

## Program Aplikasi Rekonsiliasi Data untuk Pendeteksian *Gross Error* pada Sistem Tangki Ganda yang Berinteraksi

<sup>1</sup>Andreas Julianto Sutrisno, <sup>2</sup>Parsaulian Siregar dan <sup>3</sup>Estiyanti Ekawati

<sup>1</sup>Program Studi Teknik Fisika, Institut Teknologi Bandung

<sup>2,3</sup>Fakultas Teknologi Industri, Institut Teknologi Bandung

Jl. Ganesa No.10 Bandung 40132

<sup>1</sup>andreasjsitb09@yahoo.com

<sup>2</sup>psiregar@tf.itb.ac.id

<sup>3</sup>esti@instrument.itb.ac.id

### Abstrak

Pengukuran adalah suatu kegiatan esensial dalam dunia industri. Agar proses dapat berlangsung secara efektif dan efisien, harus dipastikan bahwa pengukuran yang dilakukan akurat. Galat yang mungkin muncul dalam pengukuran ini antara lain galat acak dan galat kasar (*gross error*). Galat inilah yang harus diminimalkan agar proses dapat berlangsung secara optimal. Pada tulisan ini, tahap pertama yang dilakukan adalah pemodelan sistem yang akan digunakan untuk studi rekonsiliasi data dan deteksi *gross error*. Sistem yang digunakan berupa tangki ganda yang terhubung di bagian tengah. Pada model yang digunakan, diasumsikan sistem merupakan sistem *linear time-invariant*.

Selanjutnya, dilakukan identifikasi sistem pada model yang sudah dibuat dengan menyesuaikan dengan suatu model ARX, dengan diasumsikan terdapat *noise* pada pengukuran yang dilakukan. Identifikasi dilakukan berdasarkan data masukan dan keluaran sistem. Lalu, dirancang algoritma filter Kalman yang akan digunakan pada model di atas. Selanjutnya ditambahkan algoritma untuk deteksi *gross error*. Algoritma *gross error* ini memanfaatkan bagian dari filter Kalman, yaitu *innovations* untuk menyusun suatu statistik uji. Selanjutnya, digunakan *hypotesis testing* untuk menentukan ada atau tidaknya *gross error* pada sistem. Algoritma deteksi *gross error* digunakan untuk mendeteksi adanya perubahan pada plant dan untuk mengetahui model yang paling sesuai dengan plant dari beberapa model yang tersedia.

**Keywords:** *gross error, hypotesis testing, filter Kalman, simulasi dinamik, sistem tangki ganda, ARX*

### 1 Pendahuluan

Pada setiap proses pengukuran, tentu selalu terjadi perbedaan antara data yang terukur dengan nilai yang sebenarnya, hal ini biasa disebut kesalahan. Secara umum, ada dua jenis kesalahan yaitu *random error* dan *gross error*. *Random error* merupakan kesalahan yang tidak dapat diprediksi baik besar maupun tandanya. Kesalahan jenis ini tidak dapat dihilangkan dan akan selalu ada dalam setiap proses pengukuran. Sedangkan *gross error* adalah kesalahan yang disebabkan oleh kejadian tertentu, bukan kejadian acak, dan cenderung dapat diatasi. *Gross error* ini biasa terjadi akibat misalnya kurang baiknya kondisi instrumen atau kesalahan kalibrasi. Selain itu, *gross error* ini juga dapat terjadi apabila terdapat ketidaksempurnaan pada proses, seperti adanya material yang hilang karena kebocoran, atau hilangnya energi karena insulasi yang kurang baik.

Kesalahan pada pengukuran ini secara umum akan memperburuk performa dari sebuah *plant*. Apabila kesalahan-kesalahan yang terjadi dibiarkan tanpa penanganan, kemungkinan besar proses akan berjalan dengan tidak optimal sehingga akan terjadi inefisiensi, atau pada kondisi yang lebih buruk, proses akan bekerja pada kondisi tidak aman dan memungkinkan terjadi kecelakaan. Maka, sangatlah penting untuk melakukan berbagai cara untuk menangani kesalahan-kesalahan pengukuran yang mungkin terjadi.

Metode yang digunakan untuk deteksi *gross error* ini pun tidak eksklusif hanya bisa digunakan untuk mendeteksi adanya suatu bias pada pengukuran. Secara umum, metode ini juga bisa digunakan untuk mendeteksi adanya perubahan yang tidak diinginkan pada *plant* dan membandingkan kesesuaian suatu *plant* terhadap beberapa model yang berbeda.

## 2 Teori Dasar

### 2.1 Persamaan Ruang Keadaan *Linear Time Invariant* Kontinu [1]

Pada analisis ruang keadaan, untuk mendeskripsikan suatu model, biasanya digunakan sekumpulan persamaan diferensial orde 1. Pada sistem *linear time invariant*, persamaan ruang keadaan dapat dituliskan sebagai berikut

$$\dot{\mathbf{x}}(t) = \mathbf{Ax}(t) + \mathbf{Bu}(t) \quad (1)$$

$$\mathbf{y}(t) = \mathbf{Cx}(t) + \mathbf{Du}(t) \quad (2)$$

Persamaan (1) adalah persamaan keadaan sistem, sedangkan persamaan (2) adalah persamaan keluaran sistem.

### 2.2 Diskritisasi Persamaan Ruang Keadaan Kontinu [2]

Pada pengendalian secara digital suatu *plant*, diperlukan konversi persamaan ruang keadaan kontinu menjadi persamaan ruang keadaan diskrit. Konversi ini dapat dilakukan dengan mencacah respon dalam sistem kontinu dalam kecepatan tertentu. Kesalahan yang diperoleh akibat diskritisasi dapat dibuat kecil dengan membuat waktu cacah cukup kecil dibandingkan dengan konstanta waktu sistem, biasanya lebih kecil dari 1/10 konstanta waktu.

Persamaan ruang keadaan diskrit akan dideskripsikan dengan bentuk persamaan seperti berikut

$$\mathbf{x}_k = \mathbf{Fx}_{k-1} + \mathbf{Gu}_{k-1} + \mathbf{w}_k \quad (3)$$

$$\mathbf{y}_k = \mathbf{Hx}_k + \mathbf{v}_k \quad (4)$$

Diasumsikan gangguan,  $\mathbf{w}_k$ , dan galat acak,  $\mathbf{v}_k$ , terdistribusi normal dengan properti statistik seperti ditunjukkan pada persamaan (5) sampai (9).

$$E[\mathbf{w}_k] = E[\mathbf{v}_k] = 0 \quad (5)$$

$$\text{Cov}[\mathbf{w}_k] = \mathbf{Q} \quad (6)$$

$$\text{Cov}[\mathbf{v}_k] = \mathbf{R} \quad (7)$$

$$\text{Cov}[\mathbf{w}_k, \mathbf{w}_j] = \text{Cov}[\mathbf{v}_k, \mathbf{v}_j] = 0 \quad j \neq k \quad (8)$$

$$\text{Cov}[\mathbf{w}_k, \mathbf{v}_j] = 0 \quad (9)$$

### 2.3 Model Auto Regressive with Exogenous Input (ARX) [3]

Model *Auto Regressive* adalah sebuah representasi dari suatu proses acak. Model ini mewakili berbagai proses yang berubah terhadap waktu. Pada model *auto regressive*, variabel keluaran bergantung secara linear pada nilai-nilai sebelumnya.

Bentuk standar model *auto regressive* ini dapat dituliskan seperti berikut.

$$\tilde{\mathbf{A}}\mathbf{y}_k = \xi_k \quad (10)$$

$\xi_k$  adalah komponen yang menunjukkan adanya ketidakpastian pada hubungan di atas. Ketidakpastian ini diasumsikan terdistribusi normal dengan rata-rata nol dan suatu varians  $\sigma^2$ .  $\tilde{\mathbf{A}}$  adalah suatu polinomial dalam operator pergeseran waktu. Biasanya pada model ini dituliskan sebagai  $q^{-1}$ .

Pada model di persamaan (10), keluaran saat ini hanya dipengaruhi dari nilai keluaran-keluaran sebelumnya serta ketidakpastian. Pada konteks kendali, biasanya terdapat aksi yang dilakukan untuk mempengaruhi perilaku proses. Untuk memasukkan efek masukan pada sistem, model AR di atas dapat diekspansi menjadi

$$\tilde{\mathbf{A}}\mathbf{y}_k = \tilde{\mathbf{B}}\mathbf{u}_{k-n} + \xi_k \quad (11)$$

$\tilde{\mathbf{A}}$  dan  $\tilde{\mathbf{B}}$  adalah polinomial dalam operator pergeseran waktu, dan  $\mathbf{u}$  adalah masukan yang dimanipulasi untuk memperoleh perilaku sistem yang diinginkan.

### 2.4 Galat Acak dan Gross Error [4]

Pasti terdapat galat pada pengukuran pada proses manapun. Galat ini bisa terjadi pada pengukuran, pemrosesan, dan transmisi sinyal. Galat total pada pengukuran, yaitu perbedaan antara nilai yang terukur dan nilai sebenarnya dari suatu variabel, bisa dituliskan sebagai penjumlahan dua tipe galat, galat acak dan *gross error*.

Kata galat acak menunjukkan bahwa baik besar maupun tanda dari galat ini tidak dapat diprediksi dengan suatu kepastian. Dengan kata lain, apabila pengukuran diulangi dengan instrumen yang sama dan dengan kondisi proses yang sama, nilai yang diperoleh bisa berbeda tergantung nilai dari galat acak ini. Galat tipe ini biasanya dikarakterisasi dengan menggunakan distribusi probabilitas.

Galat ini biasanya disebabkan oleh berbagai sumber, misalnya fluktuasi suplai tenaga, jaringan transmisi, *noise* konversi sinyal, pemfilteran masukan analog, dan perubahan kondisi ambien. Karena galat ini timbul dari berbagai sumber, mereka tidak dapat sepenuhnya dihilangkan dan akan selalu ada di setiap pengukuran. Mereka biasanya merupakan komponen frekuensi tinggi dari sinyal yang diukur dan biasanya relatif kecil.

Sedangkan *gross error* bukan disebabkan oleh kejadian acak. Penyebab *gross error* antara lain instalasi peralatan pengukuran yang kurang baik, kesalahan kalibrasi, korosi, dan adanya endapan padat pada sensor. *Gross error* biasanya memiliki besar dan tanda yang tertentu. Jadi, apabila pengukuran diulangi dengan instrumen yang sama pada kondisi yang sama, kontribusi dari *gross error* pada nilai terukur akan sama.

## 2.5 Deteksi *Gross Error* [4]

Teknik statistik untuk mendeteksi *gross error* ini salah satunya berdasar pada pengujian hipotesis. Pada kasus deteksi *gross error*, terdapat 2 jenis hipotesis.

1. Hipotesis null,  $H_0$ , adalah hipotesis yang menyatakan tidak terdapat bukti signifikan tentang adanya *gross error* pada sistem.
2. Hipotesis alternatifnya,  $H_1$ , adalah hipotesis yang menyatakan ada bukti signifikan tentang adanya *gross error* pada sistem.

Semua teknik statistik untuk memilih di antara kedua hipotesis ini menggunakan statistik uji, yang merupakan fungsi dari pengukuran dan batasan model. Statistik uji ini kemudian dibandingkan dengan suatu nilai batas, kemudian keputusan mengenai keberadaan *gross error* dapat diambil. Nilai batas ini ditentukan berdasarkan suatu nilai probabilitas *false alarm* yang diijinkan.

Ada kemungkinan, algoritma pendeteksian *gross error* ini melakukan kesalahan. Secara umum, terdapat 2 jenis kesalahan:

1. Kesalahan Tipe I, atau sering juga disebut *false alarm*. Terjadi ketika terpilih hipotesis alternatif ( $H_1$ ) padahal sebetulnya tidak terdapat *gross error* pada sistem.
2. Kesalahan Tipe II, ketika sistem menyatakan tidak ada *gross error*, padahal sebetulnya ada.

## 2.6 Deteksi *Gross Error* di Sistem Dinamik Linear [4]

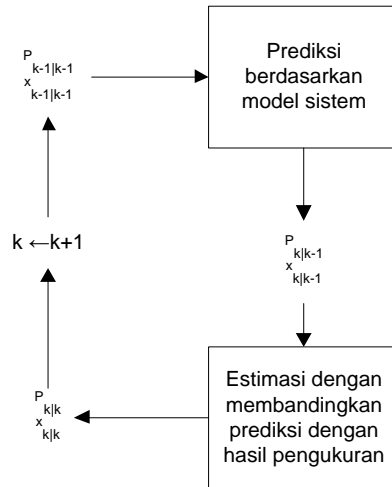
Deteksi *gross error* di sistem dinamik memanfaatkan redundansi temporal untuk memaksimalkan akurasi dari estimasi hasil pengukuran. Untuk keperluan tersebut digunakan model untuk memprediksi perubahan dari hasil pengukuran dari waktu ke waktu. Perbedaan antara prediksi tersebut dengan hasil pengukuran yang diperoleh akan digunakan sebagai dasar uji statistik yang menentukan apakah telah terjadi *gross error* atau tidak. Prediksi tersebut diperoleh dari Filter Kalman.

## 2.7 Estimasi Keadaan Optimal Dengan Menggunakan Filter Kalman [4, 5]

Filter Kalman dapat melakukan estimasi terhadap variabel-variabel keadaan pada sistem. Masukan dari filter Kalman adalah hasil pengukuran serta masukan yang diberikan pada sistem, di mana hasil pengukuran ini biasanya mengandung *noise*. Diharapkan, setelah melewati filter Kalman, diperoleh estimasi yang nilainya lebih dekat ke nilai variabel proses sebenarnya.

Estimasi akan dituliskan dengan notasi  $\hat{x}_{j|k}$ , yang dapat diinterpretasikan sebagai estimasi variabel keadaan pada waktu  $j$  yang diperoleh dengan seluruh pengukuran dari waktu  $t = 1$  sampai  $t = k$ .

Dalam bentuk grafik, persamaan filter Kalman dapat digambarkan seperti pada Gambar 1.



Gambar 1 Skema algoritma filter Kalman

Didefinisikan  $\mathbf{P}_{k|k-1}$  dan  $\mathbf{P}_{k|k}$  pada persamaan (1) dan (2)

$$\mathbf{P}_{k|k-1} = \text{cov}(\mathbf{x}_k - \mathbf{x}_{k|k-1}) \quad (12)$$

$$\mathbf{P}_{k|k} = \text{cov}(\mathbf{x}_k - \mathbf{x}_{k|k}) \quad (13)$$

## 2.8 Properti Statistik dari *Innovations* dan *Global Test* [4]

Variabel utama yang digunakan dalam deteksi *gross error* adalah *innovations*, yang didefinisikan sebagai

$$\mathbf{v}_k = \mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \quad (14)$$

*Innovations* di sini memiliki arti fisis perbedaan antara hasil pengukuran dengan hasil prediksi filter Kalman

Pada hipotesis nol, di mana tidak ada *gross error* pada pengukuran dari waktu awal sampai waktu  $k$ , *innovations* terdistribusi normal dengan ekspektasi nilai dan matriks kovariansi diberikan dengan persamaan (15) sampai (17).

$$\mathbf{E}[\mathbf{v}_k] = \mathbf{0} \quad (15)$$

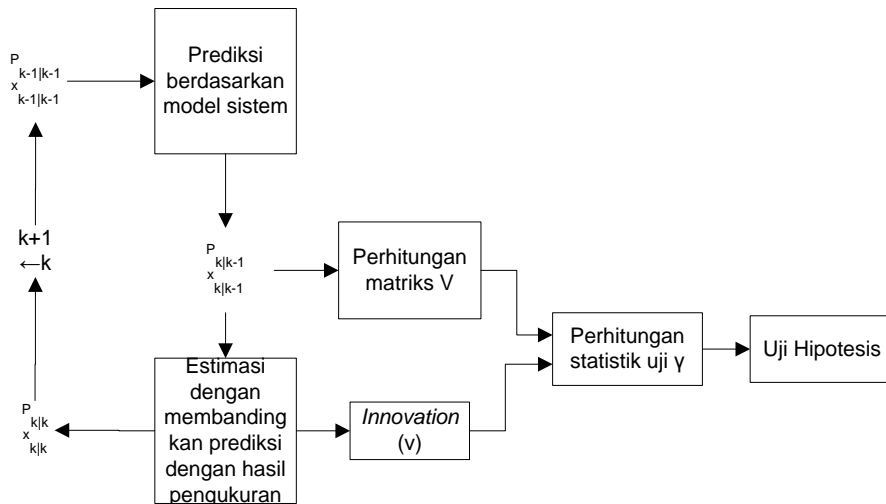
$$\text{Cov}[\mathbf{v}_k] = \mathbf{V}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \quad (16)$$

$$\text{Cov}[\mathbf{v}_k, \mathbf{v}_j] = \mathbf{0} \quad j \neq k \quad (17)$$

Dari properti statistik di atas, dapat dibuat sebuah statistik uji untuk menentukan terjadinya *gross error*. Statistik ujinya didefinisikan pada persamaan (18).

$$\gamma = v_k^T V_k^{-1} v_k \quad (18)$$

Prosedur pengambilan keputusan berdasarkan statistik uji diberikan pada Gambar 2.



**Gambar 2 Skema algoritma filter Kalman ditambah perhitungan statistik uji *gross error detection***

$\gamma$  mengikuti distribusi chi-square dengan  $n$  derajat kebebasan, dengan  $n$  adalah jumlah dari pengukuran. Batas kriteria uji ditentukan dari prosentase *false alarm* ( $\alpha$ ) yang diijinkan. Nilai  $\alpha$  bergantung dari seberapa sensitif algoritma deteksi *gross error* yang diinginkan. Secara umum,  $\alpha$  biasa diberi nilai 5%. Selanjutnya, cara pengambilan keputusan adalah:

1. Jika  $\gamma < \chi^2_{1-\alpha, n}$ ,  $H_0$  diterima, tidak terdapat *gross error* pada sistem.
2. Jika  $\gamma > \chi^2_{1-\alpha, n}$ ,  $H_1$  diterima, terdapat *gross error* pada sistem.

4.

Uji ini dapat diaplikasikan pada setiap *sampling* untuk mendeteksi apakah telah terjadi *gross error*. Apabila uji ini menolak hipotesis nol pada suatu waktu, misal  $t_0$ , maka dapat disimpulkan *gross error* telah terjadi pada waktu  $t_0$ .

Selain itu, beberapa *innovations* yang diperoleh pada suatu waktu, misalnya  $N$  waktu *sampling*, dapat digunakan bersama-sama. Statistik *global test* pada kasus ini diberikan dengan persamaan

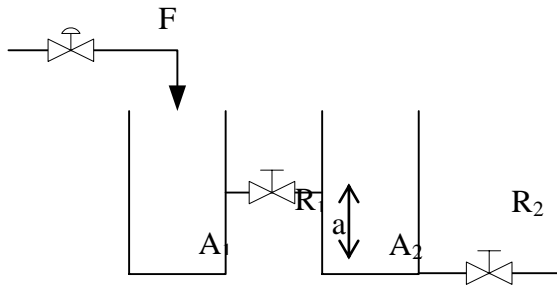
$$\bar{\gamma} = \sum_{i=k}^{k+N} v_i^T V_i^{-1} v_i \quad (19)$$

Hasil akumulasi  $N$  statistik uji ini, pada hipotesis nol, mengikuti distribusi chi-square dengan  $Nn$  derajat kebebasan. Untuk level signifikansi tertentu, kriteria uji yang dipilih untuk mendeteksi apakah terdapat *gross error* adalah  $\chi^2_{1-\alpha, Nn}$

### 3 Pemodelan Sistem

#### 3.1 Pemodelan Sistem Kontinu

Pertama-tama, perlu dibuat terlebih dahulu sistem untuk mengujicoba metode pendeteksian *gross error* ini. Skematik dari sistem yang akan dipakai ditunjukkan pada Gambar 3.



**Gambar 3 Skematik sistem tangki ganda untuk uji coba metode**

Tangki pada gambar 3 memiliki luas penampang masing-masing  $A_1$  dan  $A_2$ . Tinggi air pada masing-masing tangki akan diberi variabel  $h_1$  dan  $h_2$ . Dalam model tangki sederhana di atas, terdapat tiga jenis dinamika:

1. Saat  $h_1 < a$  dan  $h_2 < a$
2. Saat  $h_1 > a$  dan  $h_2 < a$
3. Saat  $h_1 > a$  dan  $h_2 > a$

Dalam pemodelan ini diasumsikan nilai resistansi valve konstan, tidak bergantung pada ketinggian air

1. Model sistem saat  $h_1 < a$  dan  $h_2 < a$

$$\begin{bmatrix} \dot{h}_1 \\ \dot{h}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{A_2 R_2} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{A_1} \\ 0 \end{bmatrix} F \quad (20)$$

2. Model sistem saat  $h_1 > a$  dan  $h_2 < a$

$$\begin{bmatrix} \dot{h}_1 \\ \dot{h}_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{A_1 R_1} & 0 \\ \frac{1}{A_2 R_1} & -\frac{1}{A_2 R_2} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{A_1} & \frac{1}{A_1 R_1} \\ 0 & -\frac{1}{A_2 R_1} \end{bmatrix} \begin{bmatrix} F \\ a \end{bmatrix} \quad (21)$$

3. Model sistem saat  $h_1 > a$  dan  $h_2 > a$

$$\begin{bmatrix} \dot{h}_1 \\ \dot{h}_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{A_1 R_1} & \frac{1}{A_1 R_1} \\ \frac{1}{A_2 R_1} & -\frac{1}{A_2 R_1} - \frac{1}{A_2 R_2} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{A_1} \\ 0 \end{bmatrix} F \quad (22)$$

### 3.2 Diskritisasi Pemodelan Sistem dan Simulasi

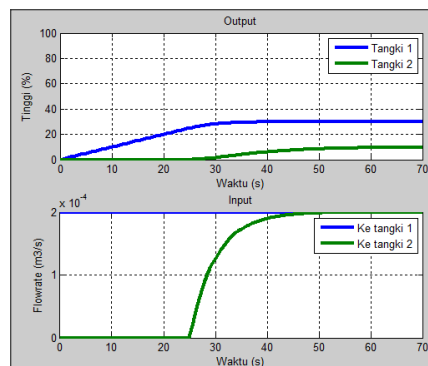
Diskritisasi pemodelan sistem dilakukan dengan perangkat lunak MATLAB. Untuk meminimalkan kesalahan yang muncul akibat diskritisasi, dipilih waktu cacah yang cukup kecil yaitu  $1/20$  dari konstanta waktu total proses, yaitu  $R_1A_1 + R_2A_2$ .

Di bawah ini akan ditunjukkan keluaran sistem, dalam hal ini ketinggian air pada tangki 1 dan tangki 2 ( $h_1$  dan  $h_2$ ), apabila diberi masukan (laju aliran air ke tangki 1) konstan. Parameter tangki diberikan nilai seperti berikut:

$A_1 = 0,01 \text{ m}^2$ ;  $A_2 = 0,01 \text{ m}^2$ ;  $R_1 = 500 \text{ s/m}^2$ ;  $R_2 = 1000 \text{ s/m}^2$ ;  $a = 0,5 \text{ m}$ .

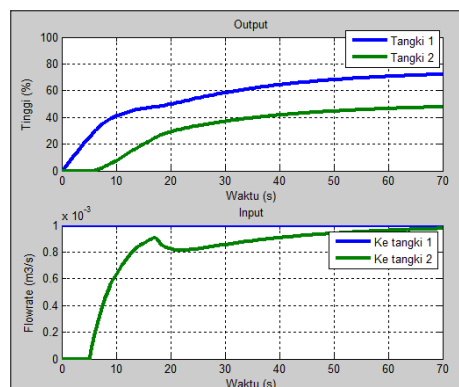
Tinggi maksimum permukaan air (tinggi tangki), ditunjukkan dengan nilai 100%, merujuk pada tinggi permukaan air 2m.

Pada simulasi di Gambar 4 diberikan nilai  $F = 0,0002 \text{ m}^3/\text{s}$ , sehingga pada kondisi tunak sistem akan berada pada kondisi 2.



Gambar 4 Grafik respon ketinggian air saat diberi masukan berupa laju aliran  $0,2 \text{ m}^3/\text{s}$ .

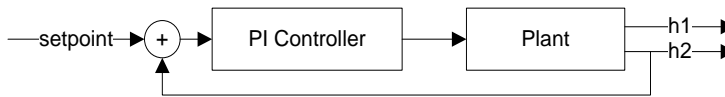
Di bawah ini adalah simulasi tangki dengan diberikan laju aliran masuk ke tangki 1 sebesar  $0,001 \text{ m}^3/\text{s}$ . Pada kondisi tunak, sistem akan berada pada kondisi 3.



Gambar 5 Grafik respon ketinggian air saat diberi masukan berupa laju aliran  $0,001 \text{ m}^3/\text{s}$ .

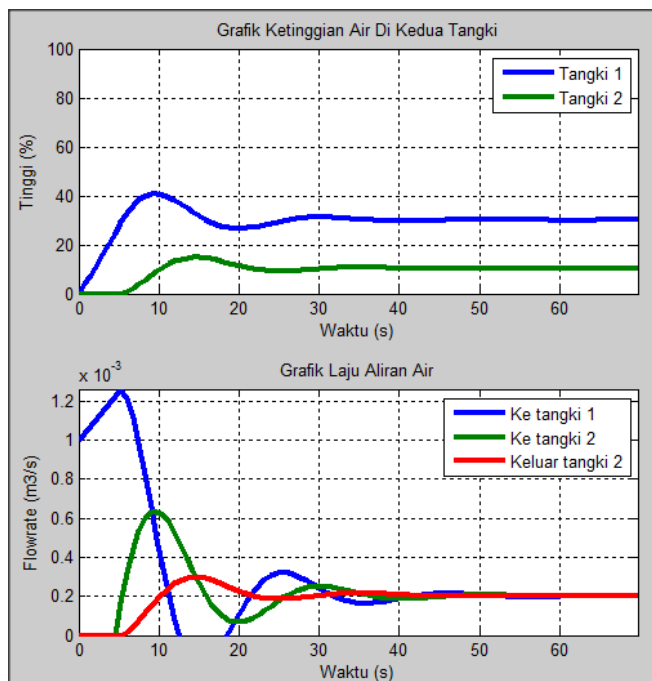


Untuk keperluan berikutnya, akan dipasang pengendali pada sistem ini untuk mengendalikan ketinggian pada tangki 2. Secara skematik, sistem keseluruhan dapat digambarkan seperti pada Gambar 6.



**Gambar 6 Skematik sederhana pengontrolan ketinggian air pada tangki 2**

Simulasi pada Gambar 7 adalah simulasi dengan setpoint = 0,2m (10%). Diberikan nilai parameter pengontrol  $K_p = 0,005$  dan  $T_i = 20$ .

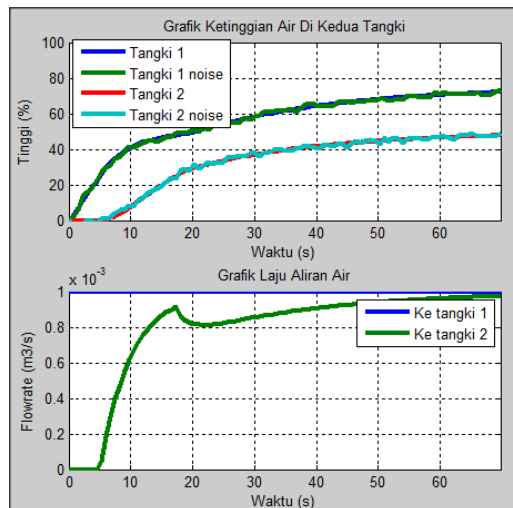


**Gambar 7 Tinggi air pada tangki 1 dan tangki 2 setelah diberikan pengontrol untuk mengendalikan ketinggian air tangki 2 (setpoint = 0,2m)**

## 4 Simulasi Penggunaan Metode Deteksi *Gross Error*

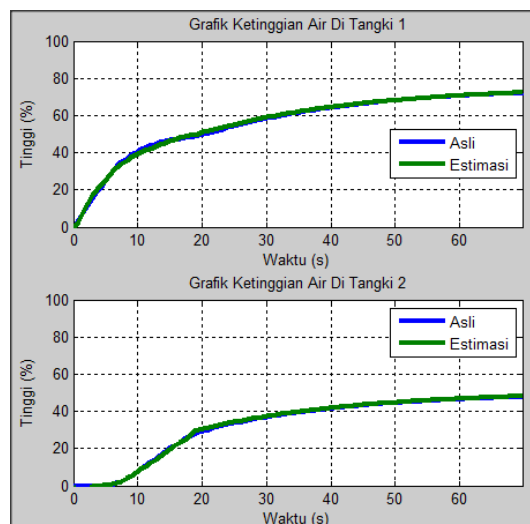
### 4.1 Identifikasi Sistem Dengan Metode ARX

Agar lebih realistis, di sini data pengukuran diberi sedikit *noise* (memiliki distribusi normal dengan rata-rata nol dan standar deviasi 1% tinggi maksimum permukaan air). Laju aliran yang masuk ke tangki 1 diberi nilai cukup besar, yaitu  $0,001 \text{ m}^3/\text{s}$ , agar seluruh dinamika dari tangki dapat diperoleh. Gambar 8 menunjukkan hasil simulasi sistem ditambah dengan *noise*.



**Gambar 8 Grafik ketinggian air di tangki 1 maupun 2 serta hasil pengukurannya yang telah ditambah *noise***

Perbandingan antara *plant* dengan model hasil estimasi, dengan laju aliran air ke tangki 1  $0,001 \text{ m}^3/\text{s}$  (sama dengan masukan pada saat identifikasi) ditunjukkan pada Gambar 9.

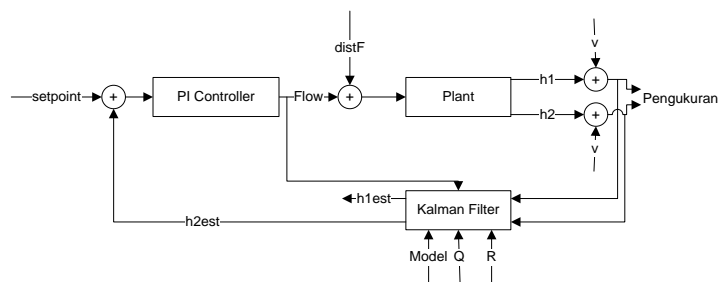


**Gambar 9 Grafik perbandingan model asli dan identifikasi ARX pada nilai laju aliran ke tangki 1 adalah  $0,001 \text{ m}^3/\text{s}$**

## 4.2 Filter Kalman

Untuk keperluan deteksi *gross error*, diperlukan algoritma filter Kalman terlebih dahulu. Pada uji coba ini, diasumsikan terdapat *noise* pengukuran dengan standar deviasi 5% dari maksimum ketinggian masing-masing tangki serta gangguan pada laju aliran ke

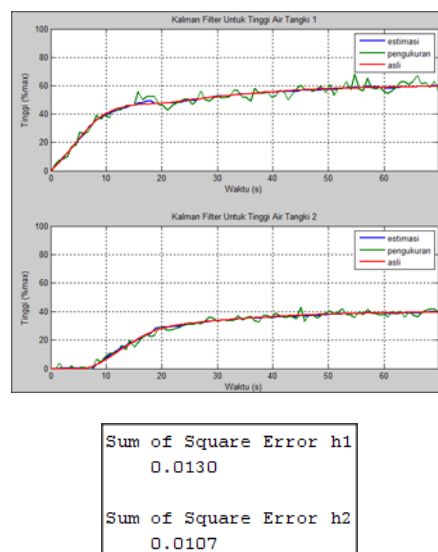
tangki 1 dengan standar deviasi 5% dari laju aliran masuk. Skematik dari sistem yang ditambahkan filter Kalman ini dapat digambarkan seperti pada Gambar 10.



Gambar 10 Skematik sistem yang ditambahkan filter Kalman

#### 4.3 Filter Kalman Dengan Model Hasil Identifikasi ARX

Pada simulasi ini, kedua pengukuran akan diberikan *noise* sebesar 5% dari nilai maksimum ketinggian masing-masing tangki. Nilai ini cukup masuk akal untuk pengukuran-pengukuran yang tidak memerlukan ketelitian sangat tinggi. Untuk pengukuran yang memerlukan ketelitian lebih tinggi, dapat dipilih instrumen yang ketelitiannya lebih baik lagi, bisa 1% atau 2%. Gambar 11 menunjukkan perbandingan nilai asli dengan hasil dari Kalman Filter.



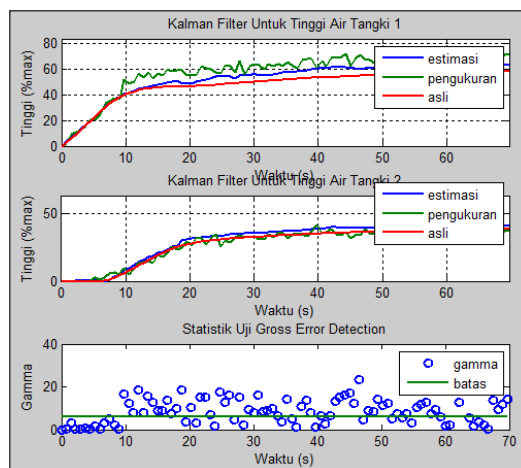
Gambar 11 Grafik estimasi filter Kalman dengan model ARX

Terlihat bahwa walaupun terdapat perbedaan antara model yang digunakan dengan model asli dari sistem, namun hasil filter Kalman masih dapat mengikuti nilai asli dengan cukup baik.

#### 4.4 Deteksi *Gross Error*

Pada simulasi pendeteksian *gross error* ini, akan diberikan bias pada kedua pengukuran dengan besar 20% dari nilai maksimum ketinggian masing-masing tangki selama simulasi.

Untuk kasus pertama, akan ditambahkan bias pengukuran sebesar 20% pada pengukuran ketinggian tangki 1. Bias ditambahkan pada  $t = 1$ . Diberikan bias 20% agar bias yang ditambahkan tidak tertutupi oleh *noise* pengukuran, di mana *noise* pengukuran ini adalah 5%. Untuk batas diputuskan terdapat *gross error*, digunakan batas pada kesalahan tipe I sebesar 5%. Gambar 12 menunjukkan hasil simulasinya.

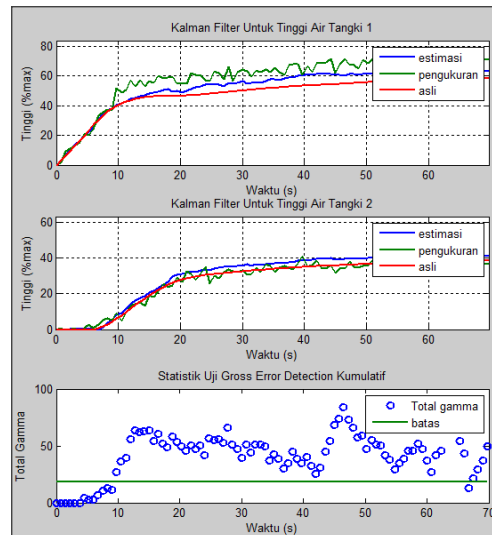


**Gambar 12** Grafik ketinggian air serta statistik uji deteksi *gross error*

Dari hasil di atas terlihat bahwa statistik uji *gross error* nilainya mulai sering melewati batas hipotesis yang ditentukan pada saat mulai terjadi bias pengukuran.

Untuk memperkecil kemungkinan kesalahan, dapat dijumlahkan nilai statistik uji di atas untuk sejumlah waktu sampel. Setelah itu, dibandingkan antara jumlah statistik uji ini dengan batas pengujian statistik uji yang baru, dalam hal ini mengacu pada distribusi chi square dengan derajat kebebasan jumlah variabel terukur dikali jumlah pengukuran yang dijumlahkan.

Pada Gambar 13 adalah hasil statistik uji kumulatif. Dalam kasus ini, lima buah pengukuran dijumlahkan terlebih dahulu sebelum dibandingkan dengan batas pengujian statistik uji yang baru.



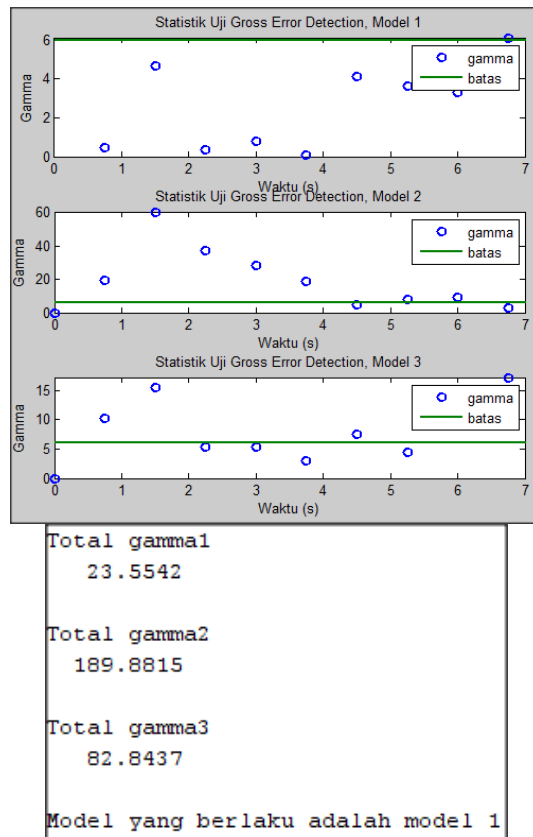
**Gambar 13. Grafik ketinggian air serta statistik uji deteksi *gross error* kumulatif**

Dari simulasi ini, dapat dilihat bahwa tidak terjadi kesalahan baik tipe I maupun tipe II pada deteksi *gross error* dengan melakukan akumulasi statistik uji gamma. Dari statistik uji kumulatif ini, dapat dibuat suatu peringatan bagi operator bahwa telah terjadi *gross error*, seperti pada Gambar 13.

#### 4.5 Deteksi Model

Algoritma deteksi *gross error* dapat juga digunakan untuk mendeteksi model yang paling sesuai dengan plant dari beberapa model yang tersedia. Pada simulasi, akan diaplikasikan deteksi *gross error* pada suatu selang waktu yang ditentukan menggunakan 3 model dari tiga dinamika yang ada pada plant di atas. Apabila model yang digunakan tepat, statistik uji akan menunjukkan nilai yang relatif kecil. Sehingga apabila ingin menebak model mana yang paling sesuai, dapat dilakukan dengan mencari nilai total statistik uji yang paling kecil pada interval berikut

Gambar 14 menunjukkan hasil simulasi saat metode diterapkan pada interval  $t=0s$  sampai  $t=7s$ . Pada interval ini berlaku model 1.



Gambar 14. Statistik uji *gross error detection* ketiga model pada interval 0s – 7s

## 5 Kesimpulan

1. Identifikasi ARX memberikan hasil yang baik karena digunakan dengan sistem linear. Kesalahan pemodelan semakin besar apabila noise pengukuran semakin besar. Agar galat rms keluaran pemodelan tidak lebih dari 1% tinggi air maksimum, maka standar deviasi noise yang digunakan tidak boleh lebih dari 1% tinggi air maksimum.
2. *Gross Error* dapat dideteksi melalui besar dari variabel *innovations* pada filter Kalman. Agar *gross error* dapat terdeteksi, diberikan bias paling tidak 3 kali nilai standar deviasi *noise*.
3. Algoritma deteksi *gross error* dapat digunakan untuk deteksi model dengan membandingkan nilai statistik uji.

## 6 Daftar Simbol

- x :vektor variabel keadaan
- u :masukan sistem

A	:matriks keadaan kontinu
B	:matriks masukan kontinu
F	:matriks keadaan diskrit
G	:matriks masukan diskrit
H	:matriks observasi
w	:gangguan sistem
v	:noise pengukuran
$K_k$	:penguat kalman
P	:matriks kovariansi vektor keadaan estimasi filter kalman
Q	:matriks kovariansi noise proses
R	:matriks kovariansi noise pengukuran
y	:hasil pengukuran
v	:innovations
V	:matriks kovariansi innovations
$\gamma$	:statistik uji deteksi gross error
$K_p$	:konstanta proporsional
$T_i$	:waktu integral
A	:luas tangki
$R_1, R_2$	:resistansi valve
$h_1, h_2$	:tinggi air di tangki 1 dan 2
F	:laju aliran masuk tangki 1

## 7 Daftar Pustaka

- [1] Ogata, Katsuhiko. 2010. *Modern Control Engineering*. New Jersey: Pearson Education
- [2] Ogata, Katsuhiko. 2010. *Discrete Time Control Engineering*. New Jersey: Pearson Education
- [3] "Dynamic Models For Controller Design," class notes Dept. of Chemical and Process Engineering, University of Newcastle upon Tyne, 1999
- [4] S. Narasimhan and C. Jordache. 1999. *Data Reconciliation and Gross Error Detection*. Oxford: Gulf Professional Publishing
- [5] Schutter, et al. 1999. *Kalman Filter: A Tutorial*. Katholieke Universiteit Leuven and Belgian Nuclear Energy Research Center