan $j = 1$

Karena sistem *Pressure Process Rig* diestimasi dengan model orde 2, maka ukuran matriks \mathbf{H} adalah $H_{2 \times 2}$. Sehingga nilai autokorelasi yang digunakan hanya $r_{uu}(0)$ dan $r_{uu}(1)$. Matriks \mathbf{H} yang digunakan pada skripsi ini adalah sebagai berikut:

$$\mathbf{H}_{2 \times 2} = \begin{bmatrix} r_{uu}(0) & r_{uu}(1) \\ r_{uu}(1) & r_{uu}(0) \end{bmatrix} \quad (3.12)$$

3.4.3.4 Perhitungan kutub model

Kutub model dihitung dengan rumus

$$x_1, x_2 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2} \quad (3.13)$$

dengan a, b, c merupakan koefisien – koefisien persamaan polinomial orde 2 dan x_1, x_2 merupakan letak kutub – kutub model. Letak kutub model menentukan kestabilan sistem. Pada sistem diskrit, sistem stabil jika berada di dalam lingkaran berjari – jari 1 pada bidang z .

Penentuan kestabilan ini terbagi menjadi 2, yaitu:

- a. Nilai kutub – kutub real

Kutub – kutub yang real diketahui dari nilai D ($\sqrt{b^2 - 4ac}$) yang lebih besar atau sama dengan 0. Pada kutub – kutub real, masing – masing nilai mutlak letak kutub diuji apakah lebih besar atau lebih kecil dari 1.

b. Nilai kutub – kutub kompleks

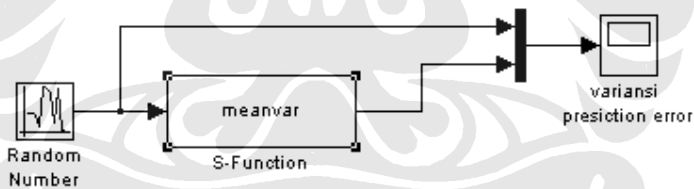
Kutub – kutub ini diketahui jika nilai D lebih kecil dari 0. Pada jenis kutub ini, pengujian kestabilan dilakukan pada nilai *norm* dari letak kutub tersebut.

3.4.4 Validasi Algoritma Rekursif

Validasi diperlukan untuk membuktikan kebenaran algoritma yang dibuat. Validasi ini dilakukan pada semua perhitungan rekursif yang telah dijelaskan pada subbab 3.3.3 Hasil dari algoritma rekursif yang dibuat dibandingkan dengan hasil perhitungan dengan menggunakan fungsi – fungsi MATLAB secara *off-line*.

3.4.4.1 Validasi algoritma rata-rata dan variansi rekursif

Setelah algoritma rata-rata dan variansi rekursif dibuat, maka algoritma tersebut diaplikasikan dalam blok-blok Simulink (Gambar 3.12) untuk mendapatkan rata-rata dan variansi rekursif suatu sinyal acak.



Gambar 3.12. Diagram blok Simulink untuk validasi mean dan variansi rekursif.

Selanjutnya, rata-rata dan variansi dari sinyal acak ini dihitung secara *off-line* dengan menggunakan fungsi MATLAB. Perbandingan 10 hasil rata-rata dan variansi dengan kedua metode tersebut dapat dilihat pada tabel berikut:

Tabel 3. 1 Perbandingan mean dan variansi rekursif dengan MATLAB

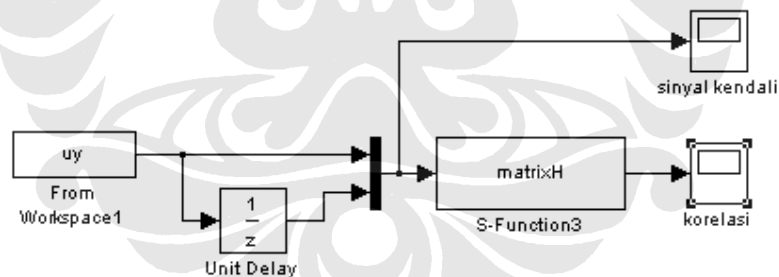
Data	Algoritma Rekursif		Algoritma Offline	
	\bar{x}_i	σ_x^2	\bar{y}_i	σ_y^2
1.165	1.165	0	1.165	0
0.62684	0.8959	0.14478	0.8959	0.1448

0.07508	0.62229	0.29697	0.6223	0.2970
0.35161	0.55462	0.2163	0.5546	0.2163
-0.69651	0.30439	0.47529	0.3044	0.4753
0.05906	0.53635	0.70306	0.5364	0.7031
1.7971	0.46817	0.61843	0.4682	0.6184
0.26407	0.63428	0.75083	0.6343	0.7508
0.87167	0.59315	0.6722	0.5932	0.6722
-1.4462	0.621	0.60527	0.6210	0.6053

Dari kedua cara pencarian nilai mean dan variansi rekursif tersebut, terlihat bahwa perbedaan yang ada hanya karena adanya pembulatan. Hal ini menunjukkan bahwa algoritma rekursif perhitungan *mean* dan variansi rekursif sudah benar.

3.4.4.2 Validasi algoritma determinan matriks $H_{2 \times 2}$ rekursif

Untuk validasi algoritma determinan matriks H_{22} , data sinyal kendali dari *workspace* MATLAB menjadi masukan blok Simulink dengan algoritma determinan matriks H_{22} . Kemudian hasil ini dibandingkan dengan perhitungan determinan matriks H_{22} secara *off-line* dengan fungsi MATLAB.



Gambar 3.13. Diagram blok Simulink untuk validasi determinan $H_{2 \times 2}$.

Secara *off-line*, untuk mendapatkan determinan matriks H_{22} dari data sinyal kendali ini diperlukan beberapa tahap, yaitu:

- Menghitung autokorelasi suatu vektor sinyal kendali tersebut dengan normalisasi
- Membuat matriks H_{22}
- Menggunakan fungsi MATLAB untuk mencari nilai determinan matriks H_{22} .

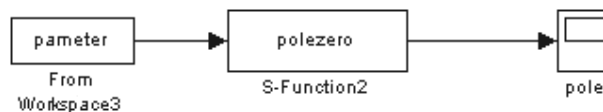
Tabel 3. 2 Perbandingan pehitungan determinan matriks H_{22} rekursif dengan MATLAB

Data		Determinan matriks H_{22}	
$u(k)$	$u(k-1)$	Algoritma Rekursif	Algoritma Offline
1	0	1	1
1	1	0.75	0.7500
1	1	0.55556	0.5556
1	1	0.4375	0.4375
1	1	0.36	0.3600
1	1	0.30556	0.3056
1	1	0.26531	0.2653
1	1	0.23438	0.2344
1	1	0.20988	0.2099
1	1	0.19	0.1900

Perbandingan 10 hasil perhitungan determinan matriks H_{22} dengan kedua metode di atas dapat dilihat pada tabel 3.2. Dari kedua metode mencari determinan $H_{2 \times 2}$ rekursif, terlihat bahwa nilai yang dihasilkan sama. Hal ini menunjukkan bahwa algoritma rekursif untuk menghitung determinan matriks H_{22} sudah benar.

3.4.4.3 Validasi algoritma perhitungan kutub model

Algoritma rekursif perhitungan kutub disimulasikan dengan Simulink seperti gambar 3.14. Masukan blok *polezero* rekursif ini adalah parameter model dan keluarannya adalah jenis kutub (kompleks atau *real*), letak kutub, dan kestabilan.



Gambar 3.14. Diagram blok Simulink untuk validasi perhitungan pole.

Hasil perhitungan kutub rekursif tersebut dibandingkan dengan perhitungan kutub dengan fungsi MATLAB. Perbandingan 10 hasil perhitungan letak kutub sistem dengan kedua metode tersebut terlihat pada tabel 3.3 berikut.

Tabel 3. 3 Perbandingan perhitungan letak pole rekursif dengan MATLAB

Data		Algoritma Rekursif		Algoritma Offline		Keterangan
a_1	a_2	Kutub		Kutub		
0	0	0	0	0	0	Stabil
0	0	0	0	0	0	Stabil
0	0	0	0	0	0	Stabil
0	0	0	0	0	0	Stabil
-0.0069698	0	0.0069698	0	0.0070	0	Stabil
-0.66875	-0.1664	0.86182	-0.19308	0.8618	-0.1931	Stabil
-0.75959	-0.30308	1.0486	-0.28903	1.0486	-0.2890	Tidak stabil
-0.75735	-0.24294	1.0002	-0.24288	1.0002	-0.2429	Tidak stabil
-0.77674	-0.17063	0.95534	-0.1786	0.9553	-0.1786	Stabil
-0.80226	-0.11319	0.92468	-0.12242	0.9247	-0.1224	Stabil

Pada tabel 3.3, terlihat bahwa perbedaan letak kutub metode rekursif dan fungsi MATLAB hanya karena pembulatan saja. Hal ini menunjukkan bahwa algoritma rekursif yang dibuat sudah benar.

BAB 4

PEMBAHASAN DAN ANALISA

4.1 SIMULASI PENERAPAN LEVEL SUPERVISI UNTUK MENGATASI FENOMENA BURSTING

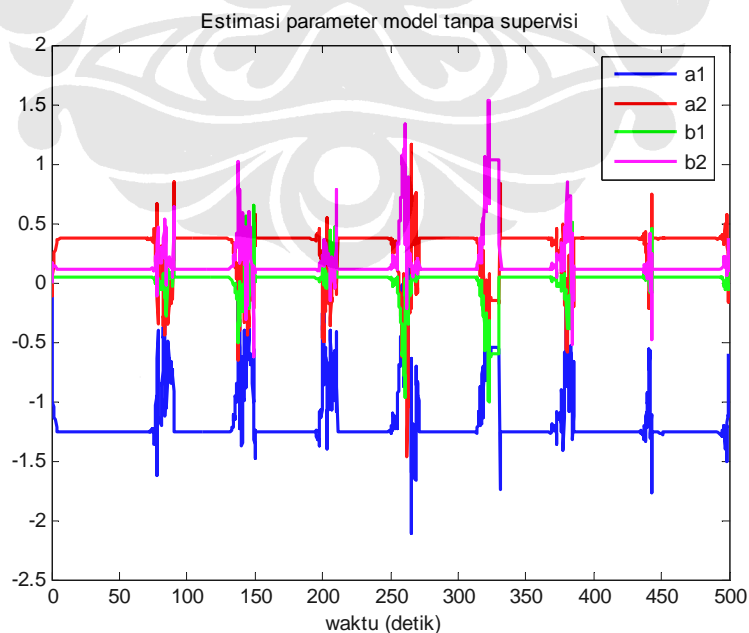
Sebelum mengaplikasikan algoritma level supervisi pada *Pressure Process Rig*, algoritma ini terlebih dahulu disimulasikan dengan Simulink terhadap suatu model yang mewakili *Pressure Process Rig* [1], yaitu:

$$\frac{Y(z)}{U(z)} = \frac{0.042277z^{-1} + 0.11468z^{-2}}{1 - 1.2616z^{-1} + 0.36937z^{-2}}$$

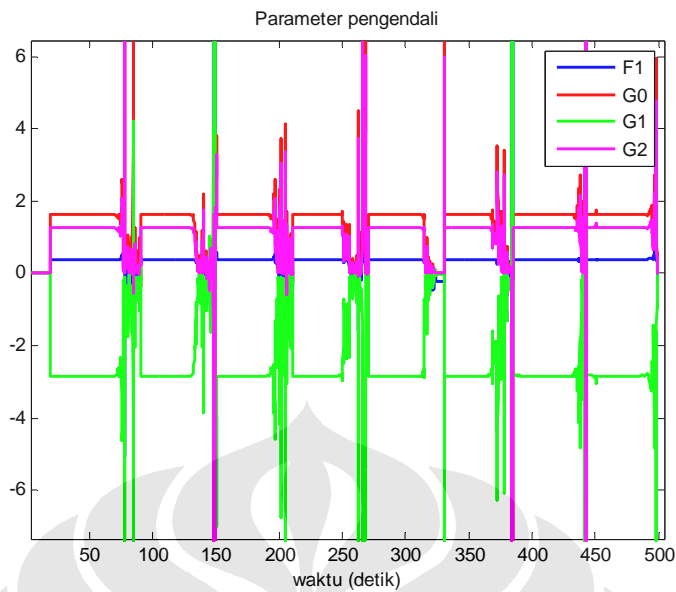
Simulasi yang dilakukan pada model terbagi menjadi dua bagian, yaitu simulasi pada *set-point* tereksitasi dan *set-point* kurang tereksitasi. Dan pada setiap bagian terdapat simulasi tanpa supervisi dan dengan supervisi.

4.1.5 Simulasi penerapan level supervisi pada *set-point* tereksitasi

Pertama, simulasi dilakukan dengan *set-point* yang tidak konstan (tereksitasi). Dengan nilai *lambda* 0.8, nilai estimasi parameter model dan penalaan parameter pengendali dapat dilihat pada gambar 4.1 and 4.2.

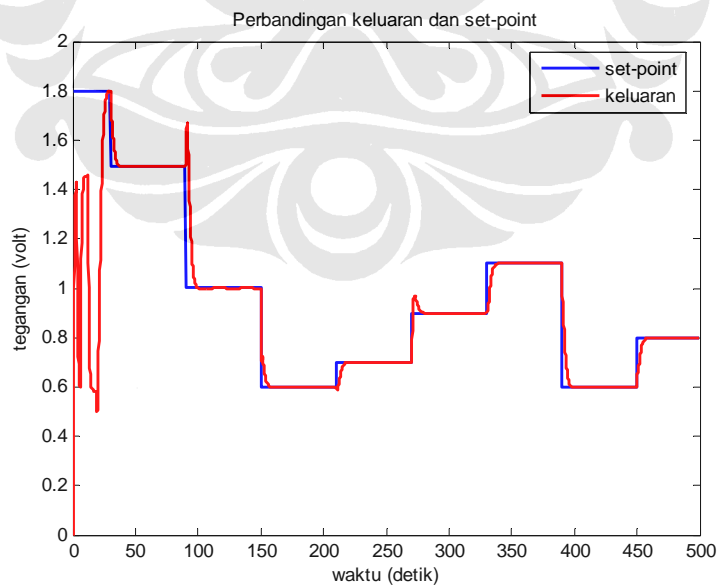


Gambar 4.1. Simulasi estimasi parameter model tanpa supervisi pada *set-point* tereksitasi.



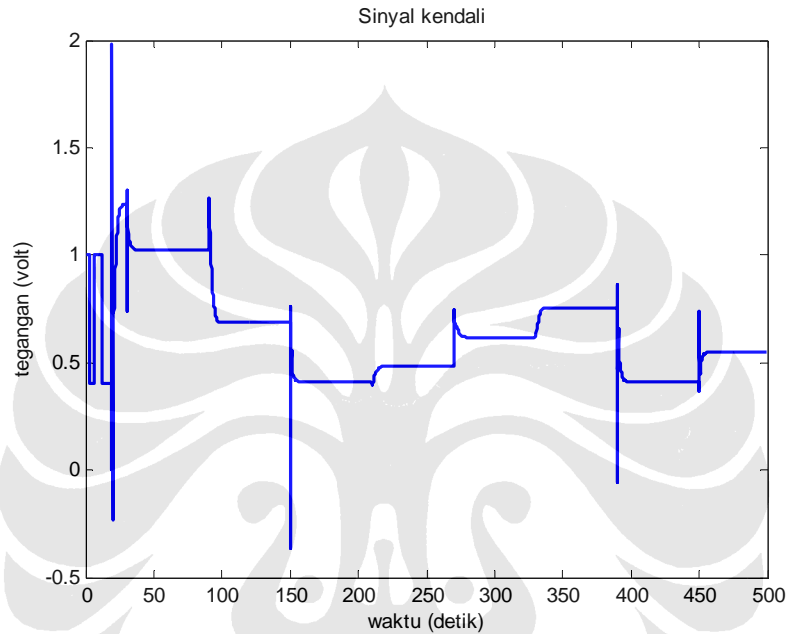
Gambar 4.2. Simulasi penalaan parameter pengendali tanpa supervisi pada *set-point* tereksitasi.

Pada Gambar 4.1, terlihat bahwa perhitungan parameter model berubah dengan cepat pada setiap perubahan *set-point* (Gambar 4.3). Berdasarkan Gambar 4.2, terlihat bahwa perubahan yang signifikan pada estimasi parameter model menyebabkan perubahan yang signifikan pada parameter pengendali. Nilai parameter pengendali yang dihitung menjadi sangat besar.



Gambar 4.3. Simulasi penalaan perbandingan keluaran dan *set-point* tereksitasi tanpa supervisi.

Walaupun perubahan yang signifikan terjadi pada estimasi parameter model dan parameter pengendali, keluaran masih dapat mengikuti *set-point* (Gambar 4.3). Pada gambar 4.4, terlihat beberapa lonjakan sinyal kendali dan beberapa sinyal kendali yang bernilai negatif. Hal ini akan menyebabkan pemotongan sinyal kendali apabila diaplikasikan pada sistem sebenarnya dan menyebabkan hasil pengendalian yang kurang baik.

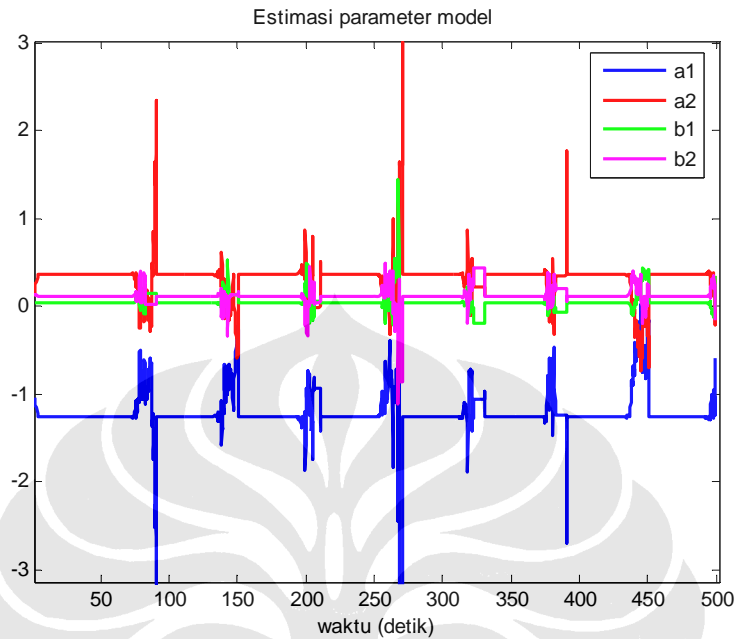


Gambar 4.4. Simulasi perbandingan keluaran dan sinyal kendali tanpa supervisi pada *set-point* tereksitasi.

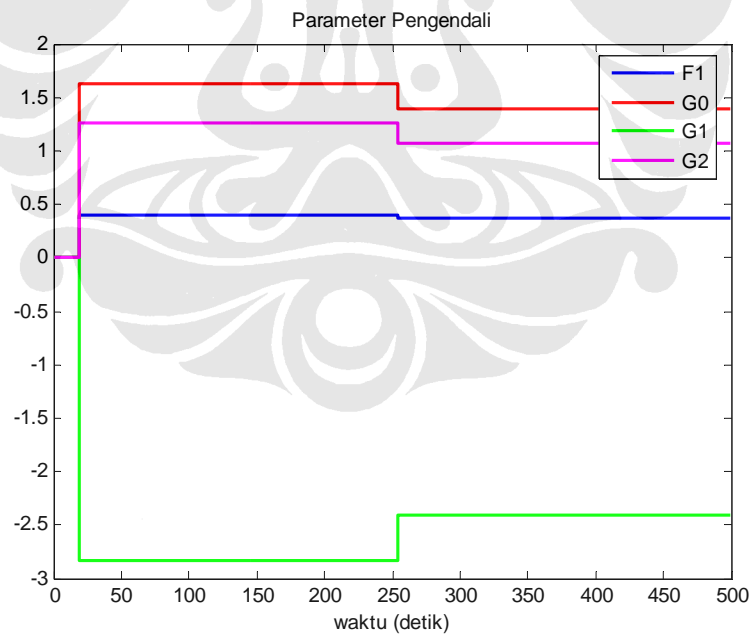
Selanjutnya, simulasi dilakukan pada sistem yang sama dengan algoritma supervisi. Dengan nilai ε_1 (batas nilai determinan matriks H_{22}) dan ε_2 (batas nilai *norm* variansi parameter model) yang ditentukan diperoleh hasil simulasi yang terlihat pada Gambar 4.5-Gambar 4.8.

Pada gambar 4.5, hasil estimasi parameter model dengan algoritma supervisi menyerupai hasil estimasi parameter tanpa supervisi (Gambar 4.1). Perbedaan yang sangat jelas terlihat pada grafik parameter pengendali (Gambar 4.2 dan Gambar 4.6). Pada simulasi dengan level supervisi, perhitungan parameter pengendali hanya dilakukan pada kondisi tertentu saja, yaitu pada saat perhitungan estimasi parameter model masih dilakukan (nilai determinan $H_{22} > \varepsilon_1$ pada Gambar 4.7), parameter model mulai mencapai kondisi tunak (*norm* variansi parameter sistem lebih kecil dari ε_2 sebanyak m pencuplikan berturut-turut pada

Gambar 4.8), dan *pole* sistem stabil (*pole* berada di dalam lingkaran berjari-jari 1 pada bidang-z yang terlihat pada Gambar 4.9).



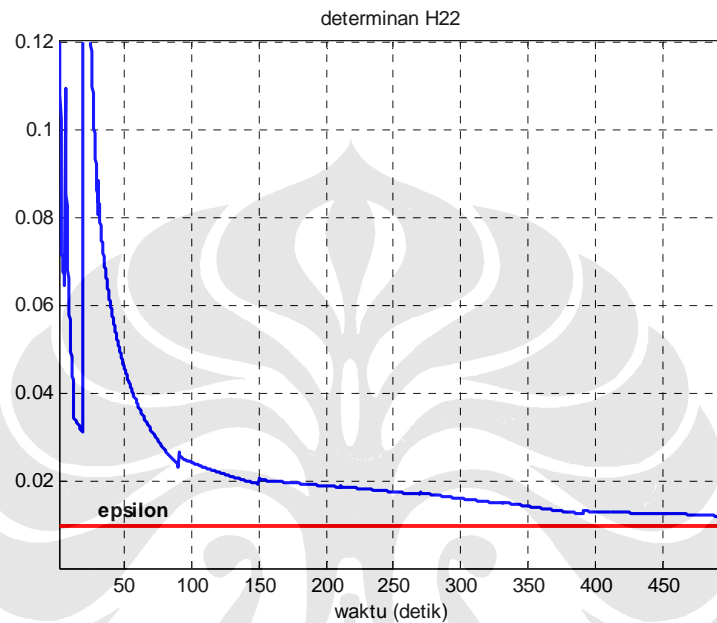
Gambar 4.5. Simulasi estimasi parameter model dengan supervisi pada *set-point* tereksitasi.



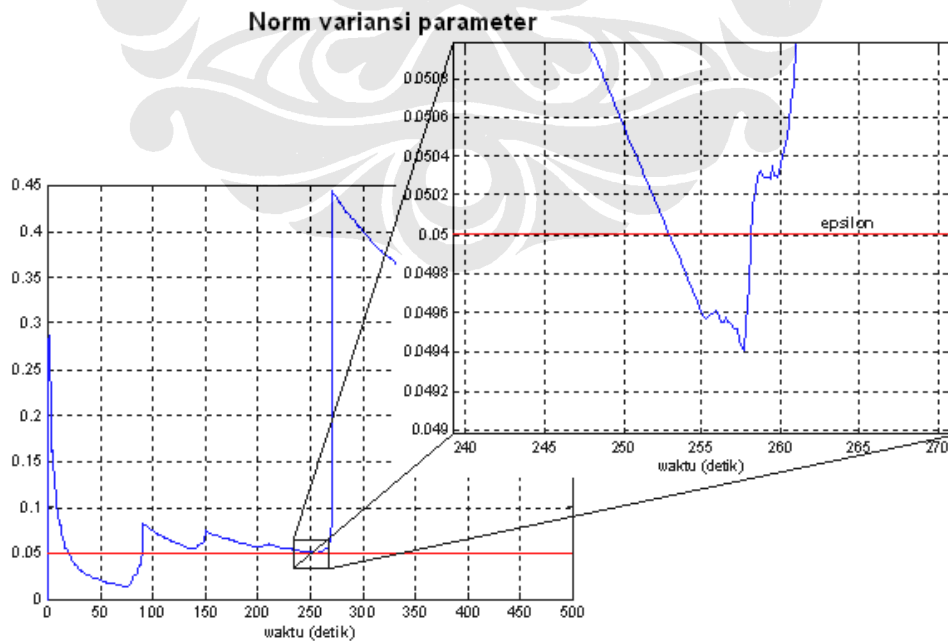
Gambar 4.6. Simulasi penalaan parameter pengendali dengan supervisi pada *set-point* tereksitasi.

Dari besaran *norm* variansi parameter model (Gambar 4.8), terlihat bahwa pada *set-point* tereksitasi nilai *norm* variansi parameter model cenderung

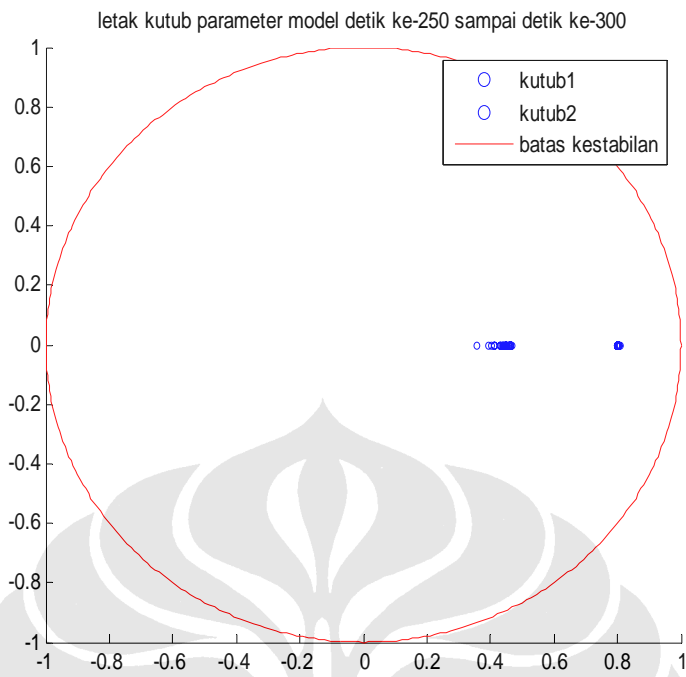
mengalami peningkatan. Jika perhitungan parameter model dipersulit, yaitu dengan menurunkan nilai ε_2 , maka dapat terjadi suatu kondisi *closed-loop* tanpa pengendali. Untuk mencegah kondisi ini, maka parameter pengendali dihitung sesaat setelah identifikasi *open-loop*.



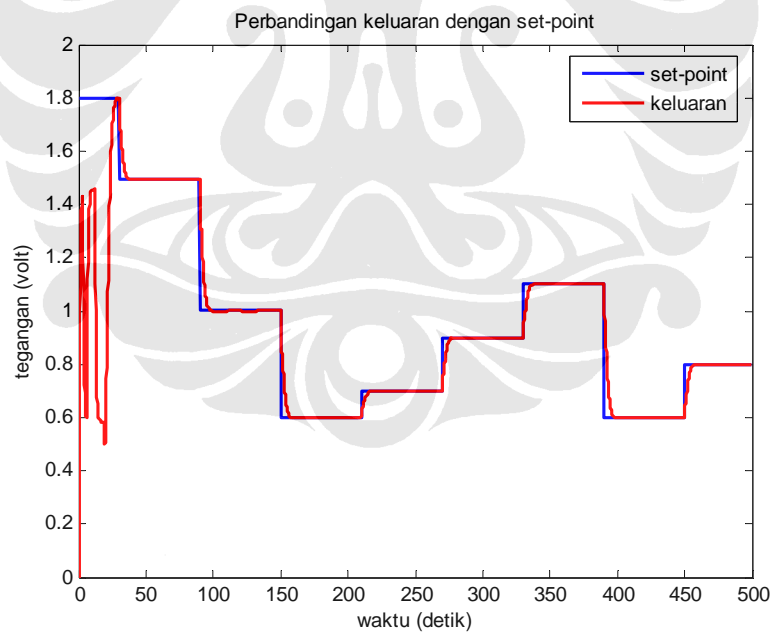
Gambar 4.7. Simulasi determinan H_{22} dengan supervisi pada *set-point*



Gambar 4.8. Simulasi *norm* variansi parameter model dengan supervisi pada *set-point*



Gambar 4.9.Letak kutub model pada simulasi dengan supervisi pada *set-point* tereksitasi.



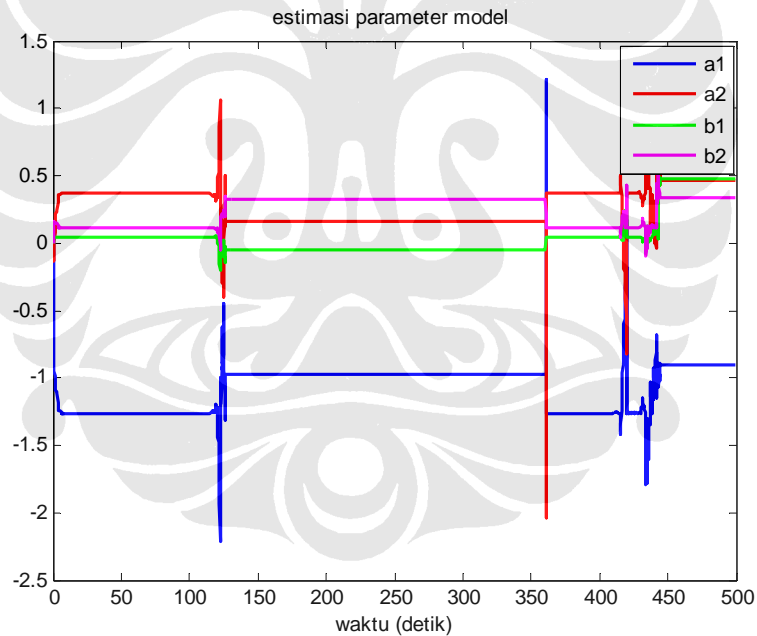
Gambar 4.10.Simulasi perbandingan keluaran dengan *set-point* tereksitasi dengan supervisi.

Dari gambar 4.10, dengan penambahan algoritma supervisi, keluaran dapat mengikuti *set-point* tanpa adanya *overshoot*. Hal ini menunjukkan hasil

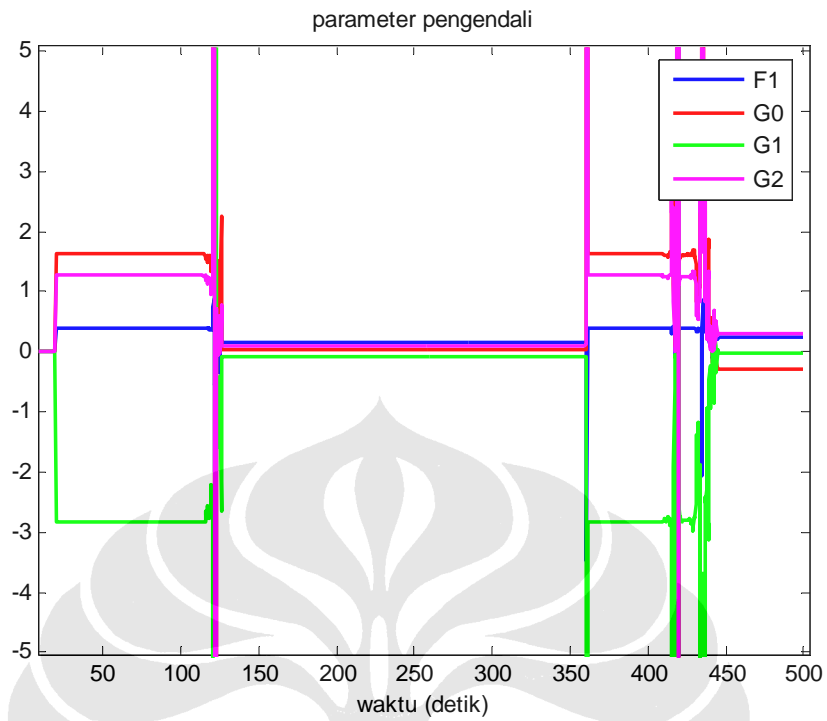
pengendalian yang lebih baik dibandingkan hasil pengendalian tanpa algoritma supervisi (gambar 4.4).

4.1.6 Simulasi penerapan level supervisi pada *set-point* kurang tereksitasi

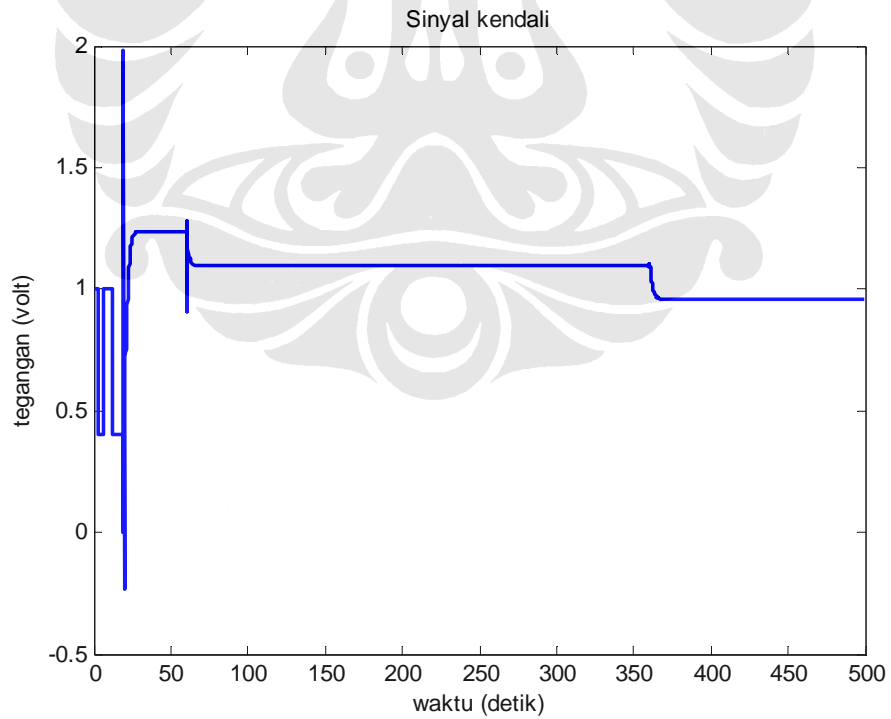
Simulasi ini dilakukan terhadap model pada *set-point* kurang tereksitasi dengan $\lambda = 0.85$. Pertama, simulasi dilakukan tanpa adanya level supervisi. Dari grafik estimasi parameter model (gambar 4.11), terlihat bahwa perubahan signifikan nilai estimasi parameter sistem terjadi pada saat perubahan *set point* (Gambar 4.11). Selain itu, estimasi parameter model juga beresilasi menjelang akhir simulasi walaupun tidak terdapat perubahan *set-point*. Hal ini menunjukkan fenomena *bursting*. Perubahan yang cukup besar pada estimasi parameter model ini menyebabkan perubahan yang lebih besar pada parameter pengendali (gambar 4.12).



Gambar 4.11. Simulasi estimasi parameter model tanpa supervisi pada *set-point* kurang tereksitasi.

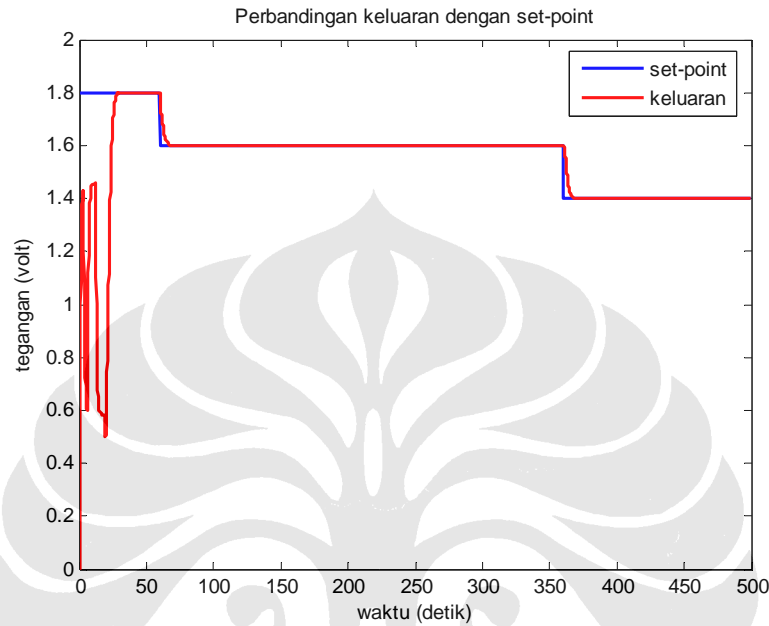


Gambar 4.12. Simulasi parameter pengendali tanpa supervisi pada *set-point* kurang tereksitasi.

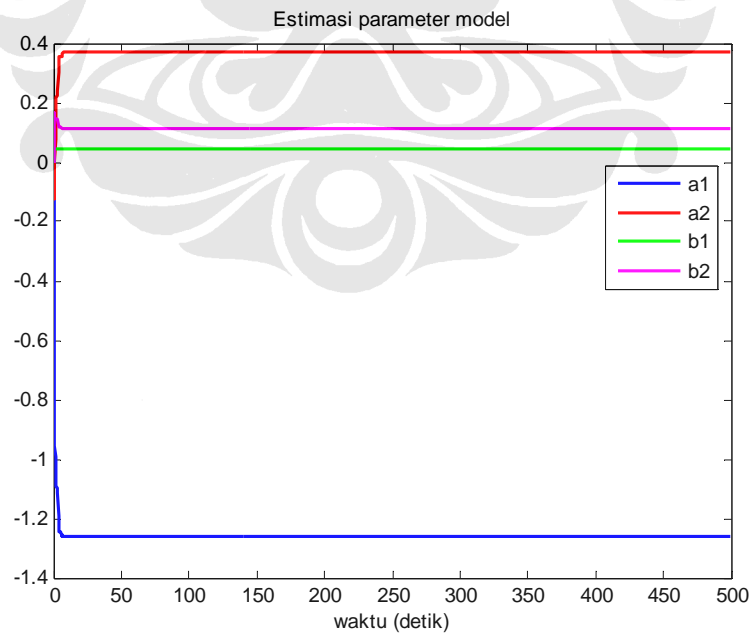


Gambar 4.13. Sinyal kendali tanpa supervisi pada simulasi dengan *set-point* kurang tereksitasi.

Walaupun nilai estimasi parameter model dan parameter pengendali berhasil, pada simulasi, sinyal kendali (Gambar 4.13) dan keluaran sistem lingkaran-tertutup masih stabil dan dapat mengikuti *set-point* yang diinginkan (Gambar 4.14).

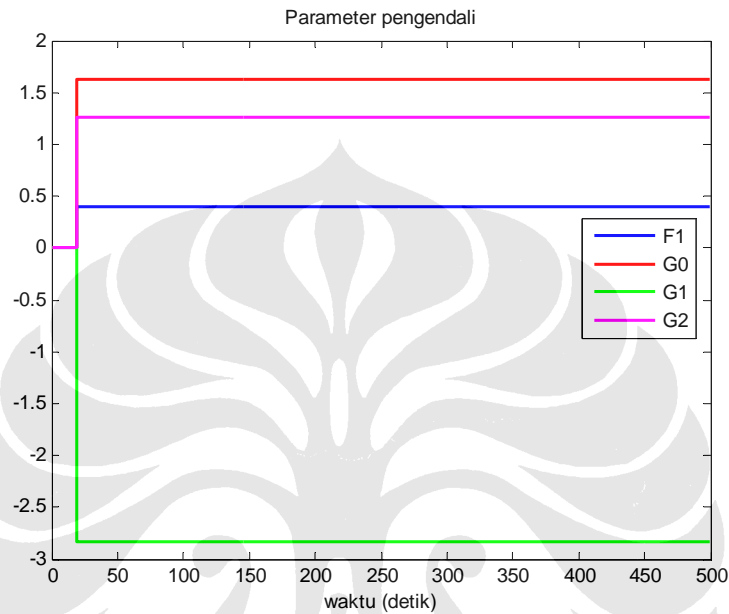


Gambar 4.14. Simulasi perbandingan keluaran dengan *set-point* kurang tereksitasi tanpa supervisi.

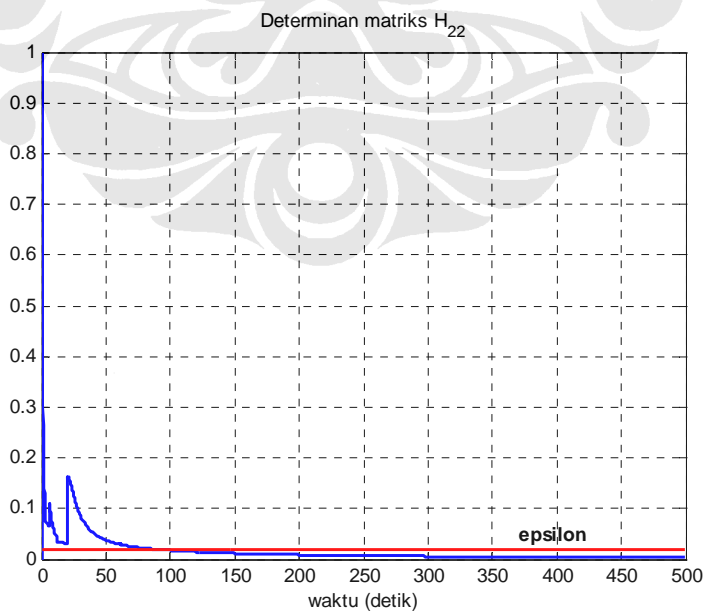


Gambar 4.15. Simulasi estimasi parameter model dengan supervisi pada *set-point* kurang tereksitasi.

Simulasi berikutnya adalah simulasi pada model yang sama dengan $\lambda=0.85$ dengan menerapkan algoritma supervisi. Dengan nilai ϵ_1 (batas nilai determinan matriks H_{22}) dan ϵ_2 (batas nilai *norm* variansi parameter model) yang sudah ditentukan, diperoleh hasil simulasi yang terlihat pada Gambar 4.15- Gambar 4.21.

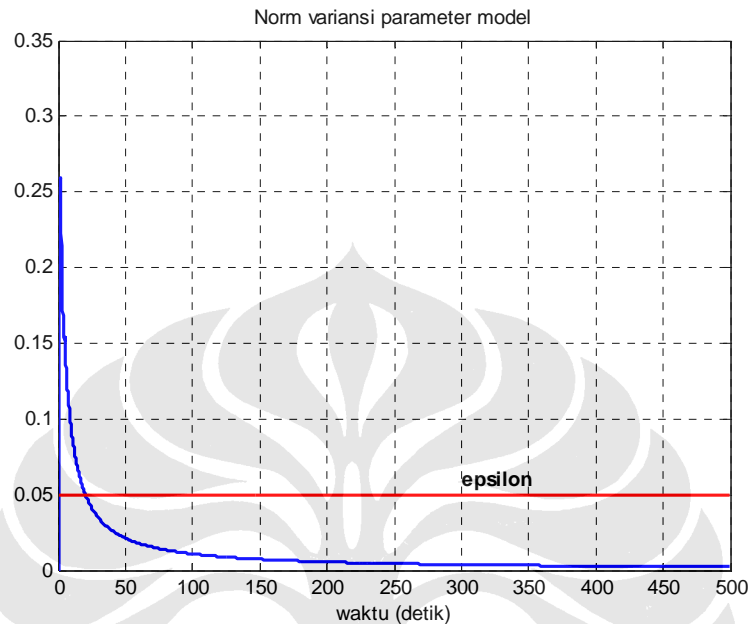


Gambar 4.16. Simulasi parameter pengendali dengan supervisi pada *set-point* kurang tereksitasi.



Gambar 4.17. Simulasi determinan matriks H_{22} dengan supervisi pada *set-point* kurang tereksitasi.

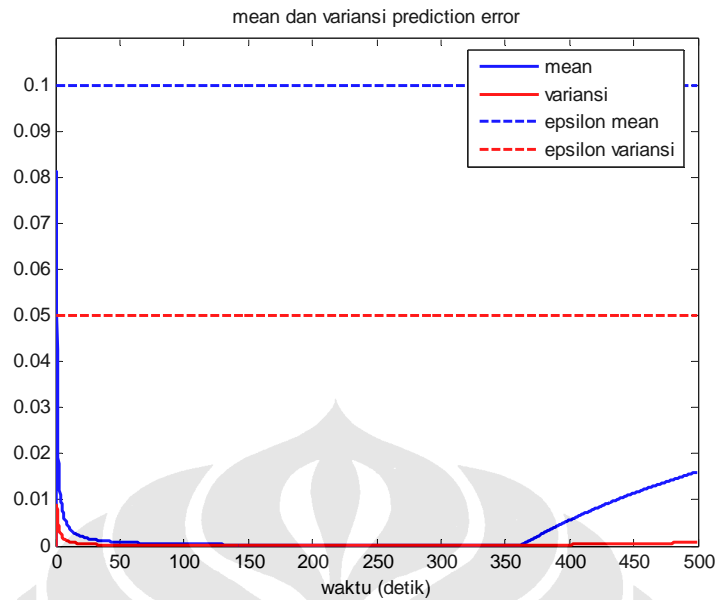
Dengan parameter model dan parameter pengendali yang konstan, terlihat bahwa sinyal kendali (gambar 4.18) dan keluaran (gambar 4.19) dapat mengikuti *set-point* yang diinginkan.



Gambar 4.20. Simulasi *norm* variansi parameter model dengan supervisi pada *set-point* kurang tereksitasi.

Seperti nilai determinan matriks H_{22} , nilai *norm* variansi parameter model (gambar 4.22) menunjukkan nilai yang menurun yang dikarenakan nilai parameter model yang konstan. Tetapi besaran *mean* dan variansi *prediction error* meningkat setelah perubahan *set-point* terakhir (Gambar 4.23).

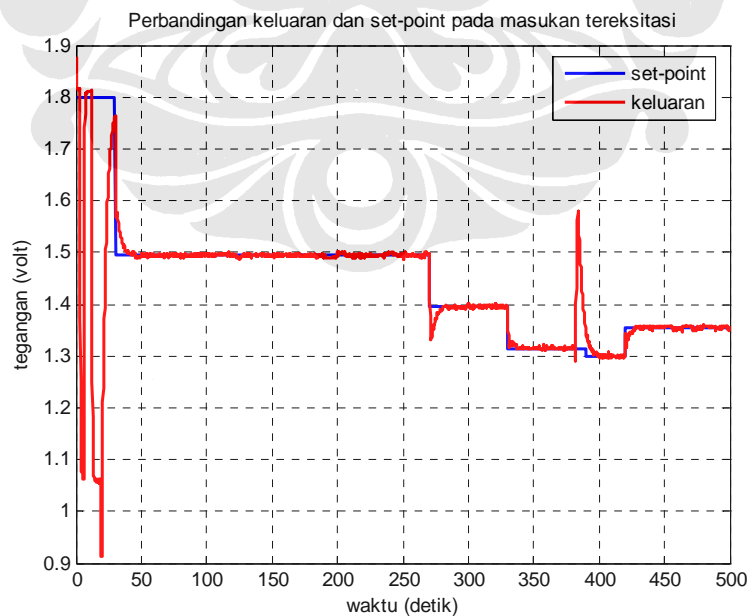
Dalam penjelasan bab 3, nilai *mean* dan variansi yang naik secara kontinu membuat level supervisi menghentikan keseluruhan proses. Dari simulasi ini, nilai *mean* dan variansi yang cenderung naik tidak membuat kinerja pengendali menjadi buruk. Sehingga pada algoritma level supervisi ditambahkan batasan maksimum untuk *mean prediction error*, ϵ_3 , dan batasan maksimum untuk variansi *prediction error*, ϵ_4 . Nilai *mean prediction error* dan variansi yang naik secara kontinu dan mencapai batas-batas maksimum yang ditentukan menunjukkan bahwa jenis model untuk estimator tidak sesuai dengan sistem.



Gambar 4.21. Simulasi *mean* dan *variansi* prediction error dengan supervisi pada *set-point* kurang tereksitasi.

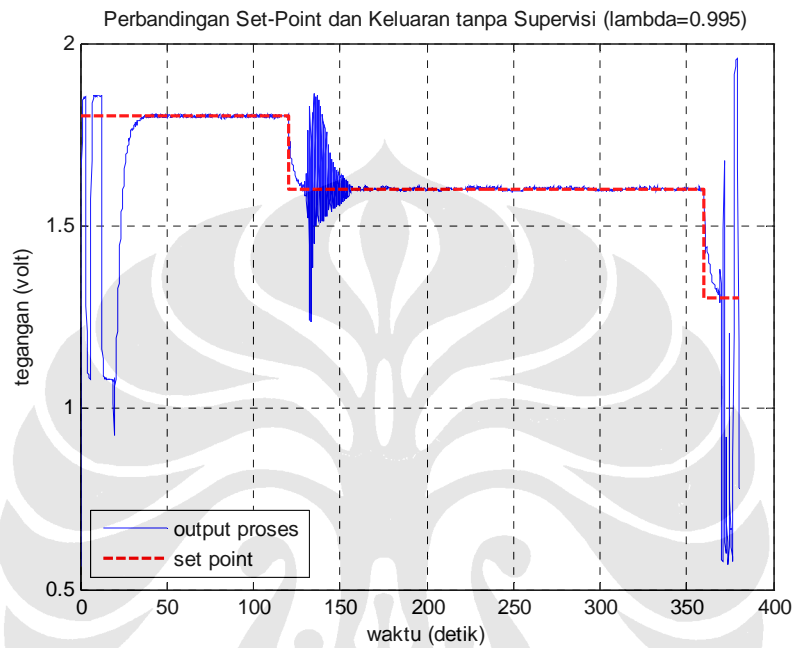
4.2 FENOMENA BURSTING PADA UJI EKSPERIMEN PENGENDALI SWATAKA PRESSURE PROCESS RIG

Pada pengendalian adaptif dengan $\lambda < 1.0$ (mempunyai mekanisme adaptasi) mempunyai kinerja yang baik pada *set-point* yang tidak konstan (gambar



Gambar 4.22. Perbandingan keluaran dan *set-point* pada kondisi *bursting*.

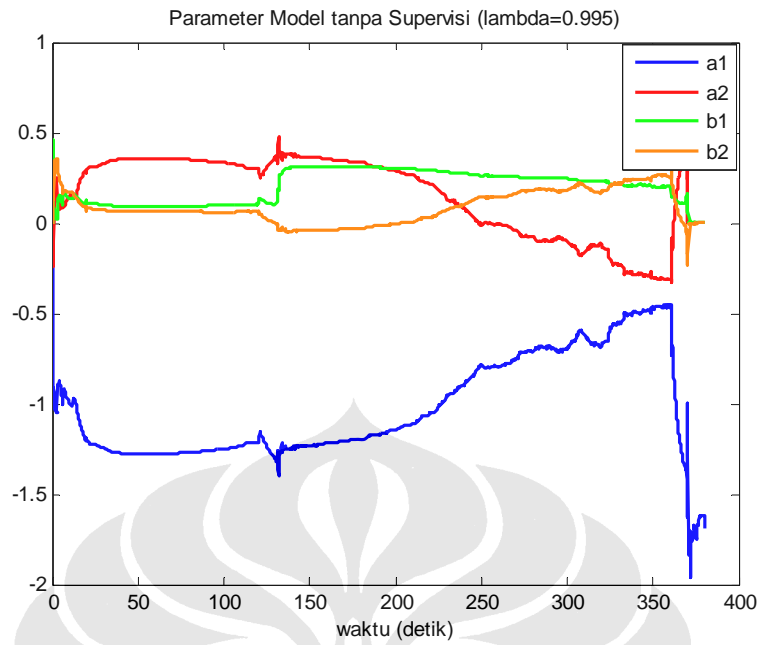
4.22). Sedangkan pada *set-point* yang cenderung konstan, pengendali adaptif dengan $\lambda < 1.0$ akan menyebabkan terjadinya fenomena *bursting* seperti terlihat pada gambar 4.23. Uji eksperimen ini dilakukan pada *Pressure Process Rig* (Feedback 38-714) dengan $\lambda = 0.995$.



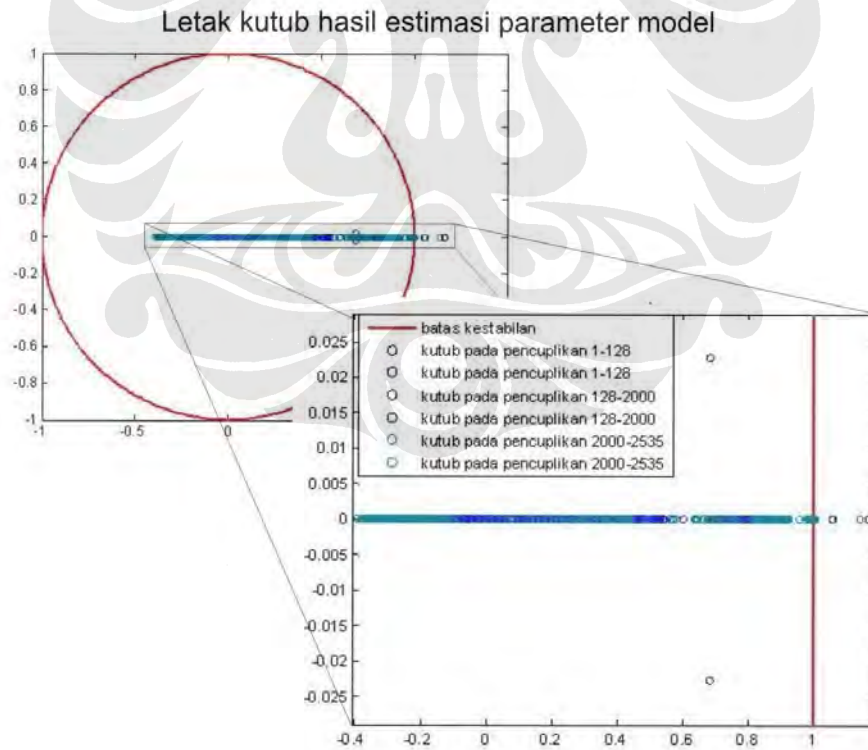
Gambar 4.23.Perbandingan keluaran dan *set-point* pada kondisi *bursting*.

Dari Gambar 4.23, terlihat bahwa fenomena *bursting* terjadi pada detik ke-100 sampai detik ke-200 dan setelah 350 detik. *Bursting* pertama masih dapat kembali mengikuti *set-point*, sedangkan *bursting* yang kedua tidak dapat mengikuti *set-point*.

Sistem menjadi tidak terkendali karena meletupnya estimasi parameter model seperti terlihat pada gambar 4.24. Dari estimasi parameter di atas dapat kita tentukan letak kutub pada setiap pencuplikan terlihat pada gambar 4.25.



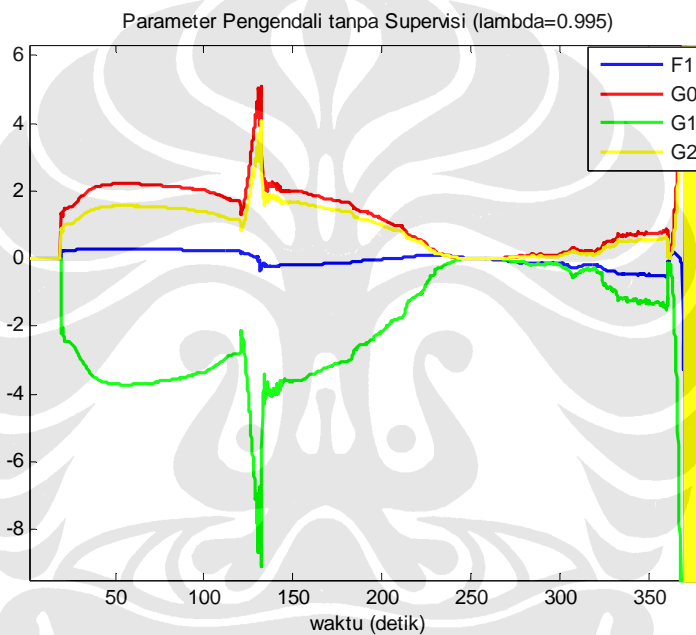
Gambar 4.24. Estimasi parameter pada kondisi *bursting*.



Gambar 4.25. Letak kutub hasil estimasi parameter pada kondisi *bursting*.

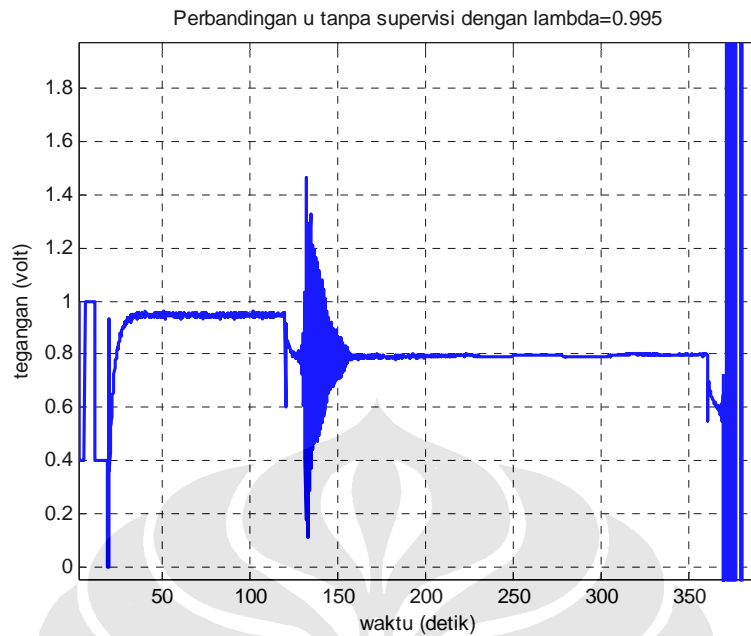
Terlihat bahwa letak kutub hasil estimasi parameter model berada di luar batas kestabilan pada saat identifikasi *open loop* (pencuplikan 1-128) dan pada pencuplikan yang lebih besar dari 2000 (lebih dari 350 detik). Hasil estimasi ini menunjukkan bahwa sistem tidak stabil.

Estimasi parameter model yang buruk menyebabkan hasil penalaan parameter pengendali yang buruk pula. Hal ini dapat dilihat pada gambar 4.26. Perubahan parameter pengendali sangat besar dan sangat cepat pada saat estimasi parameter model meletup (lebih dari detik ke-350).



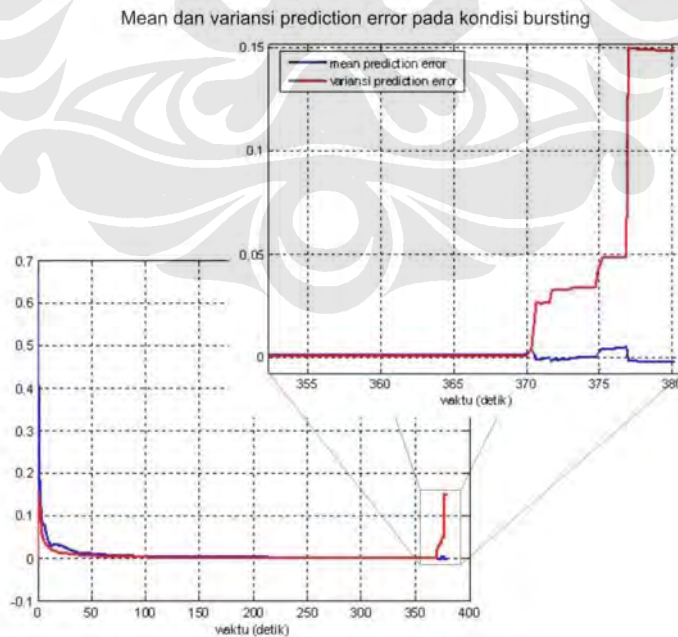
Gambar 4.26.Estimasi parameter pada kondisi *bursting*.

Penalaan parameter yang buruk menyebabkan perhitungan sinyal kendali (gambar 4.27) dan hasil keluaran yang buruk pula sehingga sistem *closed loop* keseluruhan menjadi tidak stabil.



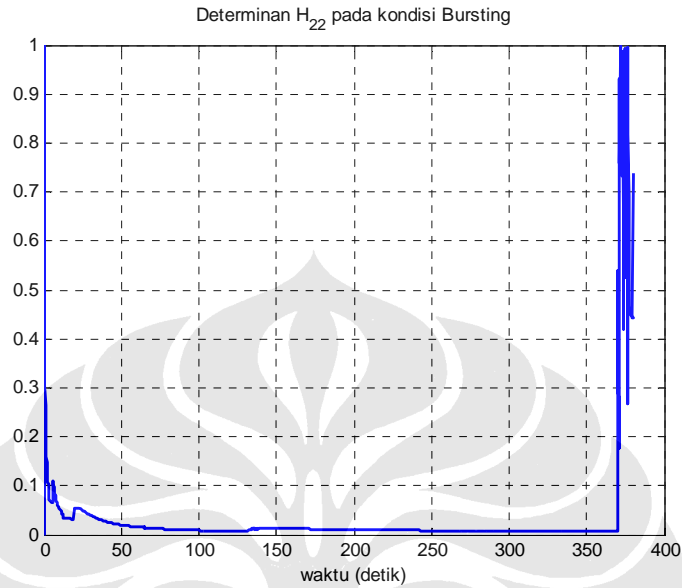
Gambar 4.27. Perbandingan sinyal kendali dan keluaran pada kondisi *bursting*.

Pada kondisi *bursting*, besaran – besaran yang menjadi parameter dalam pengambilan keputusan pada level supervisi mengalami kenaikan (Gambar 4.28- Gambar 4.30).



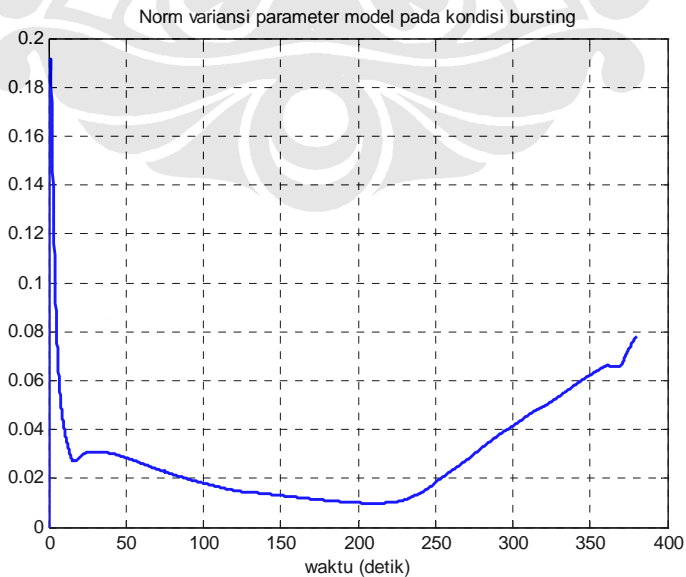
Gambar 4.28. Nilai *mean* dan *variansi prediction error* pada kondisi *bursting*.

Pada Gambar 4.28, nilai variansi *prediction error* meningkat tajam pada kondisi *bursting*. Tetapi nilai rata-rata *prediction error* mengalami penurunan. Hal ini dikarenakan adanya nilai negatif dari *prediction error*.



Gambar 4.29.Nilai Determinan matriks H_{22} pada kondisi *bursting*.

Nilai determinan matriks H_{22} juga menunjukkan lonjakan yang signifikan pada saat terjadinya fenomena *bursting* yang terlihat pada Gambar 4.29. Hal ini dikarenakan perubahan nilai sinyal kendali yang sangat besar pada saat terjadi *bursting*.



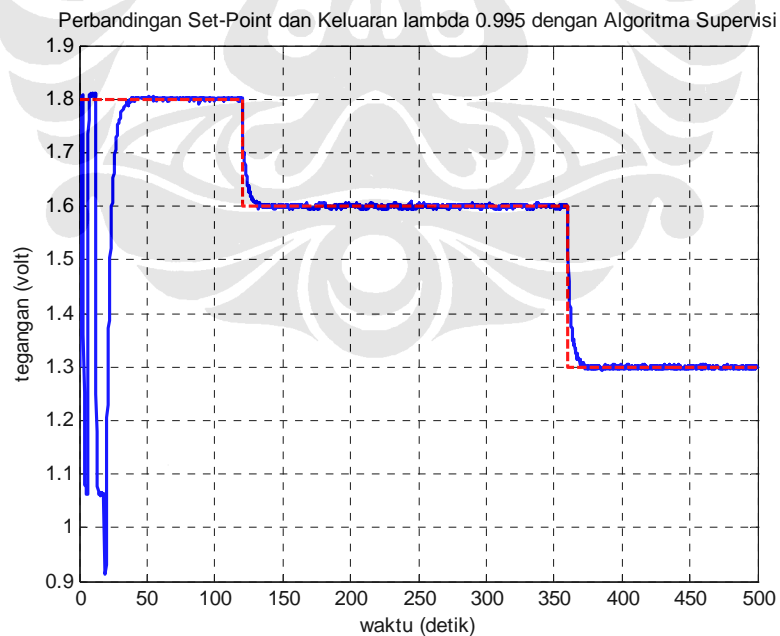
Gambar 4.30.Norm variansi parameter model pada kondisi *bursting*.

Pada Gambar 4.30, nilai *norm* variansi parameter model mulai mengalami kenaikan dari detik ke-200. Hal ini dikarenakan perubahan nilai estimasi parameter model yang kontinu pada saat tersebut (Gambar 4.24). Pada saat terjadi *bursting*, $t > 350$ detik, besaran ini menunjukkan kenaikan yang semakin tajam yang diakrenakan perubahan estimasi parameter model yang semakin besar.

Dari pantauan besaran-besaran di atas, fenomena *bursting* dapat dibedakan dari pengendalian yang baik. Sehingga berdasarkan pantauan rata-rata dan variansi *prediction error*, determinan matriks H_{22} , *norm* variansi parameter model, dan letak kutub sistem, kita dapat membuat keputusan-keputusan untuk mengatasi terjadinya fenomena *bursting*.

4.3 UJI EKSPERIMEN LEVEL SUPERVISI PADA SET-POINT KURANG TEREKSITASI

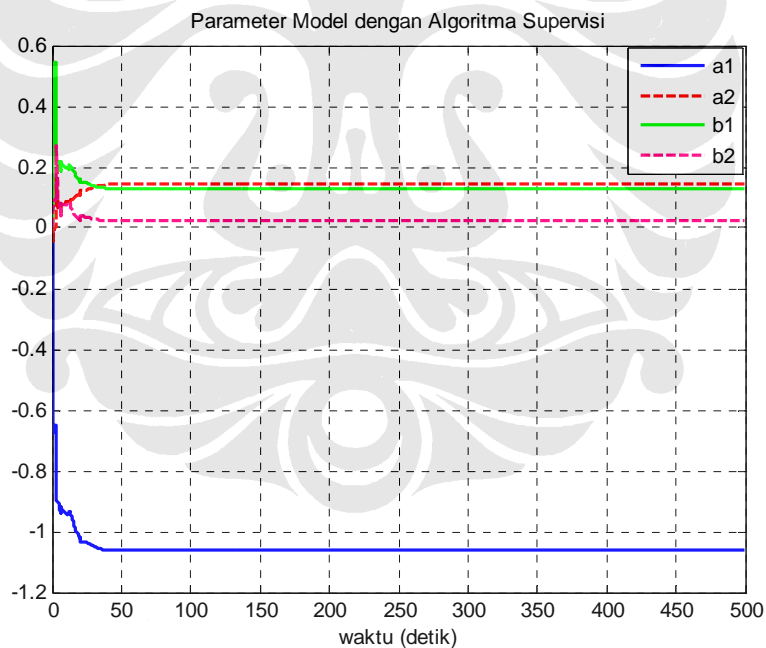
Dengan konfigurasi yang sama dengan uji eksperimen kondisi *bursting*, penerapan level supervisi menunjukkan hasil yang lebih baik. Fenomena *bursting* dapat diatasi dengan adanya level supervisi yang terlihat pada perbandingan keluaran dan *set-point* (gambar 4.31).



Gambar 4.31. Hasil eksperimen perbandingan keluaran dan *set-point* kurang tereksitasi dengan algoritma supervisi.

Dengan batas nilai determinan matriks H_{22} , ε_1 , batas *norm* variansi parameter model, ε_2 , batas maksimum untuk *mean prediction error*, ε_3 , dan batas maksimum untuk variansi *prediction error*, ε_4 yang ditentukan, hasil estimasi parameter model sistem terlihat seperti Gambar 4.32. Pada saat identifikasi *open-loop* (0 – 19.2 detik) nilai estimasi parameter model masih berubah-ubah. Beberapa detik setelah identifikasi *open loop* (± 40 detik) nilai parameter model sudah cukup baik sehingga perhitungan parameter model tidak diperlukan lagi dan menggunakan parameter model yang sudah ada. Baiknya kinerja keseluruhan bagian pengendali terlihat dari nilai determinan matriks H_{22} (Gambar 4.33).

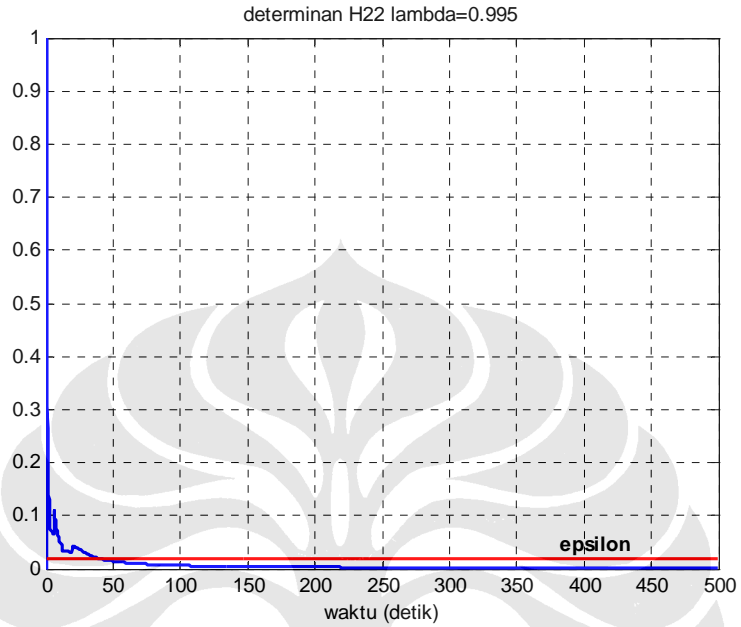
Sedangkan perhitungan parameter pengendali hanya dilakukan 1 kali, yaitu sesaat setelah identifikasi model *open-loop* (Gambar 4.35). Nilai ini tidak berubah sampai akhir karena tidak memenuhi kondisi perhitungan parameter pengendali (detik ke 19.2 – 40) dan sudah cukup baiknya kinerja sistem secara keseluruhan (detik ke 40 – 500).



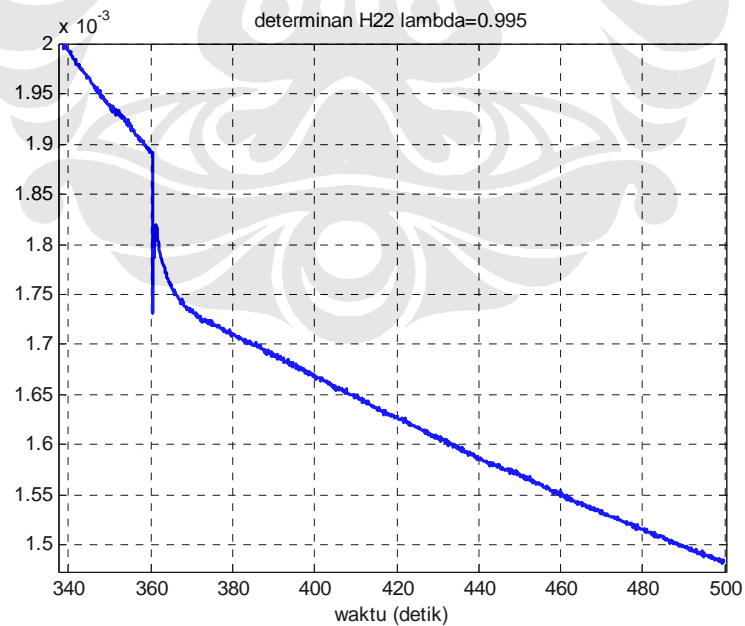
Gambar 4.32. Hasil eksperimen parameter model dengan supervisi pada *set-point* kurang tereksitasi.

Baik buruknya kinerja sistem secara keseluruhan didapatkan dengan memantau determinan matriks H_{22} (gambar 4.33). Dari gambar 4.33, terlihat

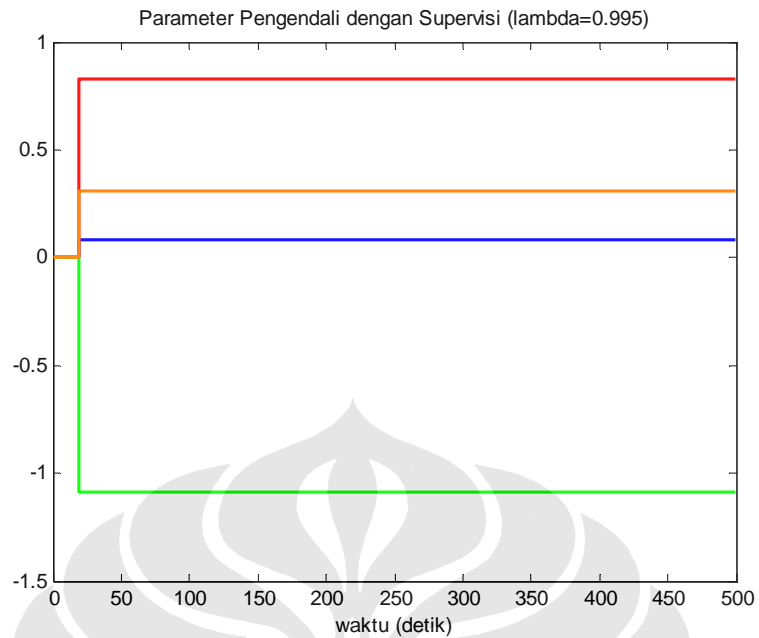
bahwa nilai determinan matriks H_{22} semakin menurun. Hal ini menunjukkan bahwa nilai sinyal kendali cenderung konstan.



Gambar 4.33. Hasil eksperimen determinan H_{22} dengan supervisi pada *set-point* kurang tereksitasi.

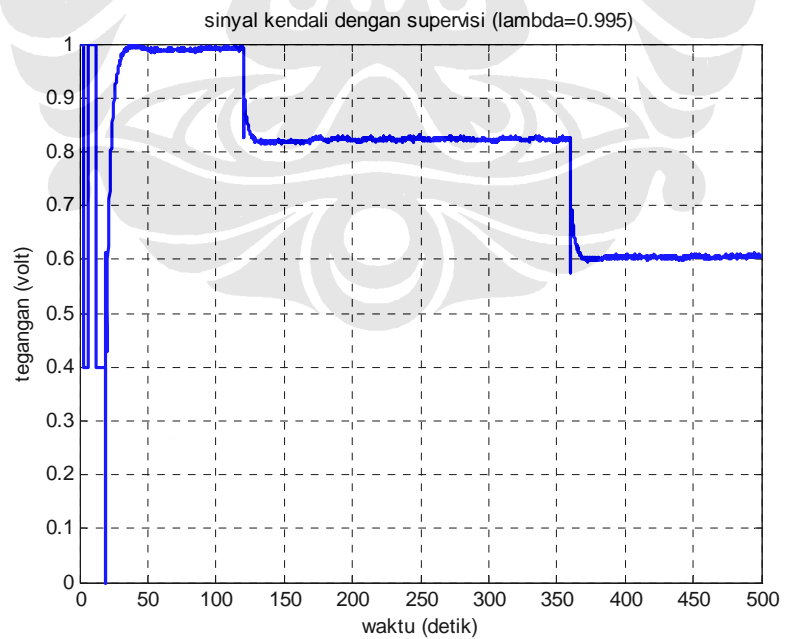


Gambar 4.34. Hasil eksperimen determinan H_{22} pada saat perubahan *set-point* dengan supervisi pada *set-point* kurang tereksitasi.



Gambar 4.35. Hasil perhitungan parameter pengendali dengan supervisi pada *set-point* kurang tereksitasi.

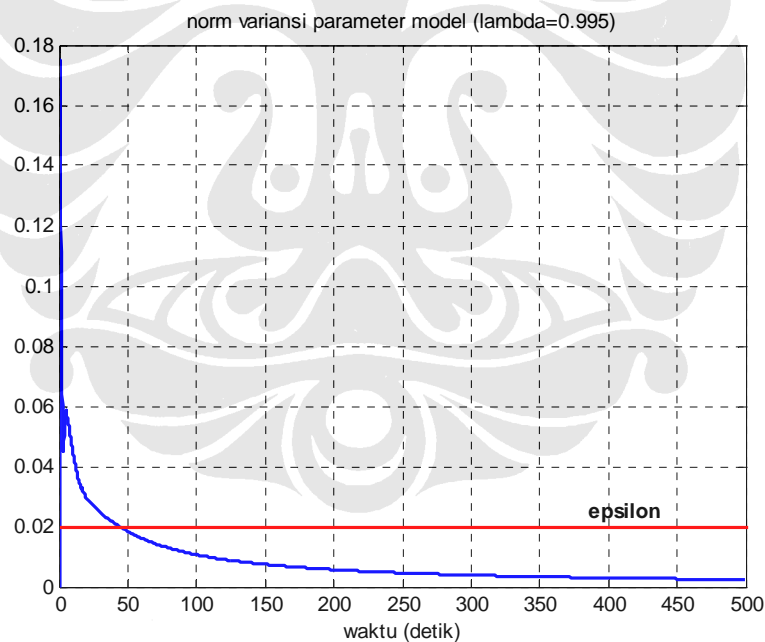
Sinyal kendali yang konstan menandakan bahwa estimasi parameter model dan disain pengendali sudah baik. Pada gambar 4.34 terlihat bahwa perubahan *set-*



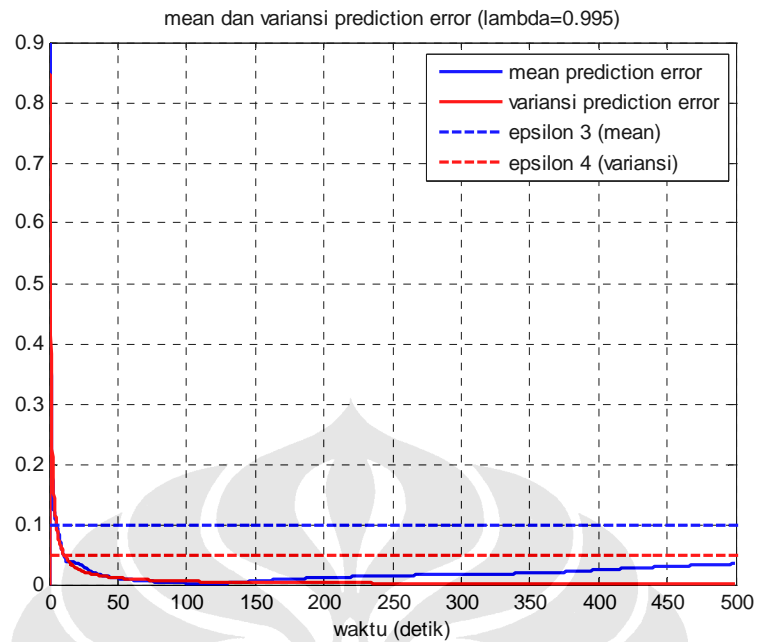
Gambar 4.36. Sinyal kendali dengan supervisi pada *set-point* kurang tereksitasi.

point menyebabkan perubahan nilai determinan matriks H_{22} karena perubahan sinyal kendali. Kemudian nilai besaran ini kembali turun yang menandakan bahwa sinyal kendali yang diberikan ke sistem kembali konstan (Gambar 4.36). Hal ini menunjukkan bahwa estimasi model dan disain pengendali lama cukup baik untuk mengendalikan sistem pada titik kerja yang baru.

Besaran *norm* variansi parameter pada gambar 4.36 menunjukkan penurunan karena nilai parameter model konstan. Dan pada gambar 4.37 terlihat bahwa nilai *mean prediction error* cenderung naik setelah perubahan *set-point*. Hal ini menunjukkan bahwa nilai *prediction error* terus naik atau dengan kata lain estimasi model kurang baik untuk titik kerja yang baru. Walaupun terus naik, nilai *mean prediction error* ini masih di bawah batas maksimal yang ditentukan, ϵ_3 , sehingga masih dianggap cukup mewakili sistem. Hal ini dapat dilihat pada perbandingan *set-point* dengan keluaran (Gambar 4.31).



Gambar 4.37. Hasil eksperimen norm variansi parameter model dengan supervisi pada *set-point* kurang tereksitasi.



Gambar 4.38. Hasil eksperimen *mean* dan *variansi prediction error* dengan supervisi pada *set-point* kurang tereksitasi.

Selain mengatasi fenomena *bursting*, level supervisi dapat mengurangi waktu komputasi. Hal ini dikarenakan perhitungan beberapa algoritma tidak dilakukan.

BAB 5

KESIMPULAN

Dari keseluruhan pembahasan skripsi dan hasil uji eksperimen pengendalian system *Pressure Process Rig* ini, dapat disimpulkan beberapa hal, yaitu:

1. Fenomena *bursting* terjadi pada uji eksperimen pengendali adaptif dengan *set-point* kurang tereksitasi, 2 buah kutub lingkaran tertutup spesifikasi $+0.9$, dan mempunyai mekanisme adaptasi (faktor pelupa $=0.995$).
2. Hasil uji eksperimen penerapan algoritma supervisi pada pengendali adaptif dengan Struktur Kendali Kenaikan *Pressure Process Rig* (*Feedback 38-714*) pada *set-point* kurang tereksitasi dan faktor pelupa lebih kecil dari 1 dapat mengatasi fenomena *bursting*.
3. Besaran-besaran yang dijadikan parameter dalam logika keputusan pada algoritma supervisi adalah rata-rata dan variansi kesalahan prediksi, determinan matriks H_{22} , *norm* variansi parameter model, dan letak kutub estimasi parameter model.
4. Pada uji eksperimen penerapan level supervisi pada pengendali adaptif *Pressure Process Rig* (*Feedback 38-714*) dapat menghilangkan fenomena *bursting* dengan batas-batas sebagai berikut:
 - a. Rata-rata kesalahan prediksi lebih kecil dari 0.1,
 - b. Variansi kesalahan prediksi lebih kecil dari 0.05,
 - c. Determinan matriks H_{22} lebih kecil dari 0.02, dan
 - d. *Norm* variansi parameter model lebih kecil dari 0.05.
5. Nilai rata-rata dan variansi kesalahan prediksi yang naik secara kontinu di bawah batas maksimum rata-rata dan batas maksimum variansi masih dapat menunjukkan baiknya kinerja estimasi parameter model.
6. Nilai determinan matriks H_{22} di bawah batas ε dapat menjadi parameter yang menunjukkan baiknya kinerja keseluruhan sistem lingkaran tertutup pengendali adaptif, yaitu estimasi parameter model dan disain parameter pengendali.

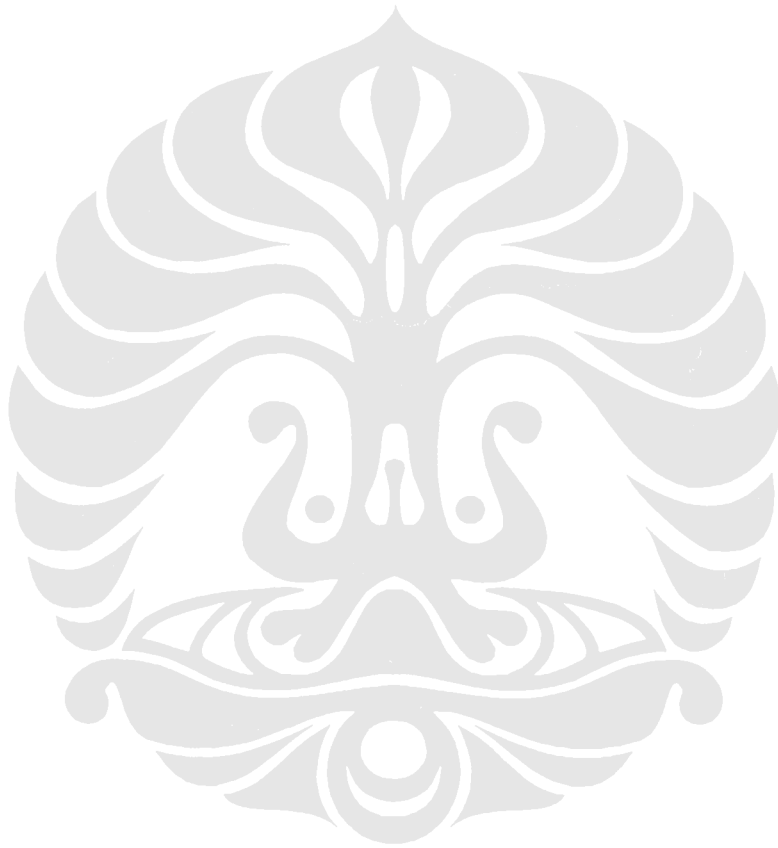
DAFTAR ACUAN

- [1] PLN P3B Jawa Bali, “Rencana Usaha Penyediaan Tenaga Listrik 2008 - 2017” tahun 2007
- [2] Alexandra Von Meier, *Electric Power Systems A Conceptual Introduction* (New Jersey: John Wiley & Sons Ltd., 2006)
- [3] PSR. Murty, *Power System Operation and Control*. (New Delhi:Tata Mc.Graw Hill Publishing Company limited,1984)
- [4] B.M. Weedy, B.J. Cory, *Electric Power System 4th edition* (West Sussex: John Wiley & Sons Ltd, 1998).
- [5] PT PLN (persero) P3B Sub Region Bali
- [6] PT PLN (persero) Bidang Operasi Sistem
- [7] PLN P3B Distribusi Bali, “Rencana Usaha Penyediaan Tenaga Listrik tahun 2007”
- [8] Adi Purwanto. ”*Analisa Keterjaminan Aliran Daya Dan Biaya Produksi PLN Sub Region Bali Tahun 2008-2017*”. Tesis. Program Pasca Sarjana Fakultas Teknik UI, Depok, 2008.

DAFTAR PUSTAKA

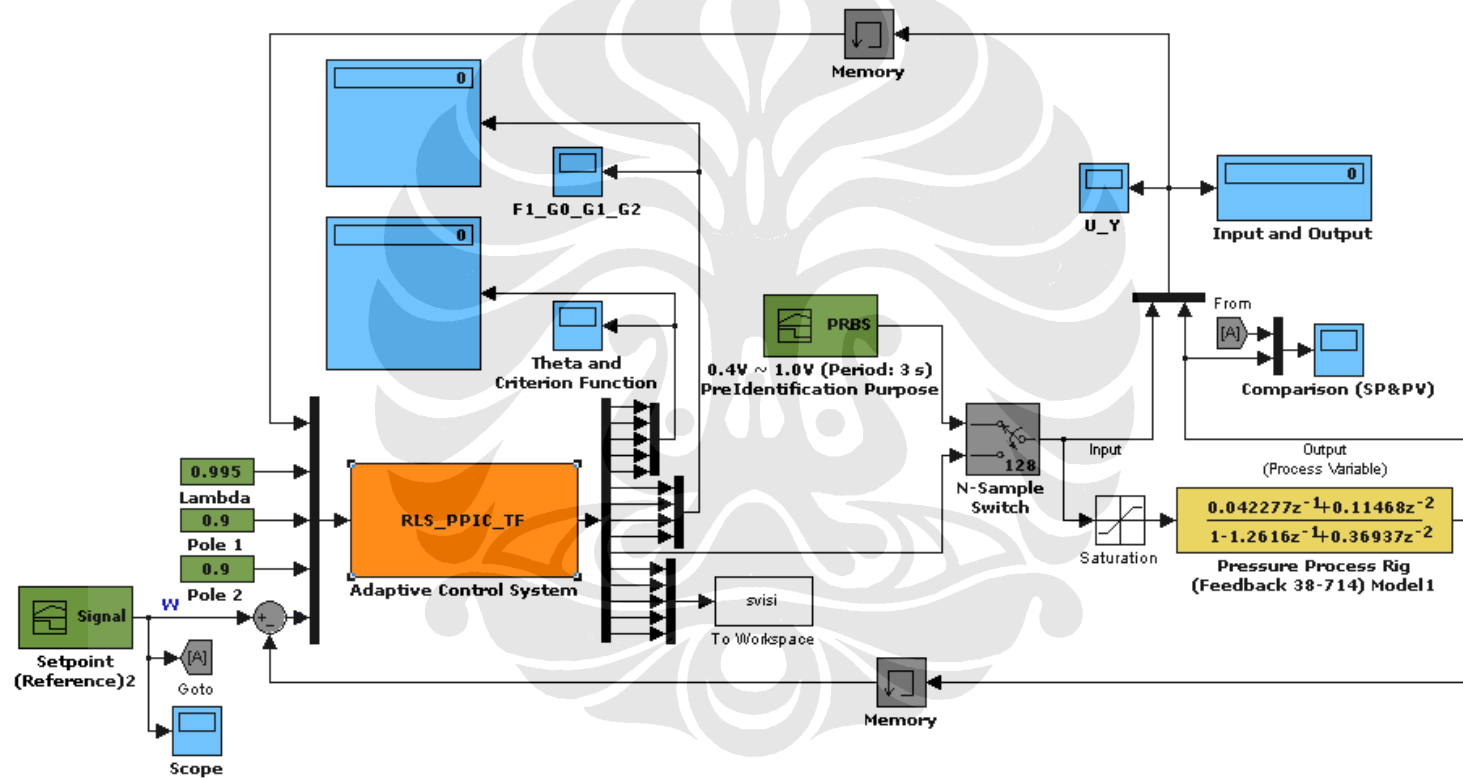
- Meier, Alexandra Von., *Electric Power Systems A Conceptual Introduction* (New Jersey: John Wiley & Sons Ltd., 2006)
- Murty, PSR., *Power System Operation and Control*. (New Delhi:Tata Mc.Graw Hill Publishing Company limited,1984)
- PLN P3B Jawa Bali, “Rencana Usaha Penyediaan Tenaga Listrik 2008 -2017” tahun 2007
- PT PLN (persero) P3B Sub Region Bali
- PT PLN (persero) Bidang Operasi Sistem
- PLN P3B Distribusi Bali, “Rencana Usaha Penyediaan Tenaga Listrik tahun 2007”
- Purwanto, Adi. ”*Analisa Keterjaminan Aliran Daya Dan Biaya Produksi PLN Sub Region Bali Tahun 2008-2017*”. Tesis. Program Pasca Sarjana Fakultas Teknik UI, Depok, 2008.
- Short, T.A., *Electric Power Distribution Handbook*.(CRC Press LLC, 2004)
- Stevenson Jr, William D., *Element of Power System Analysis third edition* (Mc Graw Hill International, 1975)
- Weedy, B.M, B.J. Cory, *Electric Power System 4th edition* (West Sussex: John Wiley & Sons Ltd, 1998)

LAMPIRAN



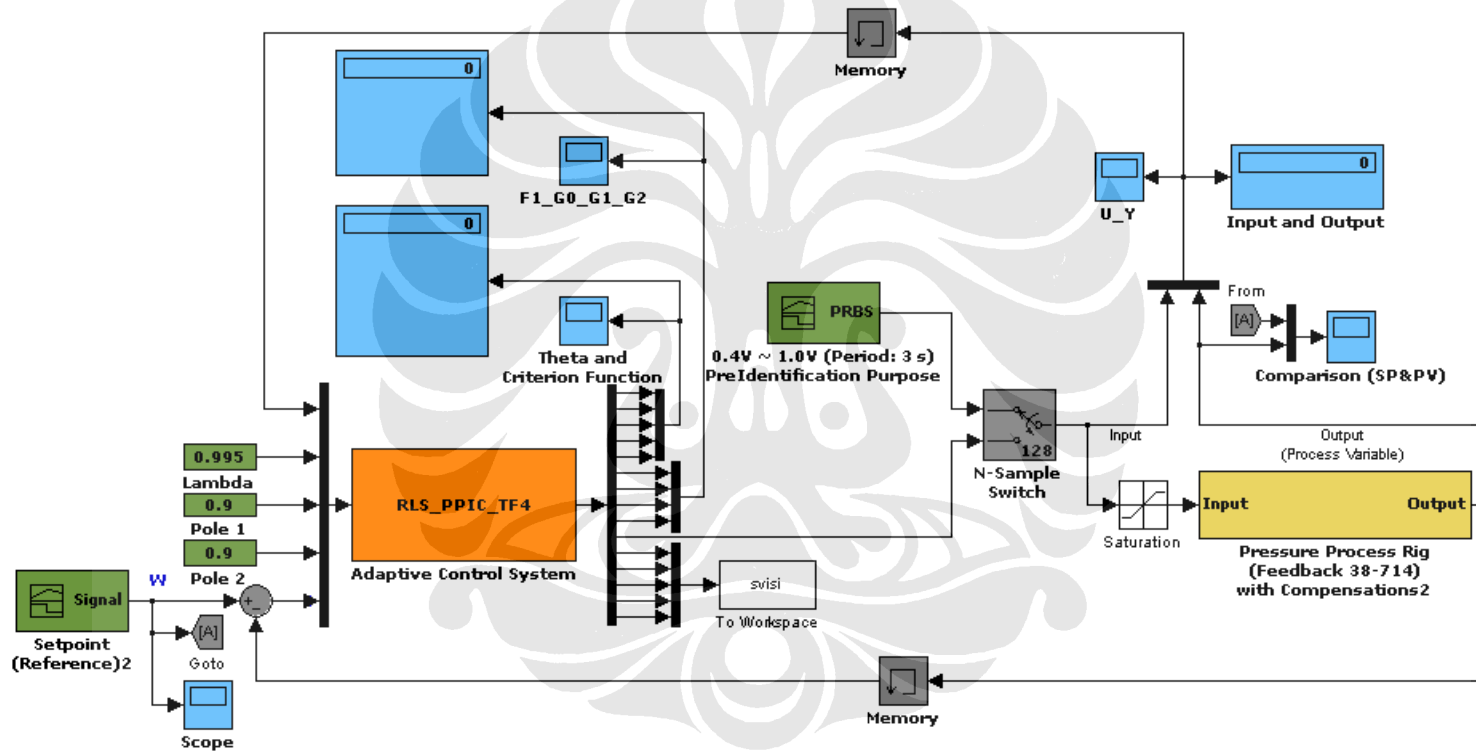
Lampiran 1. Blok SIMULINK

1. Blok SIMULINK pada Simulasi Pengendalian Adaptif dengan Level Supervisi



Gambar L.1. Gambar blok SIMULINK simulasi pengendalian adaptif dengan level supervisi pada model *Pressure Process Rig Feedback 38-714*.

2. Blok SIMULINK pada Uji Eksperimen Pengendalian Adaptif dengan Level Supervisi



Gambar L.2. Gambar blok SIMULINK uji eksperimen pengendalian adaptif dengan level supervisi pada *Pressure Process Rig 38-714*

