

CATATAN KULIAH
(*CLASSNOTES*)

TEKNOLOGI KENDALI
PROSES
(*Process Control*
***Technology*)**

Rhiza S. Sadjad

2016

RENCANA PEMBELAJARAN

NAMA MATAKULIAH : **TEKNOLOGI KENDALI PROSES**
KODE MATAKULIAH : **305D432 (Tahun III, Semester Awal, 2 SKS, TK)**
DOSEN-DOSEN : 1. Prof.Dr. Ir. H. Andani Ahmad, MT
2. Dr. Ir. H. Rhiza S. Sadjad, MSEE

1. DAFTAR PUSTAKA:

1. **Johnson**, Curtis D, "*Process Control Instrumentation Technology*", John Wiley and Sons , Inc., NY
2. **Smith**, Carlos A. and Armando B. **Corripio**, "*Principles and Practice of Automatic Process Control*", ", John Wiley and Sons , Inc., NY.

2. TUJUAN:

Penyajian matakuliah ini bertujuan memberi kesempatan kepada mahasiswa Program Studi Teknik Elektro – khususnya yang berminat mengambil konsentrasi Teknik Komputer, Kendali dan Elektronika (TK) – untuk mempelajari sistem kendali proses, serta menerapkan teori-teori kendali dalam analisis dan desain implementasi sistem kendali proses di dunia industri manufaktur.

3. SILABUS SINGKAT:

Perbedaan Sistem Kendali Proses dan Sistem Kendali pada umumnya, proses order pertama, *self-regulation*, *non-self-regulation*, proses dengan *dead-time*, proses order kedua dan lebih tinggi, pengendalian akhir, aspek-aspek desain Sistem Kendali Proses, algoritma pengendalian, klasifikasi sistem kendali proses, aplikasinya di industri, peralatan instrumentasi, pemodelan dan simulasi: *liquid level-control*, *liquid flow-control*, pengendali PID, kolam air hangat (*temperature+level control*).

4. KOMPETENSI UTAMA:

1. Menguasai dasar-dasar teori kendali, baik yang klasik mau pun modern, serta aplikasinya dalam analisis dan perancangan sistem kendali
2. Mampu memakai paket-paket perangkat lunak komputer untuk pemodelan dan simulasi masalah-masalah Teknik Elektro khususnya dan masalah rekayasa pada umumnya

5. KOMPETENSI PENDUKUNG:

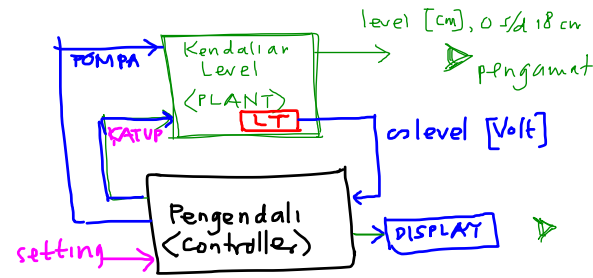
1. Mampu menggunakan bahasa asing sebagai "*second language*"
2. Mampu menggunakan bahasa-bahasa pemrograman yang umum digunakan dalam dunia engineering

6. LAINNYA:

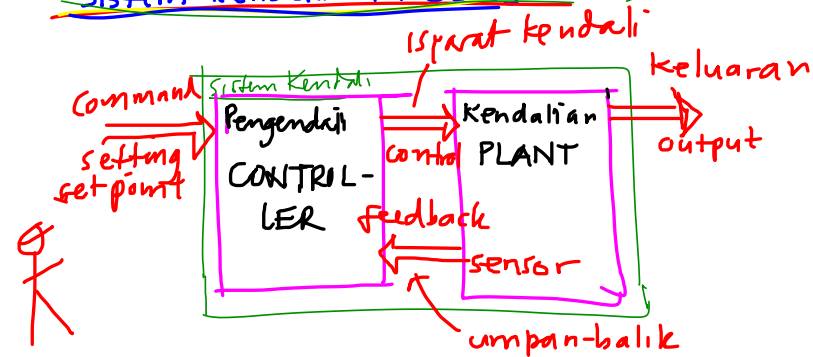
7. PEKANAN:

Pekan ke	Topik	Sub-topik
1	PENGANTAR KULIAH	Administrasi Perkuliahan
2	PENDAHULUAN	● Sistem Kendali dan Sistem Kendali Proses ● Contoh-contoh
3	DINAMIKA PROSES	● Proses Order Pertama ● <i>Self Regulation</i> dan <i>Non-Self Regulation</i>
4		● Proses Order Kedua dan lebih tinggi ● Proses dengan <i>Dead Time</i> ● Pengendalian
5	DESAIN SISTEM KENDALI PROSES	● Aspek-aspek desain Sistem Kendali Proses
6		● Elemen-elemen Sistem Kendali Proses
7		● Algoritma pengendali ● Klasifikasi dan aplikasinya, pengenalan peralatan instrumentasi
8	MIDTEST	(di kelas)
9	PEMODELAN dan SIMULASI Sistem Kendali Proses	Pengantar Pemodelan dan Simulasi
10		(Dari SOLTEQ Boiler Drum) Proyek: <i>Liquid Level and Temperature Control</i> : ● Model Kendalian tanpa Pengendali ● Verifikasi Model ● Pengendali PID
11		
12		
13		
14		
15		
16	KUNJUNGAN KE INDUSTRI di SEKITAR MAKASSAR	
	UJIAN FINAL	(sesuai jadwal)

* Project 1
GUNT Level Control
Model SIMULINK



Sistem Kendali PROSES (???)



Sistem Kendali Proses Sistem Kendali yang kendaliannya berupa PROSES (Process Plant)

"Proses" = proses di INDUSTRI manufaktur (pabrik)

PROSES FISIKA

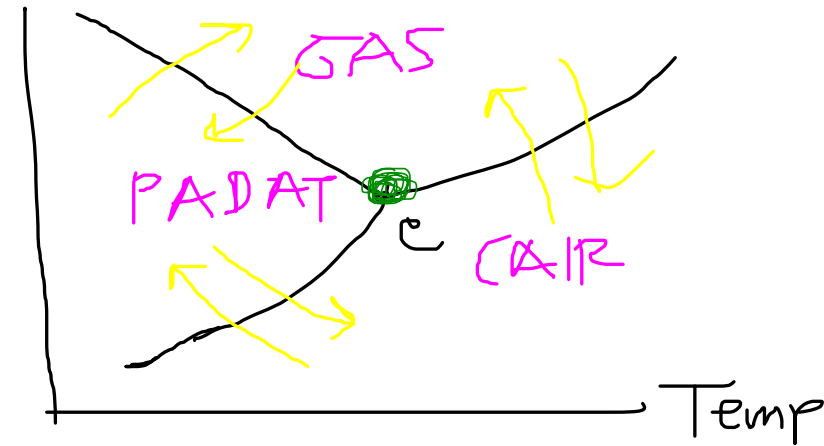
→ tidak mengakibatkan terjadinya

REAKSI KIMIA

PROSES KIMIA

→ melibatkan,
* zat / unsur berbeda
* proses fisika
* "katalis"

P (tekanan)



First Order Dynamics

$$m(t) \xrightarrow{G(s)} c(t) \quad T \frac{dc(t)}{dt} + c(t) = K_p m(t)$$

$$\text{Transfer function Laplace} \quad G(s) = \frac{K_p}{Ts + 1} = \frac{C(s)}{M(s)} \rightarrow C(s) = G(s) M(s)$$

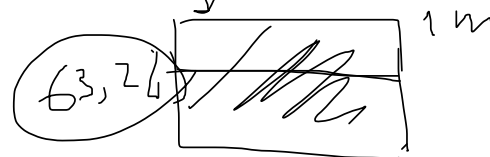
$$m(t) = \Delta m \cdot u(t) \quad M(s) = \frac{\Delta m}{s}$$

$$C(s) = \frac{K_p \Delta m}{s(Ts + 1)} = \frac{K_p \Delta m}{s} \cdot \frac{1}{Ts + 1} = \frac{K_p \Delta m}{s} \left(\frac{1}{T} \left(1 - e^{-t/T} \right) \right)$$

$$c(t) = K_p \Delta m (1 - e^{-t/T})$$

$$t = T \rightarrow c(T) = K_p \Delta m (1 - e^{-1}) = 0,6321 \cdot K_p \Delta m$$

$$t = 2T \rightarrow c(2T) = K_p \Delta m (1 - e^{-2}) = 0,8647 \cdot K_p \Delta m$$



$T = \text{konstanta waktu (time constant)}$ $[\text{second}] = \frac{[\text{Ohm}]}{[\text{Henry}]} = [\text{Ohm}] [\text{Farad}]$

"lagging"

$$T = \frac{R}{L} \text{ atau } RC$$

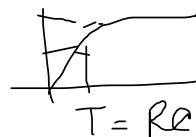
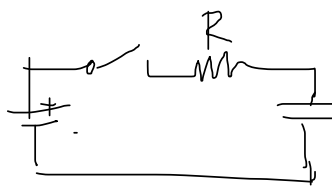
Second Order Systems

$$G(s) = \frac{K_p}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{C(s)}{M(s)}$$

$$\frac{d^2 c(t)}{dt^2} + 2\zeta\omega_n \frac{dc(t)}{dt} + \omega_n^2 c(t) = K_p m(t)$$

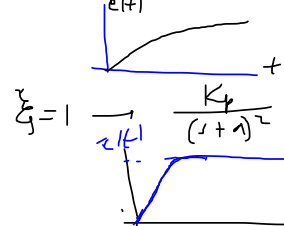
$$T = \frac{d^2 c(t)}{dt^2} + 2\zeta T \frac{dc(t)}{dt} + c(t) = K_p m(t)$$

check * hubungan antara T dan ω_n
* hubungan antara K_p dan K_p

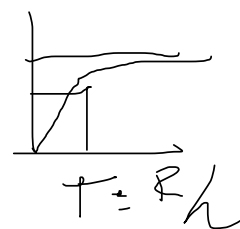
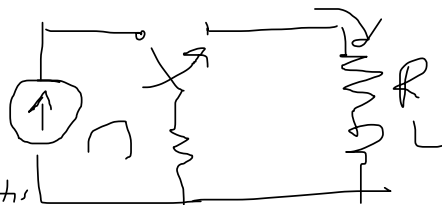


lagging

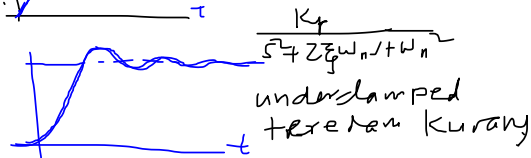
$\zeta > 1 \rightarrow \frac{K_p}{(s+a)(s+b)}$, overdamped teredam lebih



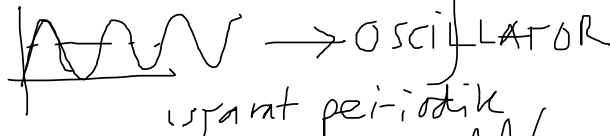
$\zeta = 1 \rightarrow \frac{K_p}{(s+a)^2}$, critically damped teredam kritis



$0 < \zeta < 1$



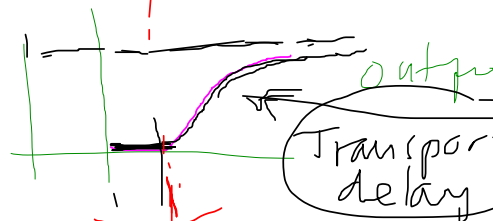
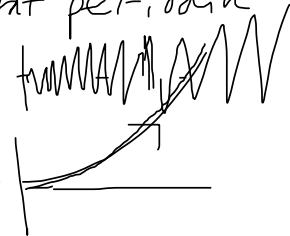
$\zeta = 0 \rightarrow \frac{K_p}{s^2 + \omega_n^2}$ oscillations



$\zeta < 0$

unstable

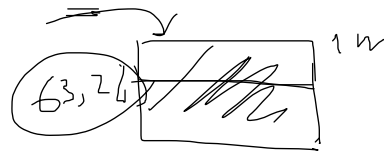
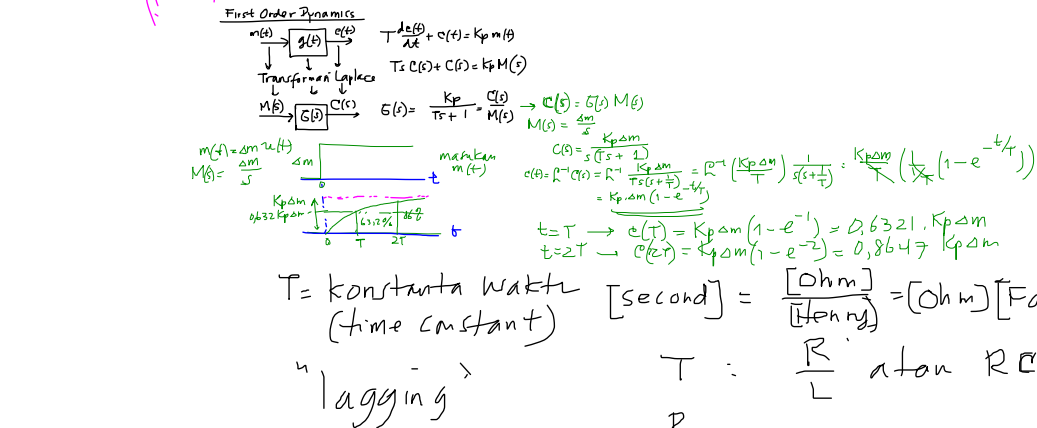
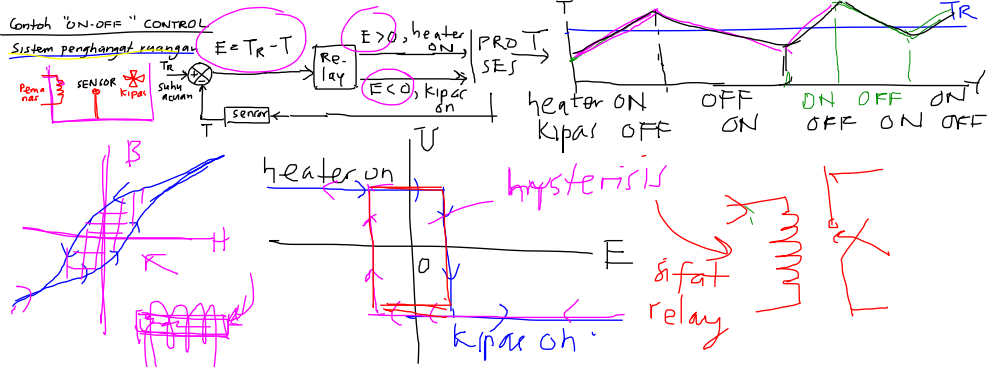
tidak bermanfaat



Dead Time

Dead Zone

lagging



Second Order Systems

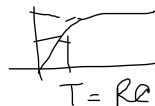
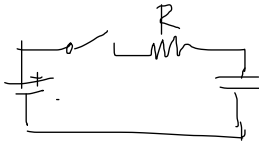
$M(s) \rightarrow G(s) = \frac{K_p}{s^2 + 2\zeta\omega_n s + \omega_n^2}$

$\zeta = \frac{c}{2\omega_n}$ (damping ratio)

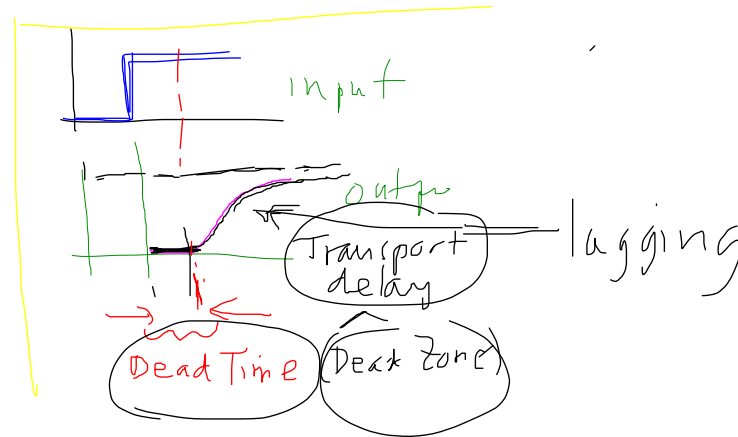
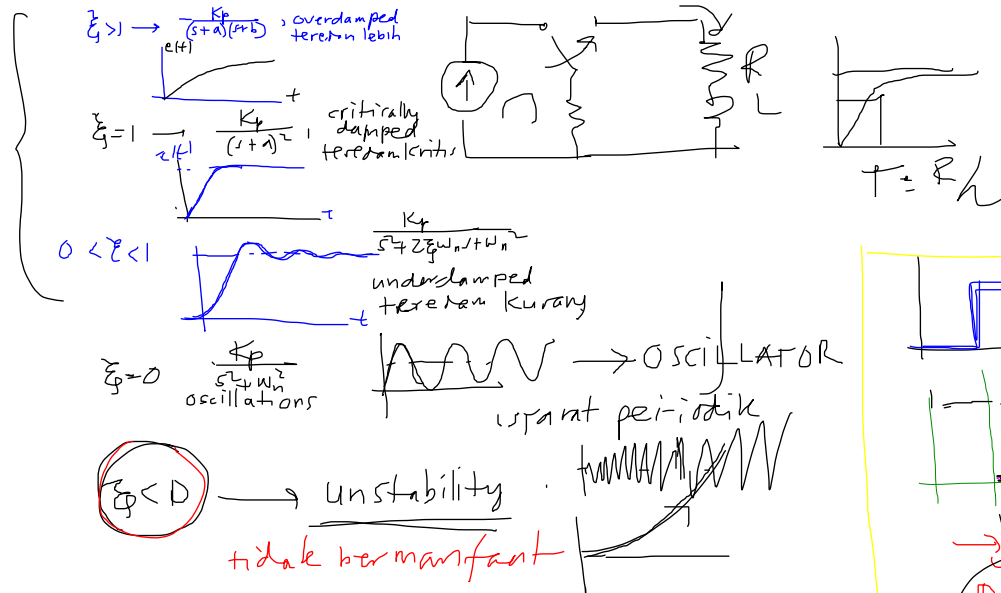
ω_n (natural frequency)

$T = \frac{1}{\omega_n}$ (time constant)

check: hubungan antara T dan ω_n



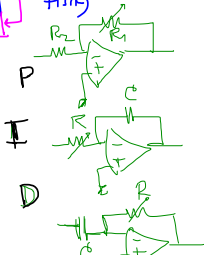
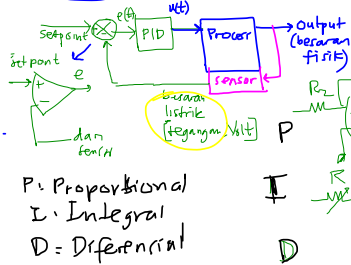
lagging



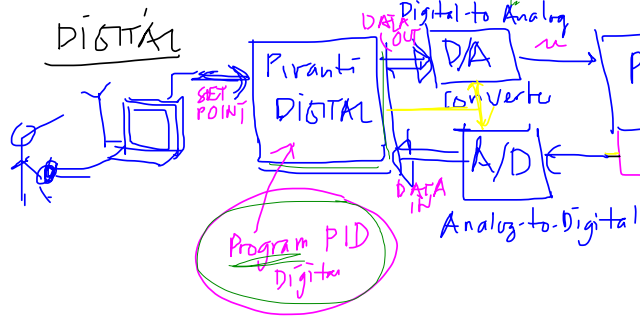
ANALOG

Tomboi

PID Controller



DIGITAL



P

$$e \rightarrow K_p \rightarrow u = K_p e$$

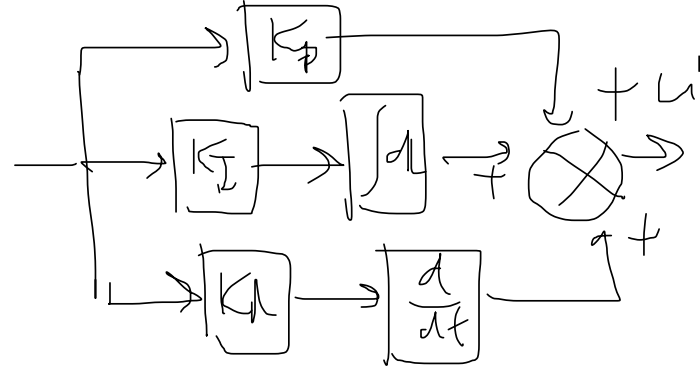
PI

$$e \rightarrow K_p \rightarrow K_p e \rightarrow u = K_p e + K_I \int e dt$$

PD

$$e \rightarrow K_p \rightarrow K_p e \rightarrow u = K_p e + K_D \frac{d e}{d t}$$

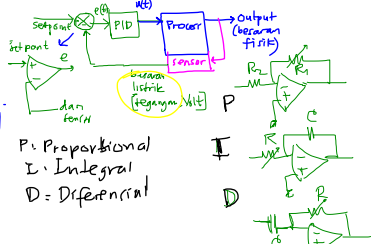
PID



ANALOG

50 Tomboi

PID Controller



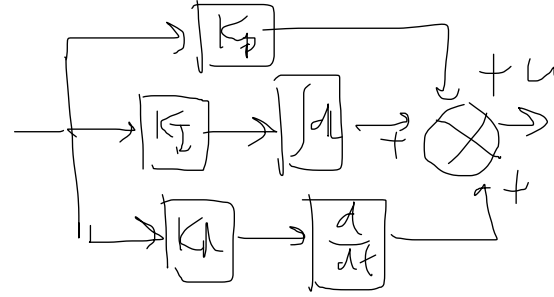
$$P \quad e \rightarrow K_p \rightarrow u = K_p e$$

$$PI \quad e \rightarrow \begin{cases} K_p e \\ K_I \int e dt \end{cases} \rightarrow u = K_p e + K_I \int e dt$$

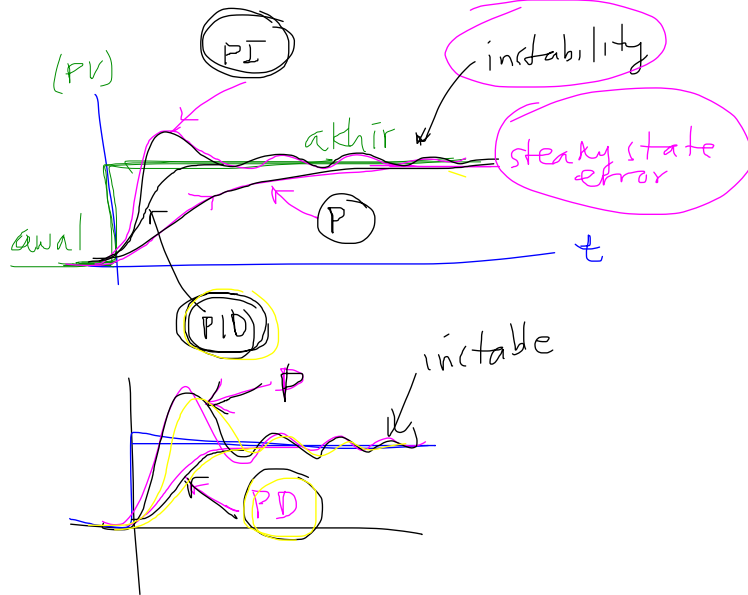
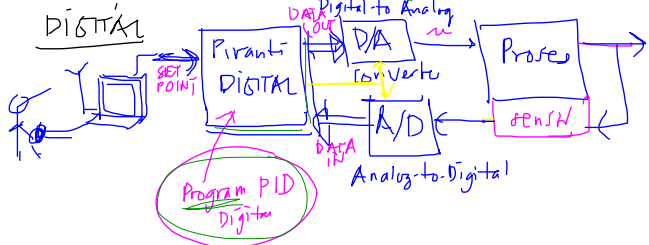
PD

$$PD \quad e \rightarrow \begin{cases} K_p e \\ K_D \frac{de}{dt} \end{cases} \rightarrow u = K_p e + K_D \frac{de}{dt}$$

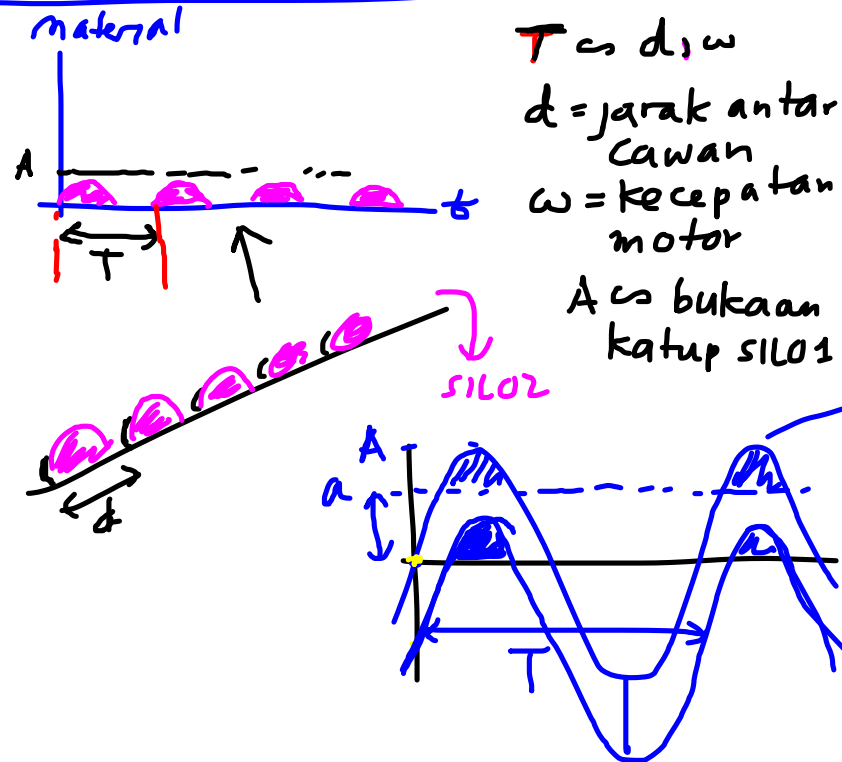
PID



DIGITAL



* MODEL SIMULINK dan CONVEYOR

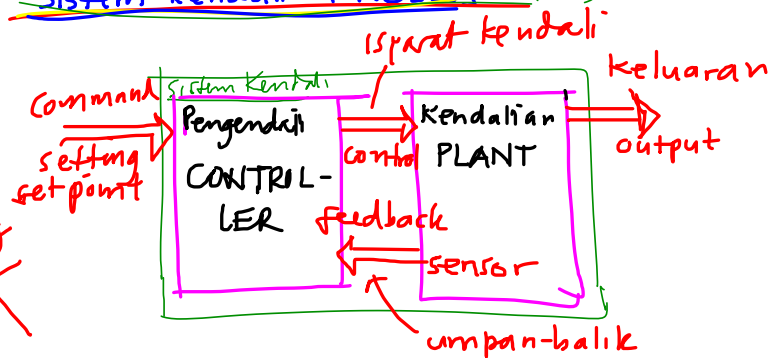


$$x'(t) = A \sin \omega t$$

$$x(t) = A \sin \omega t - a$$

if $x(t) \leq 0$
 $y(t) = 0$
 else
 $y(t) = x(t)$
 end

Sistem Kendali PROSES (???)



Sistem Kendali Proses adalah sistem kendali yang kendaliannya berupa PROSES (Process Plant)

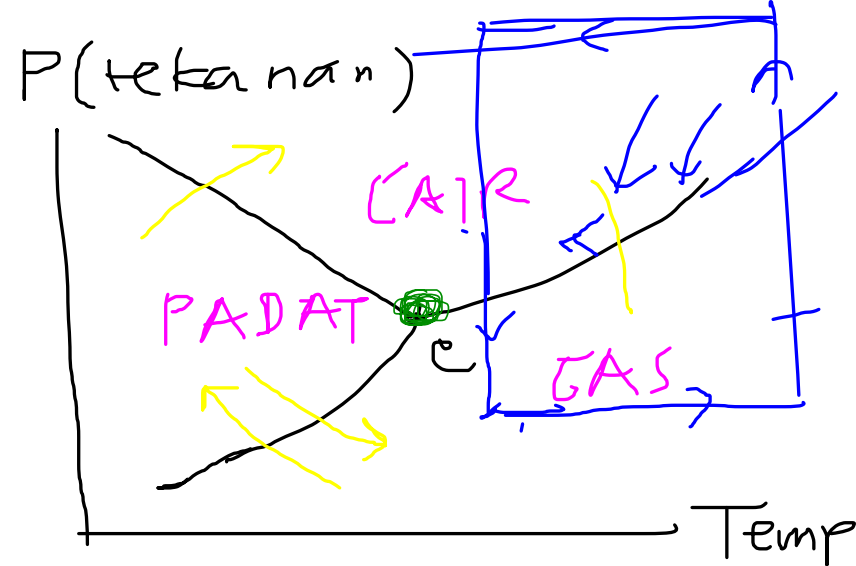
"Proses" = proses di INDUSTRI manufaktur (pabrik)

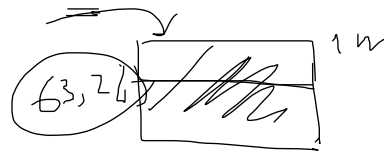
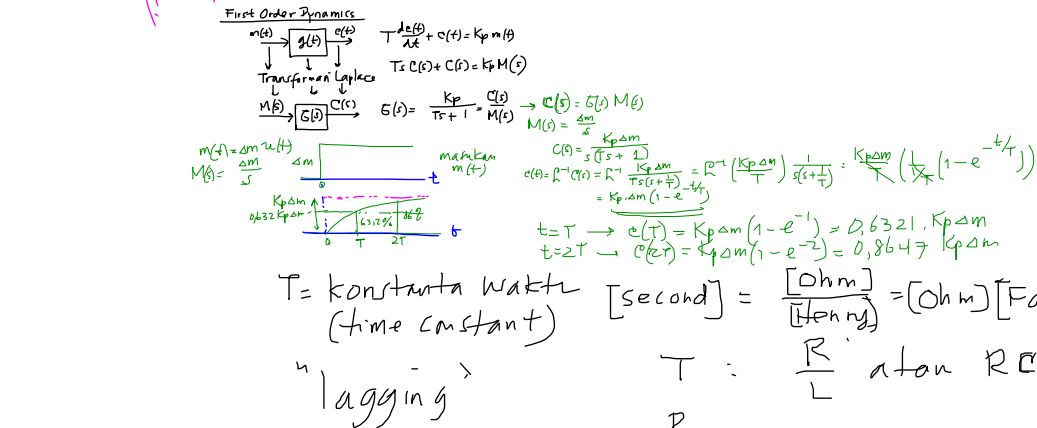
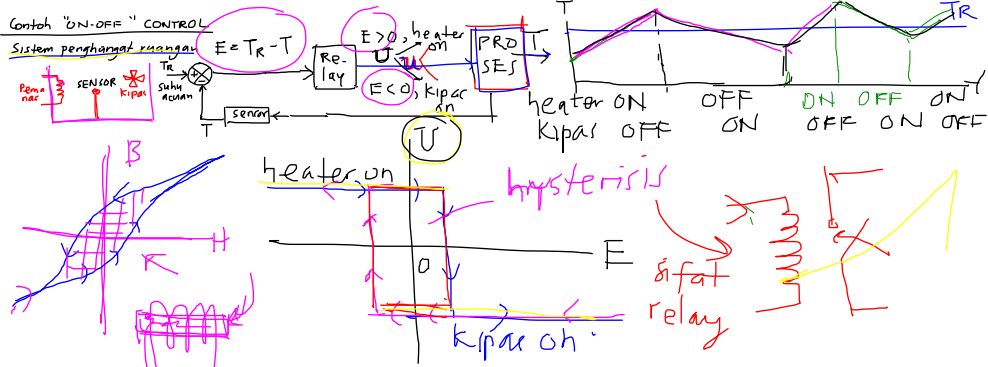
PROSES FISIKA

→ tidak mengakibatkan terjadinya REAKSI KIMIA

PROSES KIMIA

→ melibatkan,
 * zat-zat/unsur berbeda
 * proses fisika
 * "katalis"





Second Order Systems

$M(s) = \frac{C(s)}{G(s)}$

$G(s) = \frac{K_p}{s^2 + 2\zeta\omega_n s + \omega_n^2}$

$C(s) = \frac{K_p}{s^2 + 2\zeta\omega_n s + \omega_n^2}$

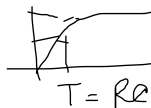
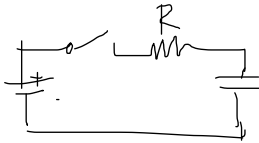
$c(t) = \frac{K_p}{\omega_d} e^{-\zeta\omega_n t} \sin(\omega_d t)$

$\omega_d = \omega_n \sqrt{1 - \zeta^2}$

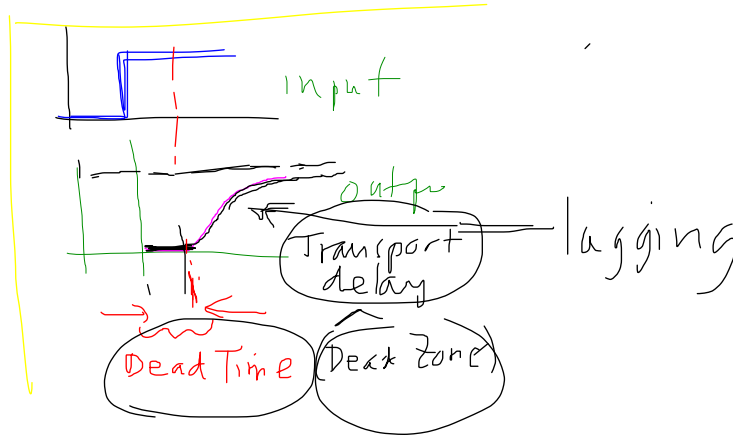
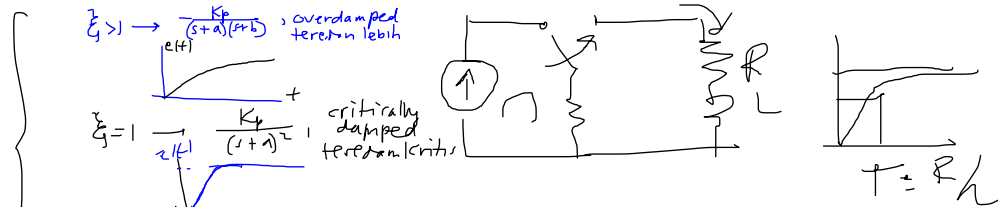
$\zeta = \text{damping ratio}$

$\omega_n = \text{natural frequency}$

check: hubungan antara T dan ω_n



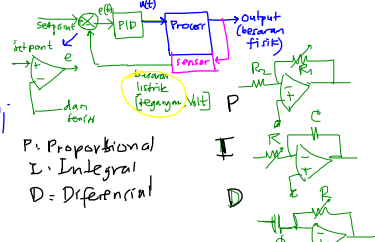
lagging



ANALOG

50 Tomboi

PID Controller



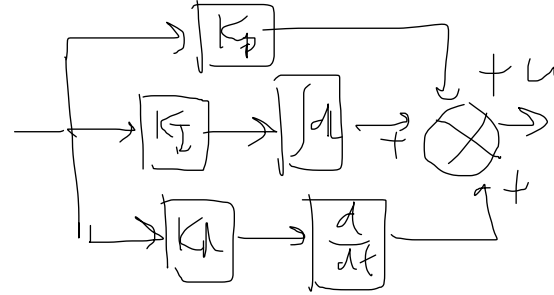
P $e \rightarrow K_p \rightarrow u = K_p e$

PI $e \rightarrow K_p \rightarrow K_p e$
 $e \rightarrow K_I \rightarrow \int e dt \rightarrow K_I \int e dt$
 $u = K_p e + K_I \int e dt$

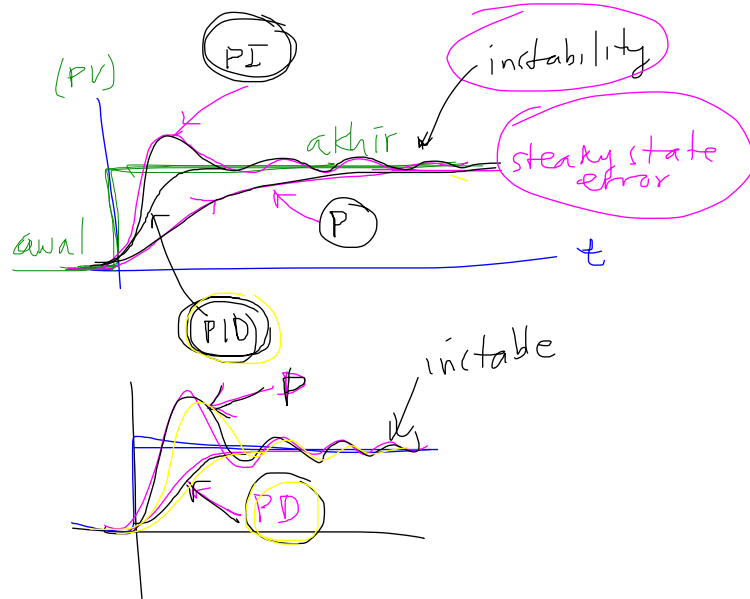
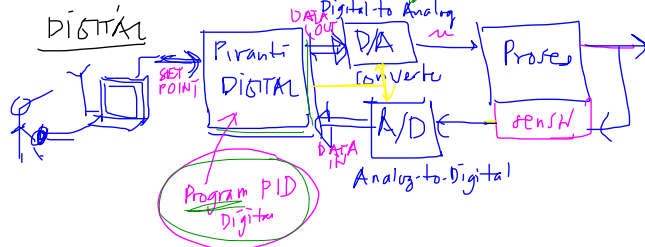
PD

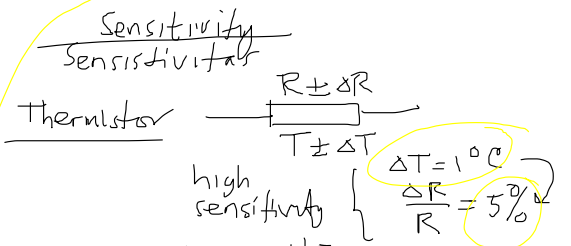
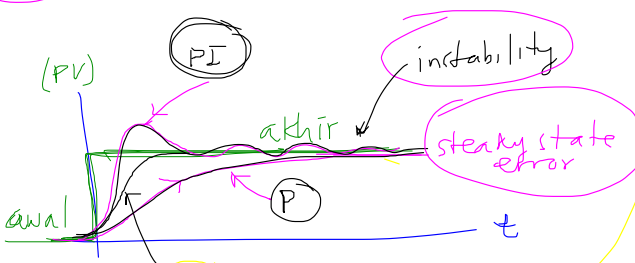
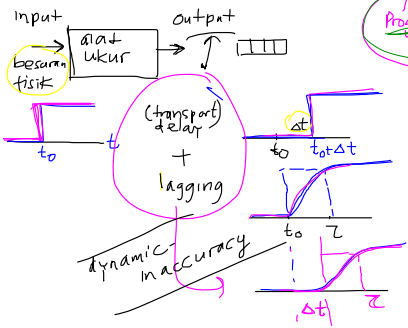
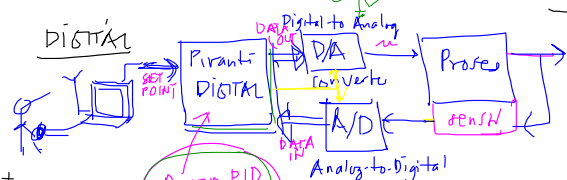
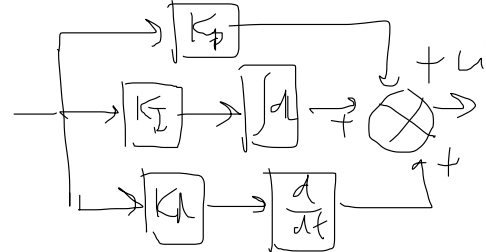
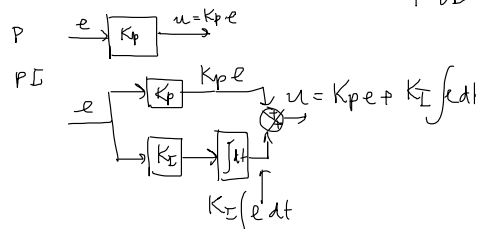
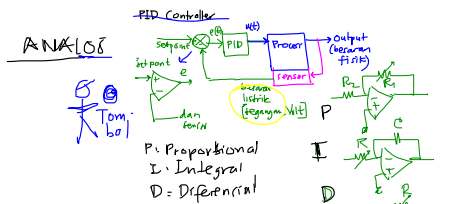
$u = K_p e + K_D \frac{de}{dt}$

PID

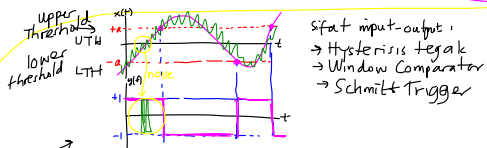
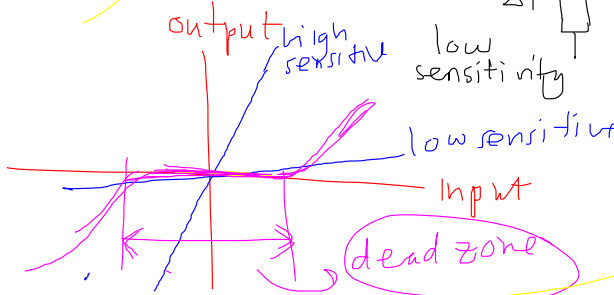
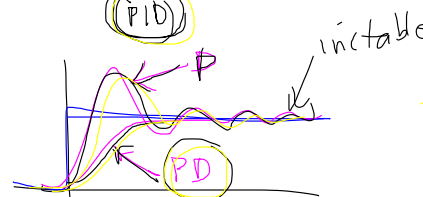
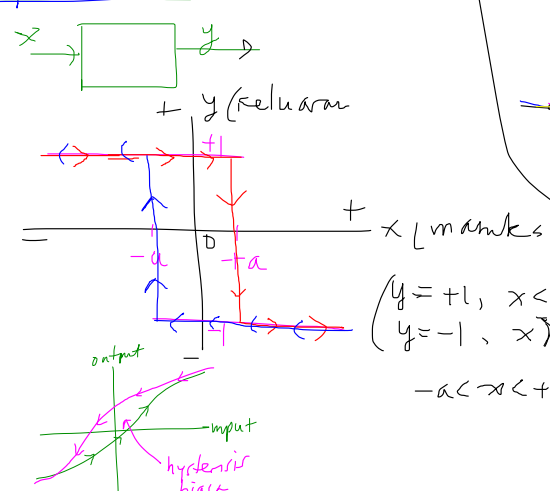


DIGITAL





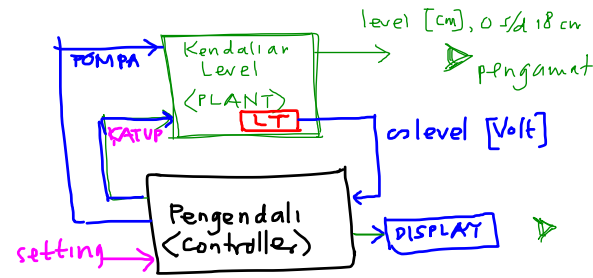
Hysteresis



stat input-output:

- Hysteresis tegak
- Window Comparator
- Schmitt Trigger

* Project 1
GUNT Level Control
Model SIMULINK



CONVEYOR

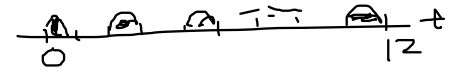
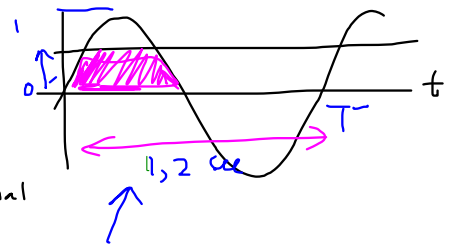
Silo 1

Silo 2

Motor dan roda gigi

Kecepatan motor 250 RPM
Roda gigi 1.5
Cawan akan menumpahkan material 50 kali per menit
 $\frac{1}{5} \times 250$

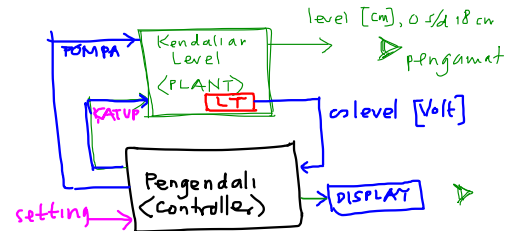
dengan asumsi 1x putaran roda gigi, 1 cawan menumpahkan material
Di atas conveyor 10 cawan, maka waktu untuk material bergerak dari silo 1 ke silo 2 adalah $\pm 10 \times \frac{60 \text{ sec}}{50} = 12 \text{ sec}$



$$\begin{aligned} T &= 1.2 \text{ sec} \\ f &= \frac{1}{T} = \frac{1}{1.2} \text{ Hz} \\ \omega &= 2\pi f \end{aligned} \quad \left. \vphantom{\begin{aligned} T &= 1.2 \text{ sec} \\ f &= \frac{1}{T} = \frac{1}{1.2} \text{ Hz} \\ \omega &= 2\pi f \end{aligned}} \right\} \sin \omega t$$

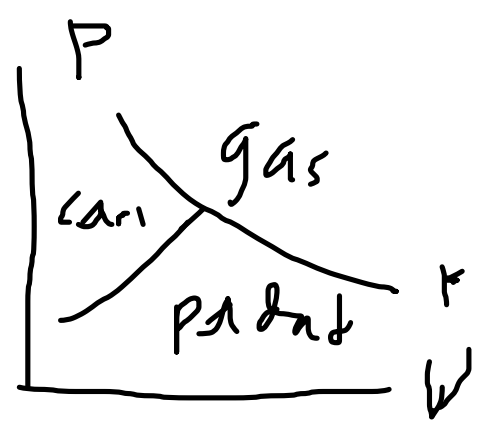
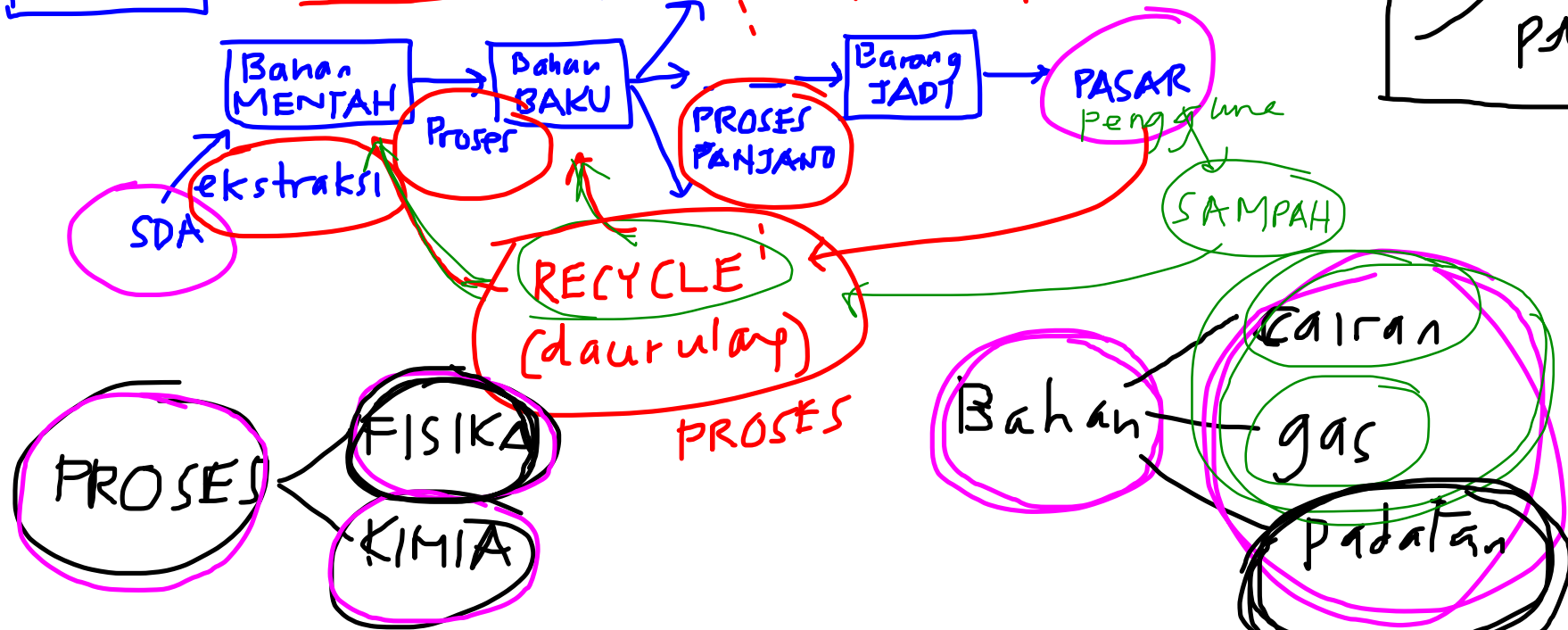
* Project 1

GUNT Level Control
Model SIMULINK



Teknologi Kendali Proses (Process Control Technology)

PROSES di industri manufaktur :

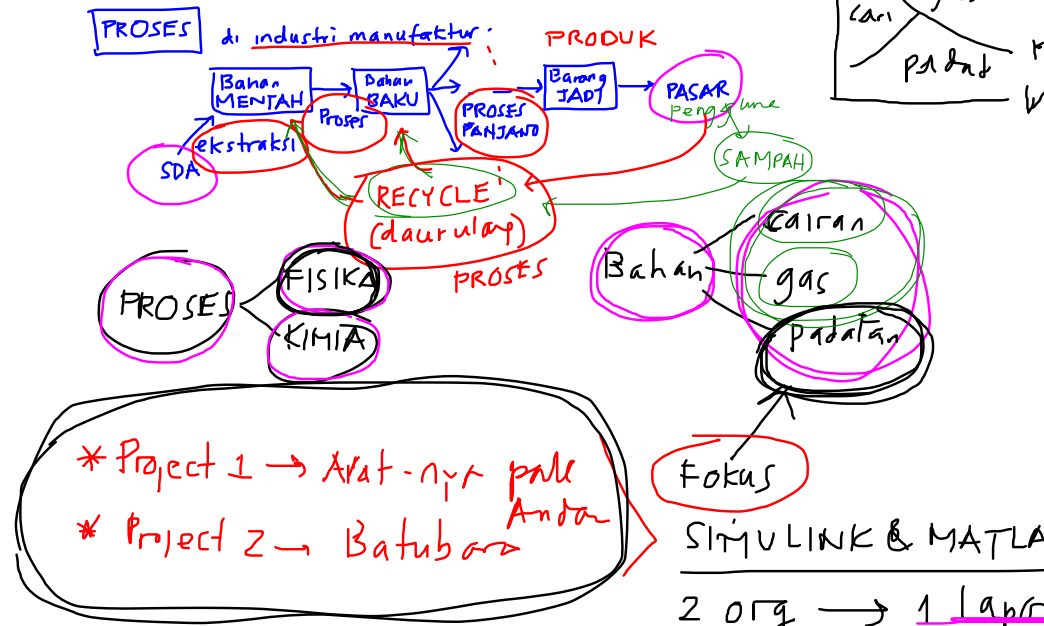


- * Project 1 → Alat - nirkabel
- * Project 2 → Batubara

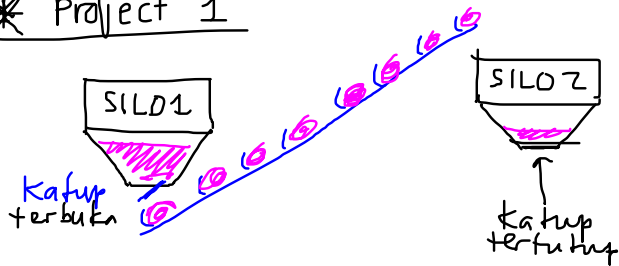
fokus

SIMULINK & MATLAB

Teknologi Kendali Proses (Process Control Technology)

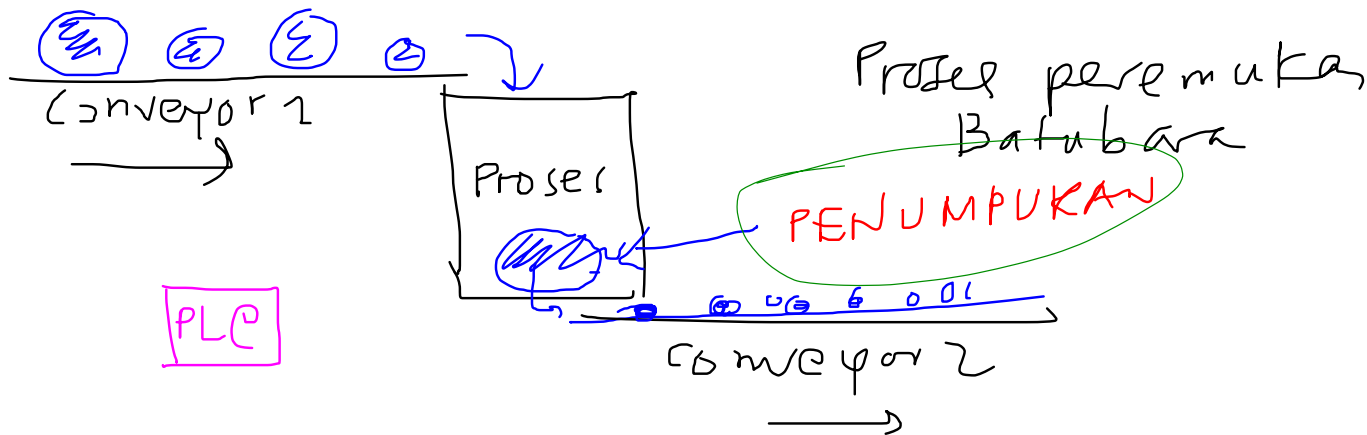


* Project 1



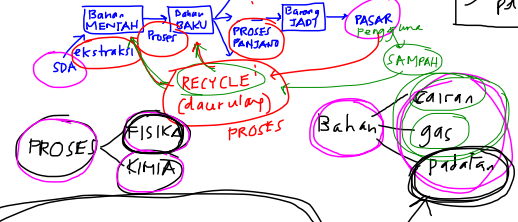
Material di SILO 1
dipindahkan
ke SILO 2
melalui conveyor

* project 2



Teknologi Kendali Proses (Process Control Technology)

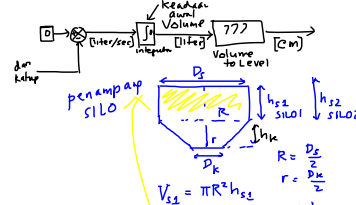
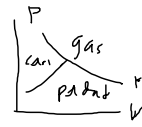
PROSES di industri manufaktur:



* Project 1 → Alat-nya pada
* Project 2 → Batubara

Simulink & Matlab

2 org → 1 laporan.
Bab I Project 1
Bab II Project 2



$$V_{s1} = \pi R^2 h_{s1}$$

$$V_k = \text{Volume kerucut besar} - \text{Volume kerucut kecil}$$

$$\frac{x}{h_k + x} = \frac{r}{R}$$

$$R \cdot x = r(h_k + x)$$

$$(R - r)x = h_k r$$

$$x = \frac{h_k r}{R - r}$$

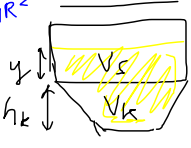
$$\text{Volume kerucut besar} = \frac{1}{3}(h_k + x) \cdot \pi R^2$$

$$\text{Volume kerucut kecil} = \frac{1}{3}x \cdot \pi r^2$$

$$V_k = \frac{1}{3}(h_k + x) \pi R^2 - \frac{1}{3}x \pi r^2$$



$$V_{in} > V_k$$



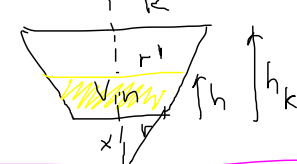
$$h = h_k + y$$

$$V_s = y \cdot \pi R^2 = V_{in} - V_k$$

$$y = \frac{V_{in} - V_k}{\pi R^2}$$

$$h = h_k + \frac{V_{in} - V_k}{\pi R^2}$$

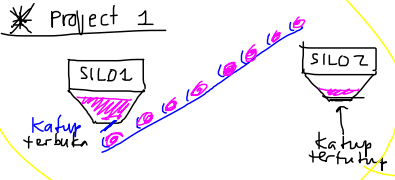
$$\frac{h+x}{h_k+x} = \frac{r}{R}$$



$$V_{ih} = \frac{1}{3}(h+x) \pi R^2 - \frac{1}{3}x \pi r^2$$

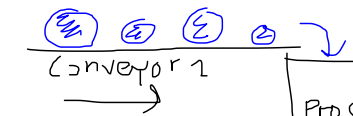
$$h = \dots \dots \dots ? ? ?$$

- * P.R: 1) Ukur R, r, h_k, h_{s1} dan h_{s2}
- 2) Selesaikan persamaan h = ... untuk V_{ih} ≤ V_k



Material di silo 1 dipindahkan ke silo 2 melalui conveyor

* Project 2



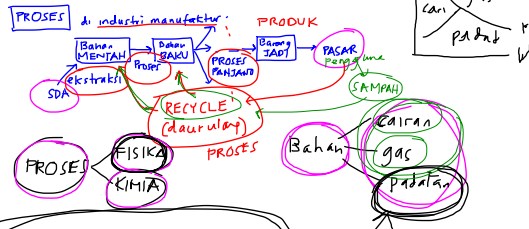
PLC

Proses permukaan Batubara

PENUMPUKAN

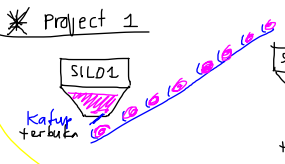
Conveyor 2

Teknologi Kendali Proses (Process Control Technology)



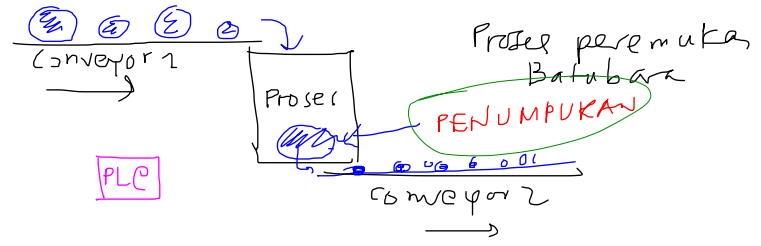
* Project 1 → Alat-nya pada Anda
 * Project 2 → Batubara

fokus
 SIMULINK & MATLAB
 2 org → 1 laporan
 Bab I Project 1
 Bab II Project 2



Material di SILO 1 dipindahkan ke SILO 2 melalui conveyor

* Project 2

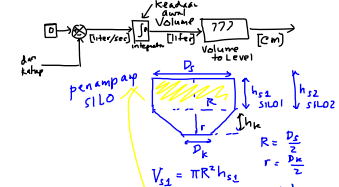


check:

$$V_{in} = V_k \text{ maka } h = h_k$$

Hitung V_{max} SILO 1

$$= V_{s1} + V_k$$



$V_{s1} = \pi R^2 h_{s1}$
 $V_k = \text{Volume kerucut besar} - \text{Volume kerucut kecil}$

$$V_k = \frac{1}{3} \pi R^2 h_k - \frac{1}{3} \pi r^2 h_k$$

$$V_{in} > V_k$$



$$h = h_k + y$$

$$V_s = y \cdot \pi R^2 = V_{in} - V_k$$

$$y = \frac{V_{in} - V_k}{\pi R^2}$$

$$h = h_k + \frac{V_{in} - V_k}{\pi R^2}$$

$$V_{in} \leq V_k$$



$$\frac{h+x}{h_k+x} = \frac{r}{R}$$

$$V_{ij} = \frac{1}{3} (h+x) \pi R^2 - \frac{1}{3} x \pi r^2 \rightarrow h = \dots \dots \dots$$

- * P.R: 1) Ukur R, r, h_k , h_{s1} dan h_{s2} → Pak Budi!
- 2) Selesaikan persamaan $h = \dots$ untuk $V_{in} \leq V_k$

1) Hubungi Pak Budi

$$\frac{(h+x)}{h_k+x} = \frac{r}{R} \rightarrow r' = \frac{R (h+x)}{h_k+x}$$

$$V_{in} = \frac{1}{3} (h+x) \pi \left[\frac{R (h+x)}{h_k+x} \right]^2 - \frac{1}{3} x \pi r^2$$

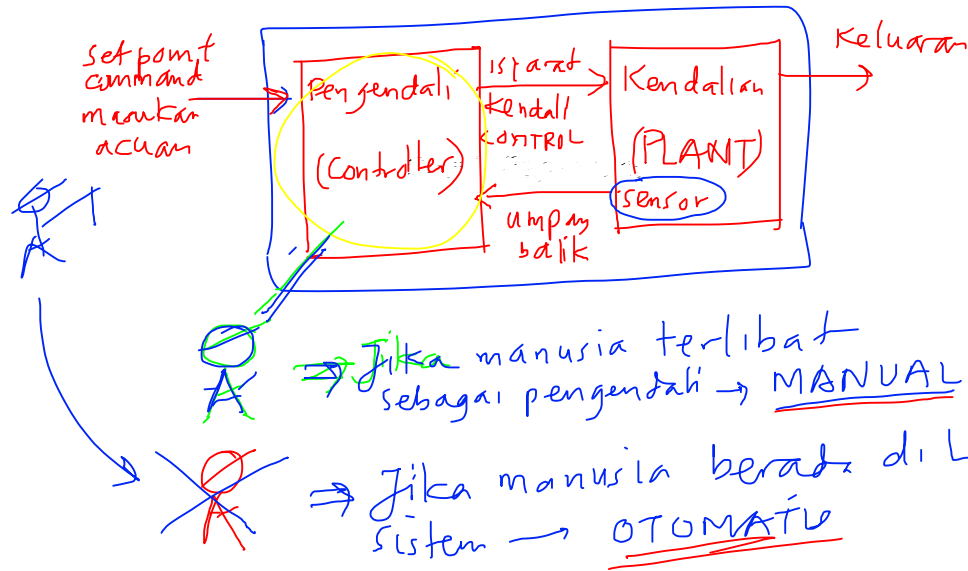
$$3V_{in} + x \pi r^2 = \frac{\pi R^2}{(h_k+x)^2} (h+x)^3$$

$$(h+x)^3 = \frac{(h_k+x)^2 (3V_{in} + x \pi r^2)}{\pi R^2}$$

$$h = \left[\frac{(h_k+x)^2 (3V_{in} + x \pi r^2)}{\pi R^2} \right]^{1/3} - x$$

Sistem Kendali PROSES

Sistem Kendali secara umum:

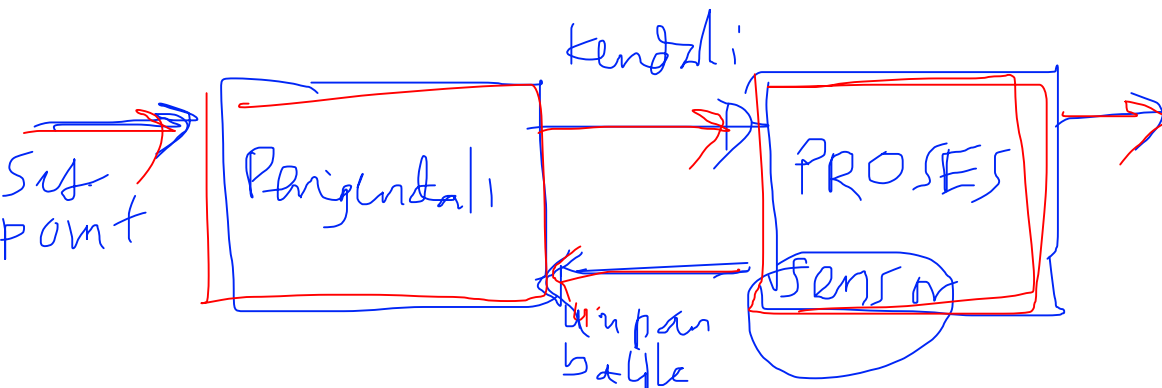


PROSES ???

pengolahan bahan di industri

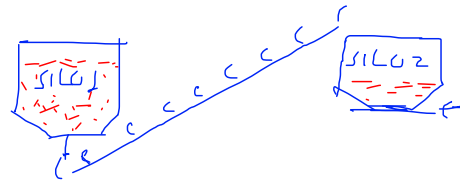
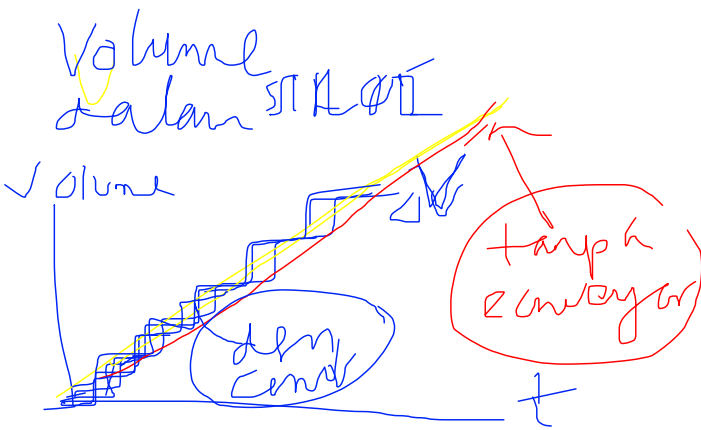
Proses Fisik
Proses Kimia

Sistem Kendali PROSES ← Sistem Kendali dengan kendalian berupa PROSES



GAS CAIR PADAT

Project :



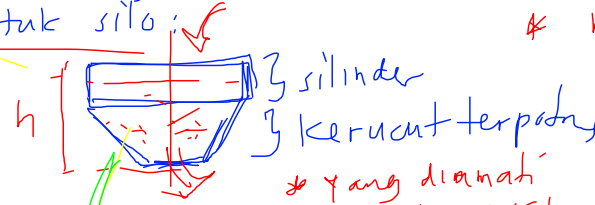
memindahkan material dari SILO 1 ke SILO 2, dengan skenario :

- * Katup pada SILO 1 dalam keadaan tertutup
- * SILO 1 diisi penuh
- * Katup silo 1 dibuka, dan material dibawa oleh conveyor ke SILO 2

* Beberapa hal yang perlu mendapat perhatian :

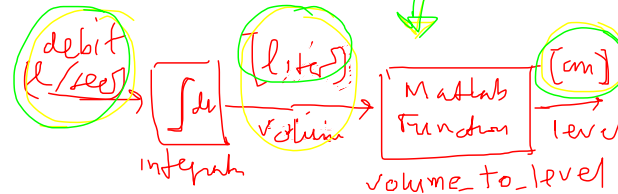
⇒ angka-angka dalam model, tidak akurat

⇒ bentuk silo :



* material yang keluar masuk dalam wadah misal : [liter/sec]

* yang diarahkan adalah LEVEL rata-rata [cm]



~~*** DITURUNKAN !!!~~

* conveyor bekerja menggunakan 'mangkuk'



**LAPORAN SIMULASI
SISTEM WATER LEVEL CONTROL DENGAN PID DAN SILO TO SILO
DENGAN MENGGUNAKAN KONVEYER**



Dajukan sebagai tugas Final
Mata Kuliah Teknik Kendali Proses

Disusun oleh :
M. Yusuf (D411 12 288)
Ruli Adi Lestari (D41112 270)

**JURUSAN TEKNIK ELEKTRO FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
2014 / 2015**

BAB 1

WATER LEVEL CONTROL DENGAN PID

A. PENDAHULUAN

Pada project ini akan dilakukan proses control level ketinggian air dari suatu prototype water level control dari GUNT. Jadi pada project ini akan dipertahankan level air dengan level tertentu sesuai dengan keinginan atau yang di setting. Selisih keinginan atau harapan dengan kenyataan disebut sebagai error. Pengendalian level ketinggian air sangat bermanfaat dalam beberapa hal diantaranya pengendalian level ketinggian air di bak penampungan air, pengendalian level ketinggian air pada proses industry dan lain-lain.

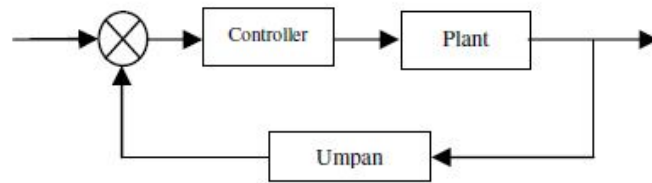
Dalam project ini digunakan suatu pengendali dengan jenis PID. Pada pengendali ini terdapat tiga mode yaitu proporsional, integral dan derivatif (PID), yang kesemua ini adalah merupakan pengendali kontinu. Dalam proses pengendalian level ketinggian air akan mengalami kesulitan dalam mencapai kendalian yang menghasilkan ketinggian air yang stabil sesuai dengan yang di setting. Untuk itu pengendali PID digunakan agar dapat mengendalikan atau menjaga agar ketinggian air tersebut tetap stabil. Pada project ini kami tidak melihat langsung alat atau prototype tersebut melainkan hanya dilihat melalui internet. Alat tersebut kemudian disimulasikan menggunakan matlab untuk mencoba melakukan pengendalian PID pada simulasi prototype tersebut.

B. TUJUAN

- a. Melakukan simulasi proses kendali ketinggian air dengan pengendali PID
- b. Melihat respon dari berbagai perlakuan yang diberikan kepada pengendali
- c. Mengetahui pengendali yang cocok dalam mengendalikan level ketinggian air
- d. Melihat output dari sistem yang tidak memiliki pengendali, diberikan pengendali, dan diberikan gangguan
- e. Mengamati pengaruh tiap-tiap pengendali pada kinerja control sistem

C. DASAR TEORI

Pengendali menghasilkan konfigurasi bertingkat (kaskade) yakni dengan menyisipkannya pada loop tertutup sehingga merupakan bagian dari penguatan (forward gain) atau dalam hal ini memiliki feedback seperti ditunjukkan pada gambar berikut:



Gambar 1. Diagram Blok Sistem Kendali

PID (*Proportional–Integral–Derivative controller*) merupakan kontroler untuk menentukan **presisi** suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut.

Adapun komponen kontrol PID terdiri dari tiga jenis yaitu **Proportional**, **Integratif** dan **Derivatif**. Ketiganya dapat dipakai bersamaan maupun sendiri-sendiri tergantung dari respon yang kita inginkan terhadap suatu plant.

Bentuk persamaan dari Pengendali PID adalah:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_D \frac{de(t)}{dt}$$

Dengan :

K_p = Konstanta *proportional*

T_i = waku *integral*

T_D = waktu *diferensial*

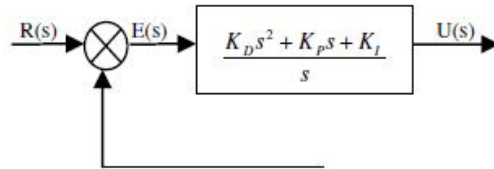
Dalam bentuk transformasi laplace adalah:

$$U(s) = K_p E(s) + K_i \frac{E(s)}{s} + K_D s E(s)$$

bentuk fungsi alihnya adalah :

$$\frac{U(s)}{E(s)} = K_p + \frac{K_I}{s} + K_D s$$

$$\frac{U(s)}{E(s)} = \frac{K_D s^2 + K_p s + K_I}{s}$$

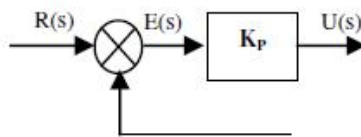


Gambar 2 Diagram Blok Pengendali PID

Kombinasi dari masing-masing pengendali P, I, dan D disebut juga sebagai pengendali PID. Kombinasi ini mempunyai keunggulan yang dimiliki masing-masing pengendali apabila dipergunakan sendiri-sendiri.

Kontrol Proporsional

Kontrol **P** jika $G(s) = kp$, dengan k adalah konstanta. Jika $u = G(s) \cdot e$ maka $u = Kp \cdot e$ dengan Kp adalah Konstanta Proporsional. Kp berlaku sebagai Gain (penguat) saja tanpa memberikan efek dinamik kepada kinerja kontroler. Penambahan harga Kp akan menaikkan penguatan sistem sehingga dapat digunakan untuk memperbesar kecepatan respons dan mengurangi ess (kesalahan dalam keadaan mantap). Pemakaian pengendali jenis ini akan membuat sistem lebih sensitif, tetapi juga cenderung mengakibatkan ketidakstabilan.

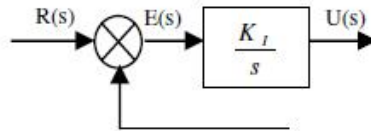


Gambar 3 Diagram Blok Pengendali Tipe P

Kontrol Integral

Jika $G(s)$ adalah kontrol I maka u dapat dinyatakan sebagai $u(t) = [\int e(t) dt] Ki$ dengan Ki adalah konstanta Integral, dan dari persamaan diatas, $G(s)$ dapat dinyatakan sebagai $u = Kd [\Delta e / \Delta t]$. Jika $e(t)$ mendekati konstan (bukan nol) maka $u(t)$ akan menjadi sangat besar sehingga diharapkan dapat memperbaiki error. Jika $e(t)$ mendekati nol maka efek kontrol I ini semakin kecil. Kontrol I dapat memperbaiki sekaligus menghilangkan respon steady-state, namun

pemilihan K_i yang tidak tepat dapat menyebabkan respon transien yang tinggi sehingga dapat menyebabkan ketidakstabilan sistem. Pemilihan K_i yang sangat tinggi justru dapat menyebabkan output berosilasi karena menambah orde system.



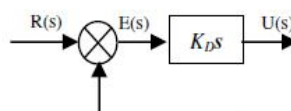
Gambar 4 Diagram Blok Pengendali tipe I

Pengendali tipe I dimaksudkan untuk menghilangkan kesalahan posisi dalam keadaan mantap.

Pengendali ini biasanya digunakan bersama tipe P + D.

Kontrol Derivatif

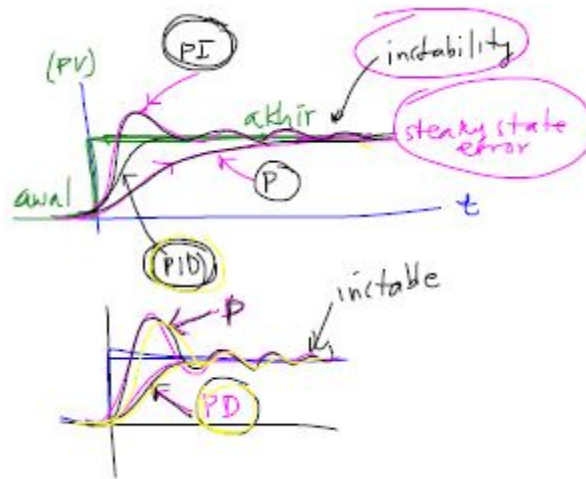
Sinyal kontrol u yang dihasilkan oleh kontrol D dapat dinyatakan sebagai $G(s) = s.K_d$. Dari persamaan di atas, nampak bahwa sifat dari kontrol D ini dalam konteks "kecepatan" atau rate dari error. Dengan sifat ini ia dapat digunakan untuk memperbaiki respon transien dengan memprediksi error yang akan terjadi. Kontrol Derivative hanya berubah saat ada perubahan error sehingga saat error statis kontrol ini tidak akan bereaksi, hal ini pula yang menyebabkan kontroler Derivative tidak dapat dipakai sendiri.



Gambar 5 Diagram Blok Pengendali Tipe D

Pengendali jenis ini digunakan untuk memperbaiki prestasi respons transien sebuah sistem kontrol. Mode ini biasanya selalu disertai tipe P, sedangkan tipe hanya digunakan bila diperlukan. Efek pengendali ini menghasilkan tindakan pengendalian yang cepat. Hal ini sangat penting bagi sistem kontrol yang perubahan bebannya terjadi secara tiba-tiba, karena dapat menghasilkan sinyal pengendali selama kesalahan (error) berubah. Karena pengendali jenis ini melawan perubahan-perubahan yang terjadi pada keluaran yang dikontrol, efeknya akan menstabilkan sistem lup tertutup dan ini dapat meredam osilasi yang mungkin terjadi.

Berikut penggambaran hasil sinyal respon dari masing-masing pengendali



Dari gambar di atas dapat dilihat bahwa pengendali P akan memiliki eror di akhir dan ada delay di awal sinyal. Pengendali PD akan mengalami instablity. Dan pengendali PID memiliki hasil sinyal respon yang grafiknya lurus sesuai dengan yang diinginkan. Jadi jika seandainya suatu sistem diberi pengendali P dan kondisi awalnya memang sudah tidak stabil, maka diberi pengendali D. Jadi pengendali PD akan menghasilkan sinyal yang mengikuti grafik yang diinginkan

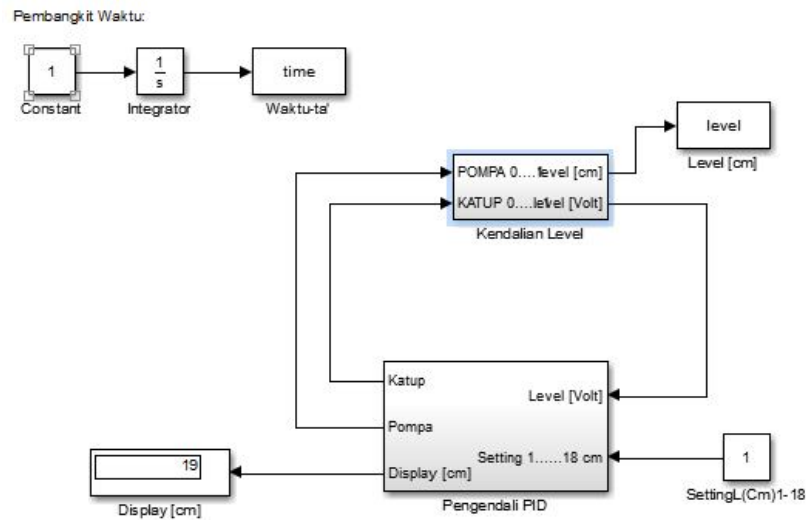
D. HASIL SIMULASI SISTEM WATER LEVEL KONTROL

Pada proses simulasi water level control ini dilakukan dengan menggunakan program matlab, yaitu toolbox simulink. Pada proses ini sistem yang diamati dibagi menjadi tiga bagian, yaitu simulasi tanpa pengendali, dengan pengendali dan dengan gangguan. Simulasi water level control tanpa pengendali adalah simulasi dari alat water level control yang ketinggian airnya selalu pada level maksimum (overflow). Level ketinggian air tersebut tidak dapat disetting atau diubah melainkan akan tetap sesuai dengan level overflow dari alat tersebut.

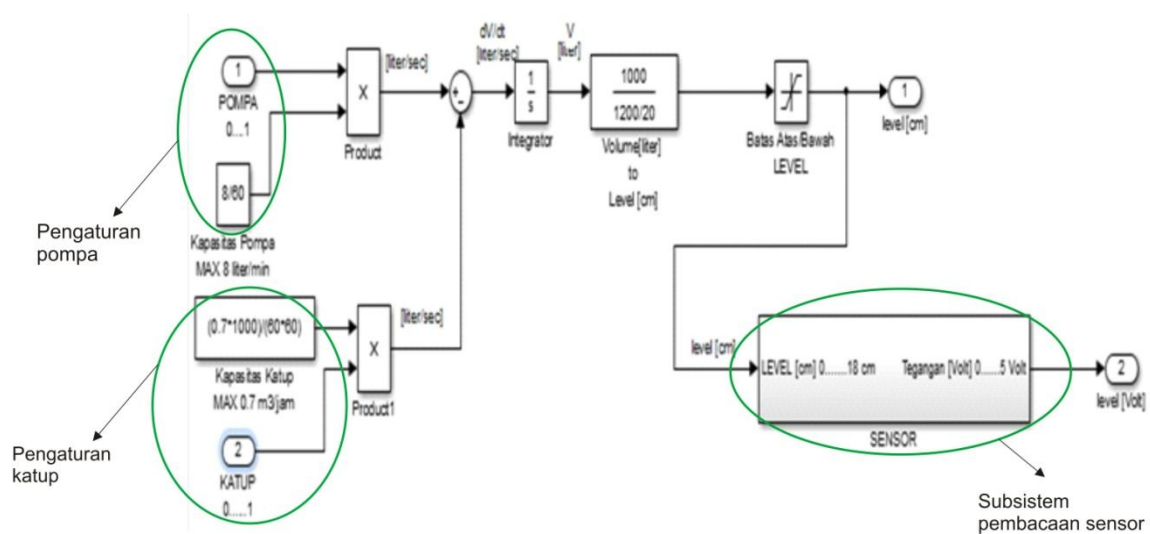
Simulasi dengan pengendali adalah simulasi water level control yang memanfaatkan pengendali PID dalam mengendalikan ketinggian air pada tangki. Hal-hal yang berpengaruh dalam kendalian tersebut adalah pompa dan katup yang akan kita atur agar menghasilkan level ketinggian air pada tangki yang sesuai dengan yang disetting atau yang diinginkan. Simulasi dengan gangguan adalah simulasi water level control yang telah memiliki pengendali namun terdapat gangguan di dalamnya. Pengendalian sistem seperti ini bertujuan untuk melihat bagaimana pengaturan pengendali PID yang akan diberikan terhadap plant agar menghasilkan output yang sesuai dengan harapan walaupun mengalami gangguan

D.1. SIMULASI WATER LEVEL CONTROL TANPA PENGENDALI

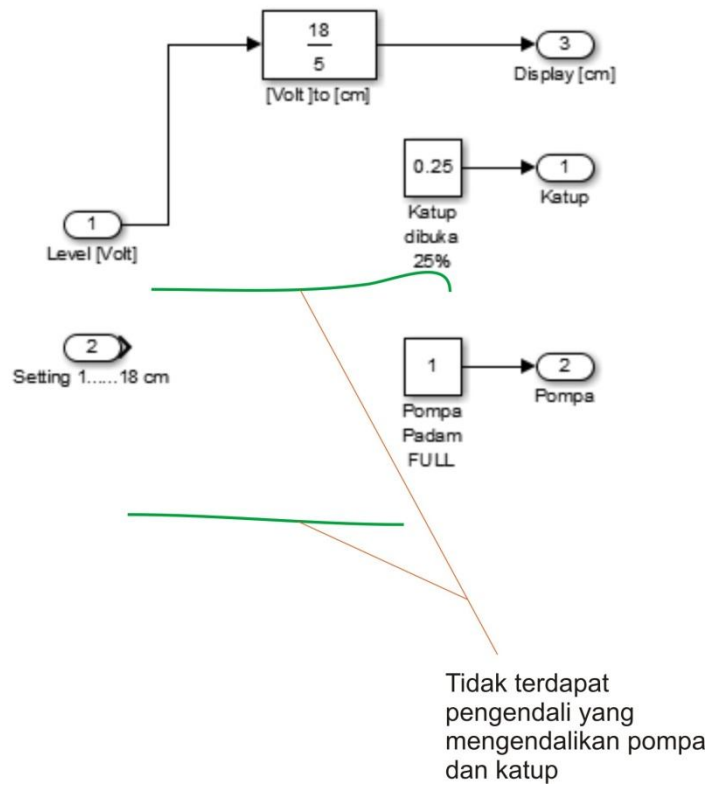
Berdasarkan gambar dari GUNT kapasitas tank atas adalah 1200 ml dan pompa adalah 8 l/m. Sedangkan luas penampangannya adalah 60 Cm². Sehingga dalam pembuatan simulasinya diatur seperti berikut.



Gambar Simulasi Water Level Control Tanpa Pengendali

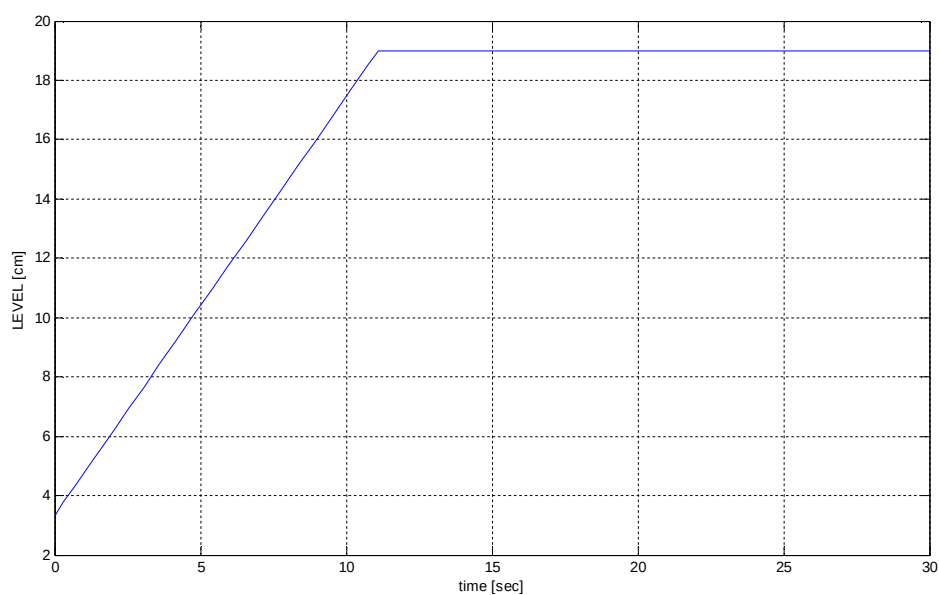


Gambar Subsistem Kendalian Level



Gambar Subsistem Pengendali PID

Dari gambar di atas dapat dilihat bahwa pada sistem belum memiliki pengendali PID pada pompa dan katup sehingga level ketinggian air pada tangki akan selalu berada pada level overflow atau maksimum yaitu pada ketinggian 19 cm. Seperti yang dapat dilihat pada hasil simulasi dari sistem di atas.

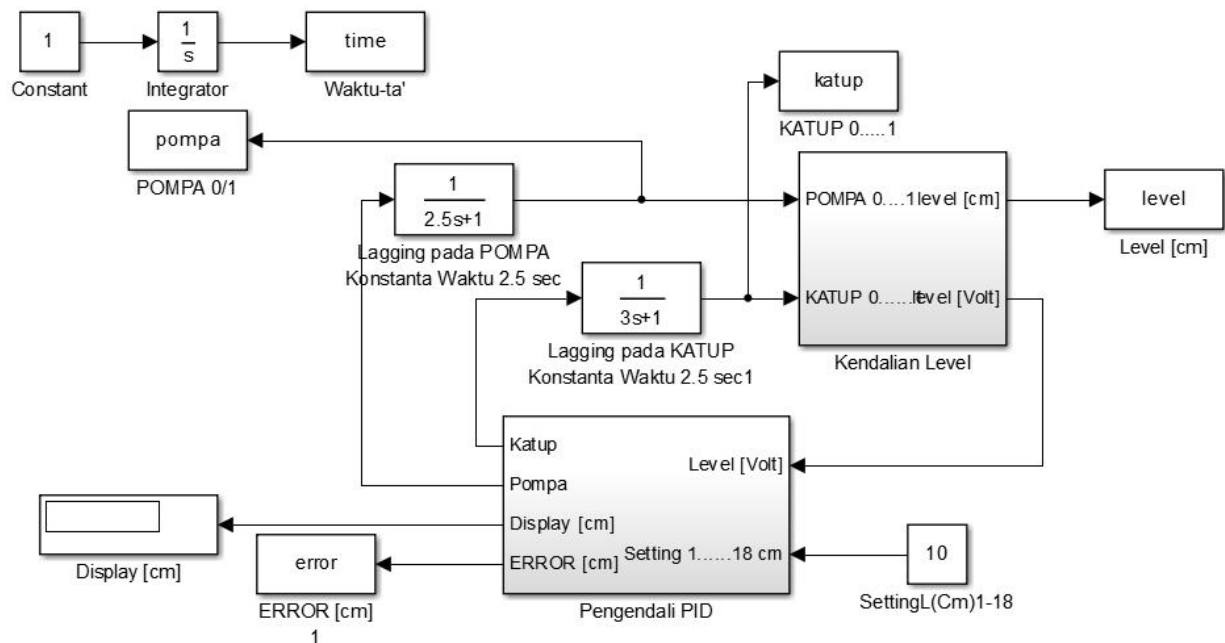


Gambar Hasil Keluaran Simulasi Tanpa Pengendali

Dari simulasi ini dapat dilihat bahwa yang menjadi variabel masukan bagi level ketinggian air dalam tangki adalah pompa dan katup dan variabel keluarannya adalah Level ketinggian air dalam tangki. Keluaran level ketinggian air ini ada dua yaitu dalam bentuk tegangan dan dalam bentuk cm. Keluaran dalam bentuk tegangan tersebut merupakan hasil pembacaan murni dari sensor yang ada pada tangki sedangkan hasil ketinggian level air dalam bentuk cm tersebut merupakan hasil dari pembacaan sensor yang telah diubah dengan mengalikannya dengan 5/18

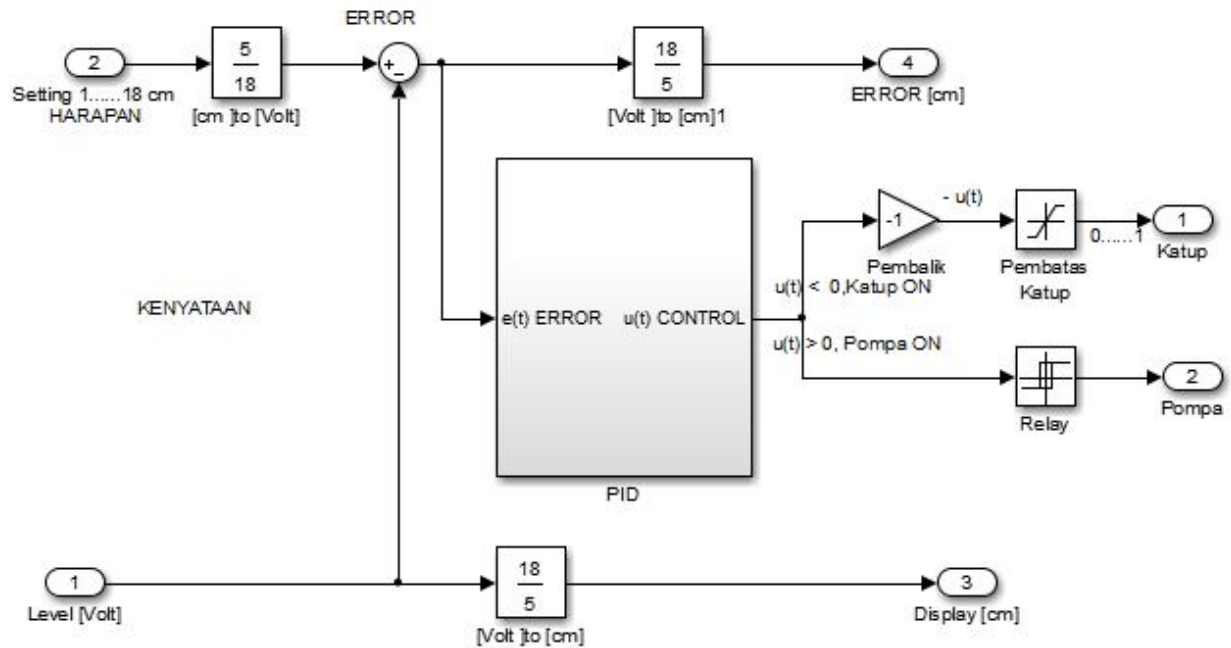
D.2. SIMULASI WATER LEVEL CONTROL DENGAN PENGENDALI

Pembangkit Waktu:



Gambar Simulasi Water Level Control dengan Pengendali

Dari gambar di atas dapat dilihat bahwa tidak ada perubahan variabel masukan dan keluaran pada bagian ketinggian level air pada rangki. Namun pada pengendali PID terdapat beberapa perubahan variabel masukan dan keluaran yang berpengaruh dalam sistem tersebut. Yang menjadi variabel keluaran dalam sistem tersebut adalah EROR, Display, Pompa dan Katup dan variabel masukannya adalah yang disetting (setting) dan hasil pembacaan sensor(level[volt]). Setting atau harapan dalam sistem kendalian ini adalah 10 cm

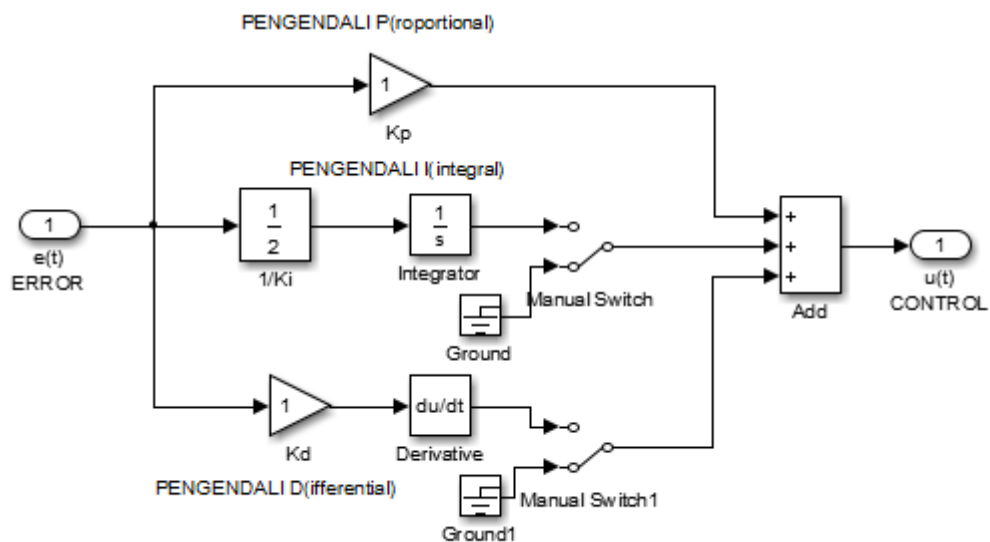


Gambar Simulasi Subsistem Pengendali PID

Pada subsistem di atas dapat dilihat bahwa pompa dan katup di atur oleh keluaran yang dihasilkan oleh PID. Apabila:

Error + = Harapan > Kenyataan = Pompa On

Error - = Harapan < Kenyataan = Katup On



Gambar Simulasi Subsistem PID

Gambar di atas adalah gambar simulasi yang pengendali mempengaruhi control pompa dan katup. Berikut ini hasil keluaran yang diperoleh dengan memberikan pengendali yang berbeda-beda dengan

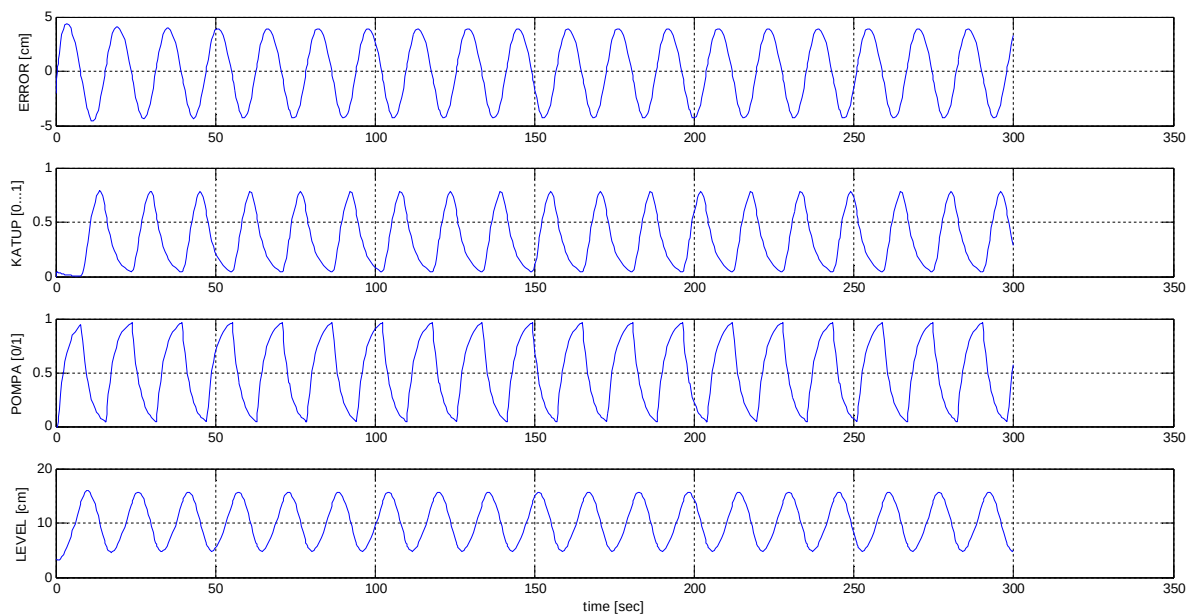
besar nilai tiap pengendali yang berbeda-beda pula. Besar nilai yang digunakan dalam tiap pengendali menggunakan metode Heuristik

Pengendali P

Dari hasil simulasi dapat dilihat bahwa pengendali P bersifat penguat sehingga mempercepat kecepatan respon. Semakin besar nilai K_p semakin cepat pula respon kerja yang dihasilkan seperti yang dapat dilihat pada hasil data berikut ini.

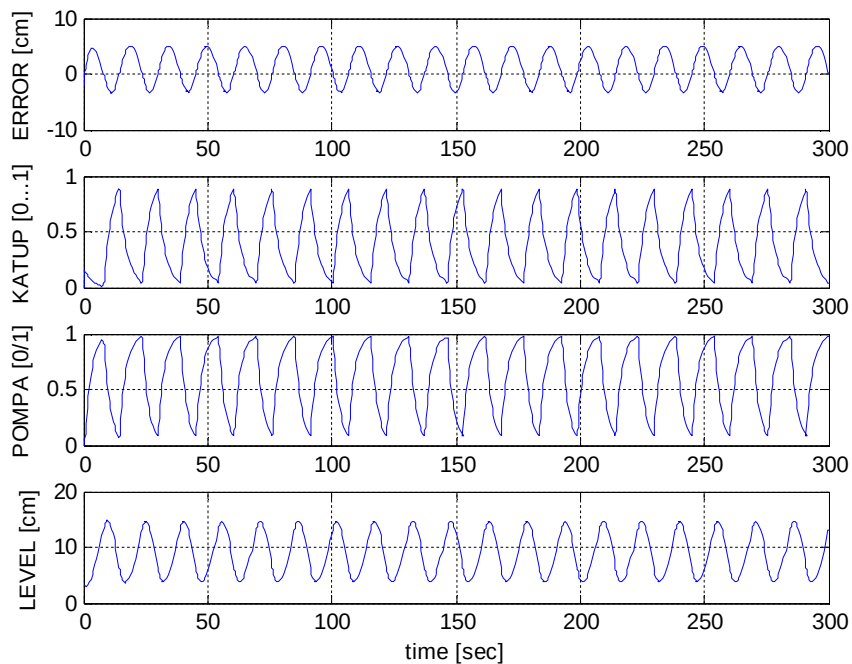
- Ketika $K_p = 1$

Ketika hanya pengendali P yang mempengaruhi dalam pengendalian. Dengan besar $K_p = 1$ maka akan diperoleh hasil keluaran dari tiap variabel keluaran yang tidak stabil. Seperti yang dapat dilihat pada gambar berikut ini.



- Ketika $K_p = 100$

Hasil yang masih tidak stabil dan cenderung sama dengan kondisi sebelumnya. Pada kondisi ini pompa dan katup mengalami kondisi on off yang begitu tajam. Hal ini karena sifat dari K_p yang semakin mempengaruhi sistem yaitu kecepatan respon sehingga diperoleh hasil seperti gambar berikut ini.

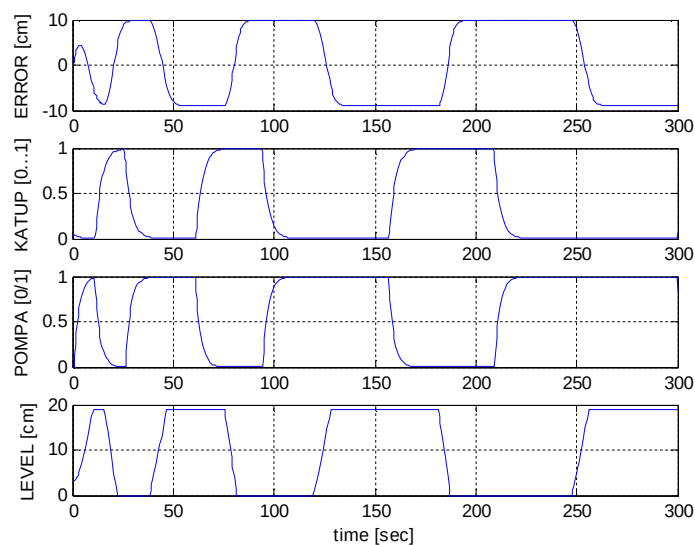


Pengendali PI

Dari hasil simulasi diperoleh hasil bahwa pengendali ini cenderung mengalami Instability. Seperti yang dapat dilihat pada hasil data berikut ini.

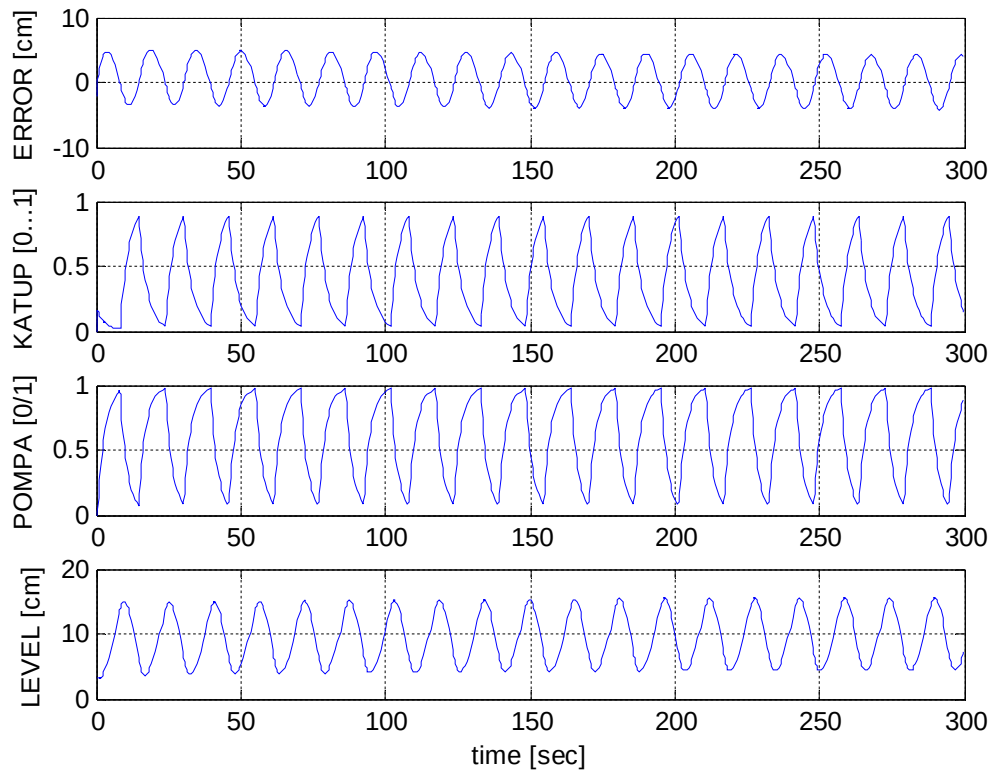
- Ketika $K_p=1$; $K_i=2$

Hasil yang dihasilkan sangat tidak stabil (Instability). Error cenderung mengalami pembesaran. Ini menyebabkan level ketinggian air selalu mengalami Overflow. Ini karena kurangnya pengaruh pengendali P sehingga repon terhadap hasil yang terjadi begitu lambat.



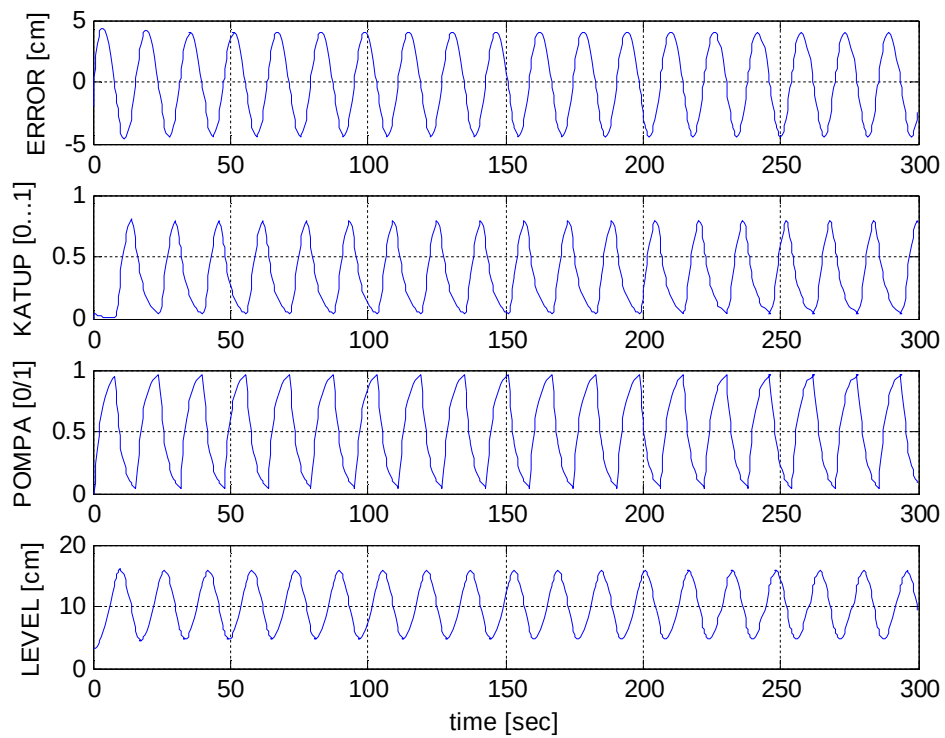
- Ketika $K_p=100$; $K_i = 2$

Keluaran yang dihasilkan masih tidak stabil tetapi dalam kondisi ini overflow sudah tidak terjadi. Dan eror mengecil



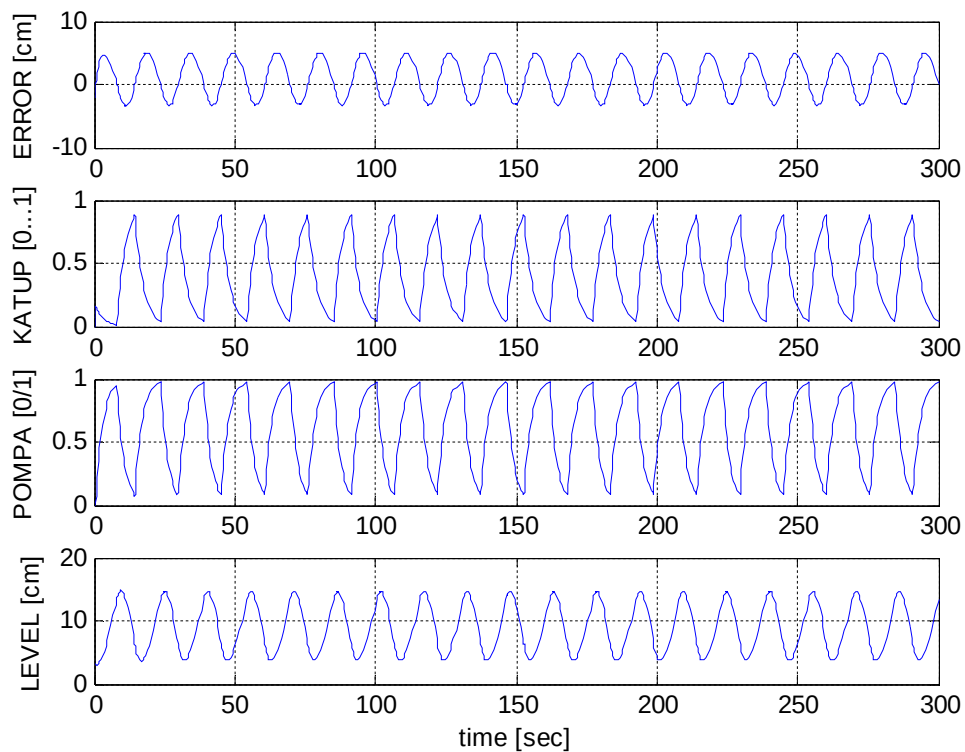
- Ketika $K_p = 1$; $K_i = 200$

Dalam kondisi ini kondisi yang dihasilkan cenderung sama dengan kondisi sebelumnya walaupun tetap memiliki perbedaan diantaranya adalah kondisi buka tutup katup sudah tidak begitu tajam. Hal ini karena pengaruh pengendali P sudah berkurang.



- Ketika $K_p = 200$; $K_i = 200$

Pada kondisi ini nilai K_p mempengaruhi kecepatan respon. Sehingga garfik yang dihasilkan pada katup dan pompa tajam. Ini seperti ketika nilai $K_p = 100$ dan $K_i = 2$. Hal ini karena adanya pengaruh pengendali P yang mempercepat respon

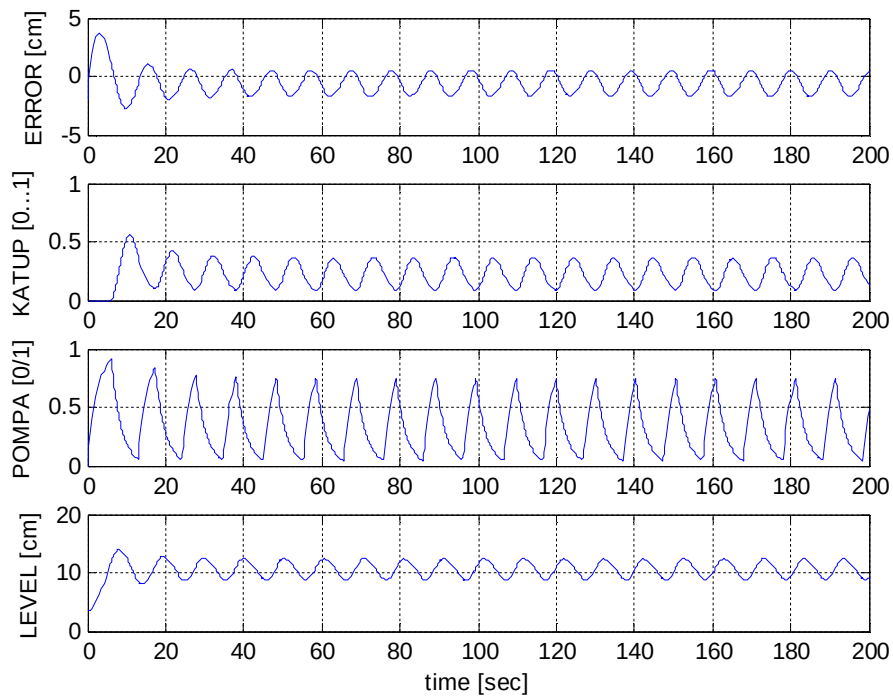


Pengendali PD

Dari hasil simulasi ini dapat dilihat bahwa pengendali ini memiliki respon terhadap laju perubahan kesalahan yang menghasilkan koreksi yang berarti sebelum kesalahan semakin besar. Jadi efeknya adalah menghasilkan tindakan pengendalian yang cepat. Semakin besar nilai K_d maka ketinggian level air dalam tangki akan semakin mendekati stabil karena adanya perbaikan kesalahan yang cepat dari pengendali D seperti yang dapat dilihat pada hasil data berikut ini.

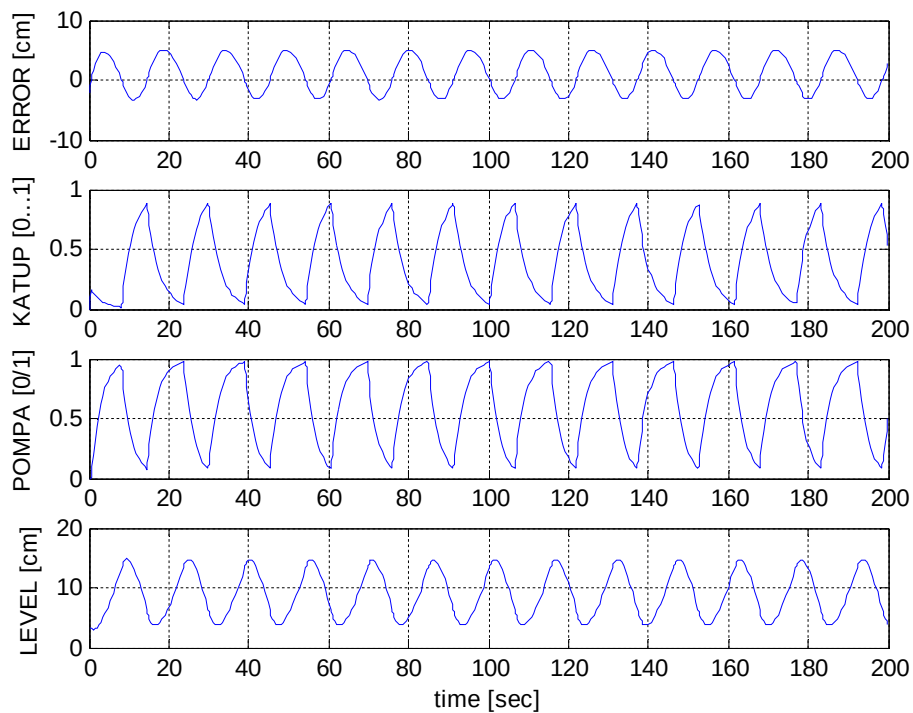
- Ketika $K_p = 1$; $K_d = 1$

Tiap-tiap variabel keluaran awalnya memiliki gerakan atau respon yang besar. Tapi kemudian mengalami peredaman dan cenderung berosilasi pada garis stabil. Hal ini karena adanya pengaruh dari pengendali D yang memiliki tindakan pengendalian yang cepat.



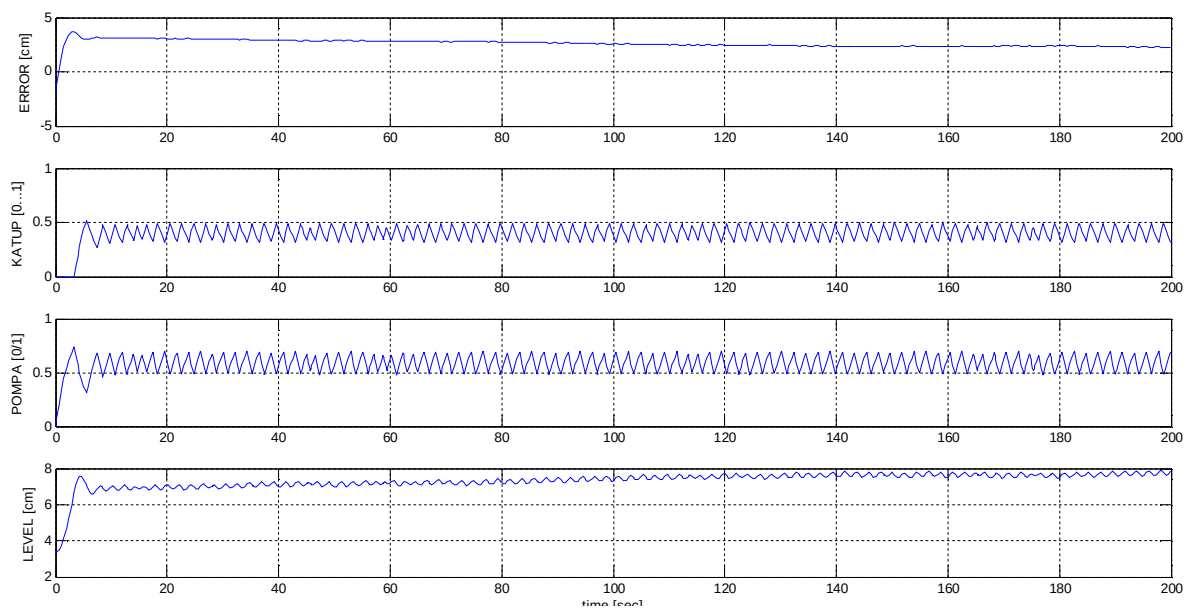
- Ketika $K_p = 300$; $K_d = 1$

Error tetap mengalami osilasi namun berada pada garis nol dengan osilasi yang cenderung tetap. Namun kerja pompa dan katup begitu tajam (cepat) sehingga level ketinggian air yang dihasilkan juga bertambah dan berkurang dengan cepat. Walaupun level air berosilasi dengan tetap pada garis seimbang (setting).



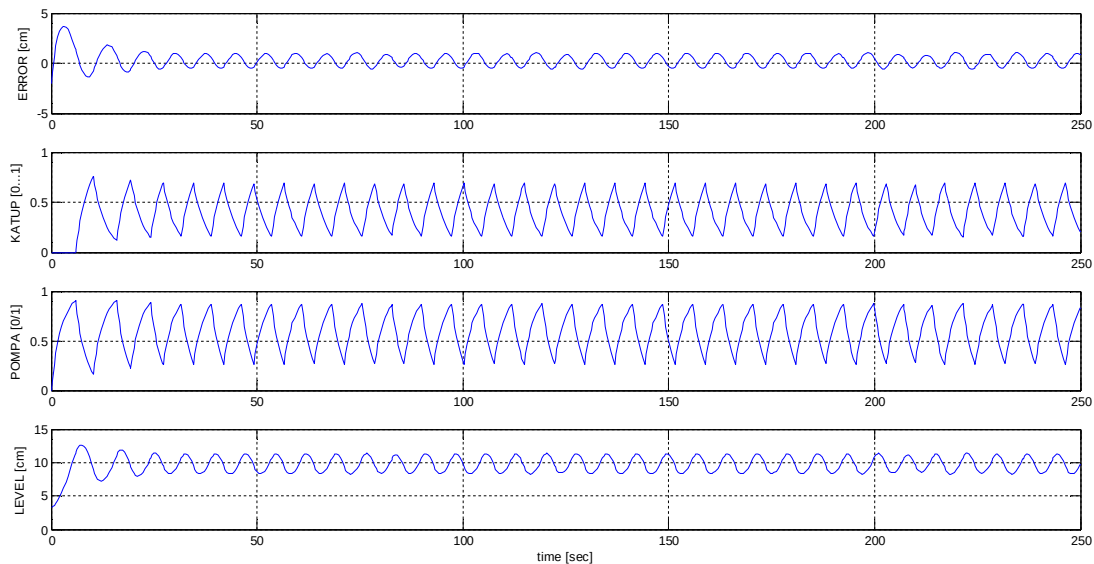
- Ketika $K_p = 1$; $K_d = 300$

Error cenderung menuju titik nol. Kerja katup dan pompa tidak begitu lama dalam suatu kondisi sehingga menghasilkan buka tutup katup dan on off pompa yang cepat. Sehingga dihasilkan level air yang menuju garis seimbang (setting). Ini karena adanya pengaruh pengendali D yang memperbaiki kesalahan dengan cepat.



- Ketika $K_p = 300$; $K_d = 300$

Nilai K_p kembali mempengaruhi kinerja sistem sehingga diperoleh hasil yang mirip ketika $K_p = 300$ dan $K_d = 1$.

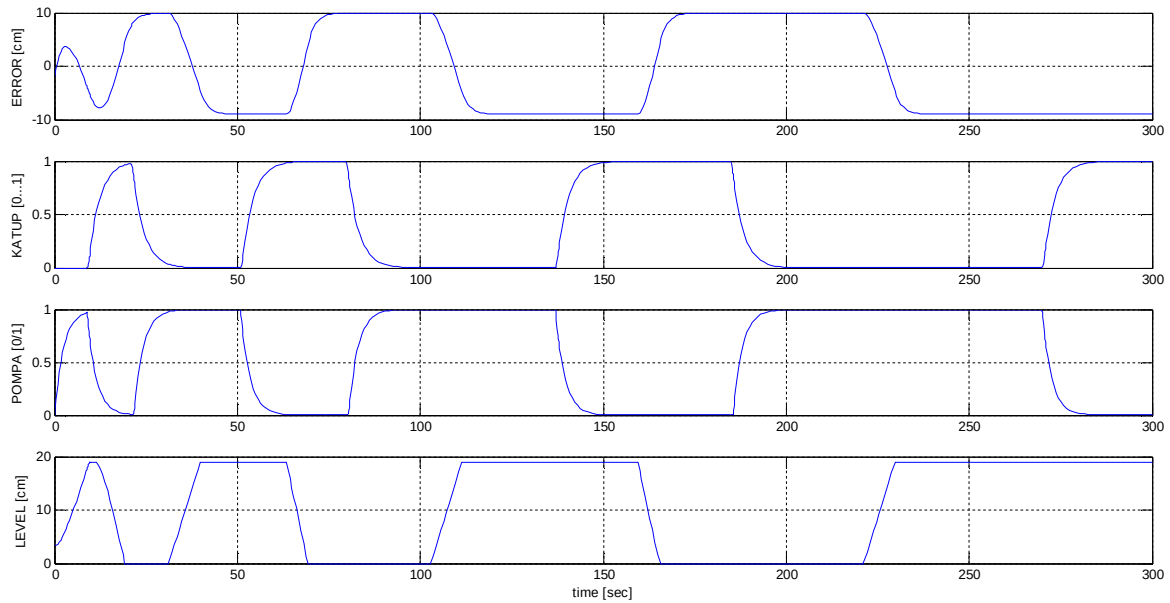


Pengendali PID

Dari hasil simulasi ini dapat dilihat bahwa pengendali PID menggabungkan keunggulan-keunggulan yang dimiliki oleh tiap-tiap pengendali. Pengendali P yang bersifat sebagai penguat yang mempercepat respon atau kinerja, pengendali I yang memperbaiki kesalahan akibat buruknya respon transien dan pengendali D yang memiliki respon yang cepat terhadap kesalahan sehingga kesalahan tidak akan membesar. Seperti yang dilihat pada pengambilan hasil data berikut ini.

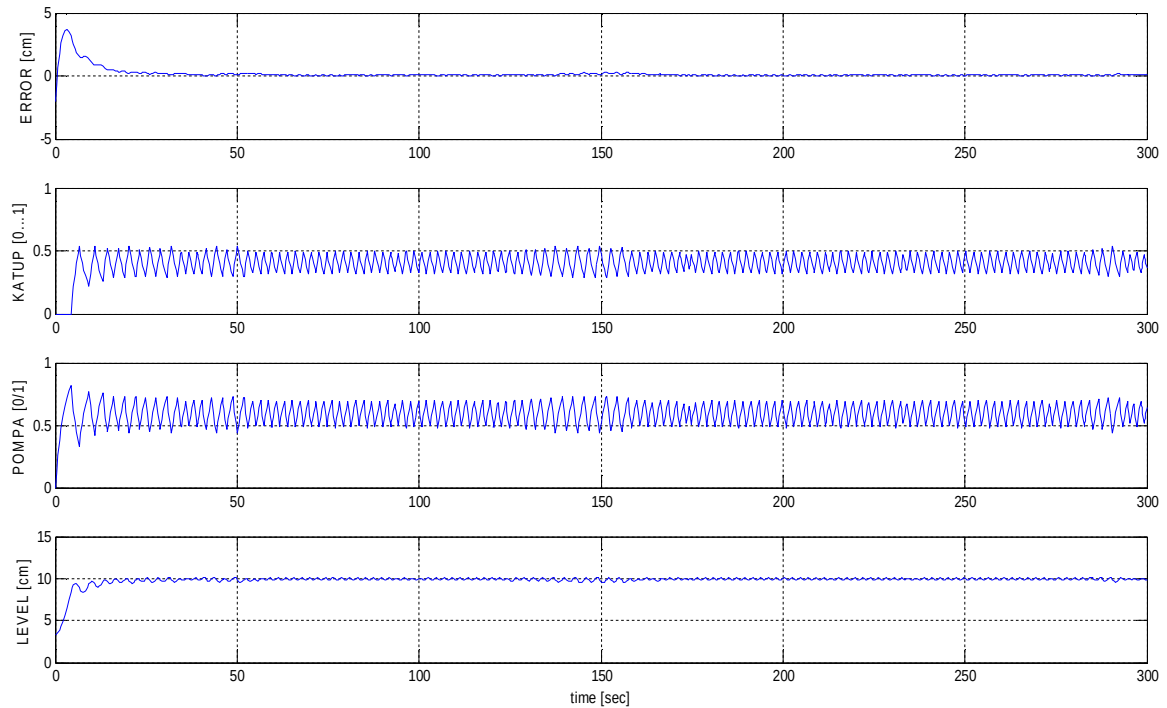
- Ketika $K_p = 1$; $K_i = 2$; $K_d = 1$

Pengaruh I sangat berpengaruh dalam sistem ini dapat dilihat respon semakin yang cenderung semakin lama. Sehingga level ketinggian air cenderung mengalami overflow dan kekosongan dalam waktu yang sangat lama.



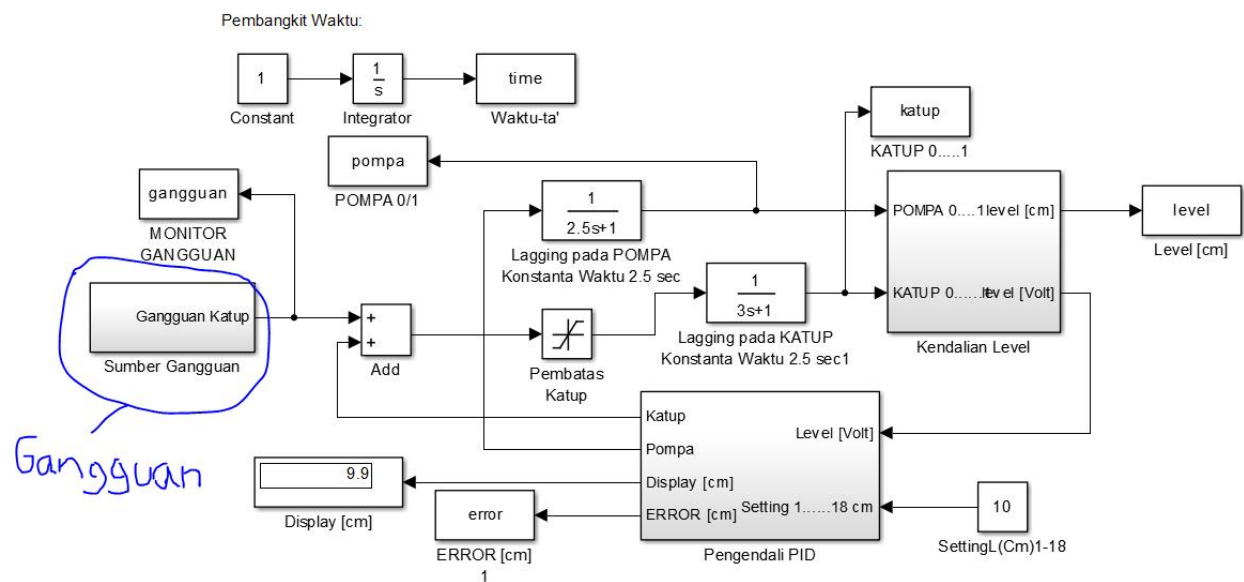
- Ketika $K_p = 200$; $K_i = 100$; $K_d = 1000$

Untuk itu kami melakukan pengaturan pada besar nilai K_p , K_i dan K_d untuk menghasilkan nilai level ketinggian air pada tangki yang seimbang sesuai dengan yang disetting. Nilai K_d dibuat besar agar respon terhadap kesalahan semakin cepat. Dan selanjutnya nilai K_p dibuat sebesar 200 agar penguatan dari kecepatan respon dari kerja sistem tidak begitu besar namun tidak pula labat. Selanjutnya nilai K_i dibuat paling rendah yaitu sebesar 100 yang mempengaruhi perbaikan kesalahan akibat buruknya respon transien. Sehingga dihasilkan level ketinggian air pada tangki yang cenderung seimbang. Dan dapat dilihat kinerja dari pompa dan katup semakin cepat dalam hal buka tutup dan on off namun tidak begitu lama dalam satu kondisi saja (misalnya lama dalam kondisi on). Sistem ini tentunya memiliki eror yang rendah. Semua hasil tersebut dapat dilihat pada gambar berikut ini.

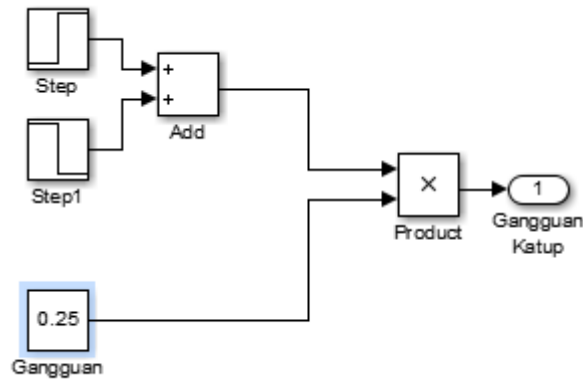


D.3. SIMULASI WATER LEVEL CONTROL DENGAN PEMBERIAN GANGGUAN

Pada percobaan simulasi ini kita akan mencoba untuk mengendalikan ketinggian level air pada tangki apabila terdapat gangguan pada sistem. Istilah yang digunakan dalam memberikan gangguan adalah step, yaitu pemberian gangguan pada detik ke n-n. Berikut gambar hasil simulasinya dengan matlab.



Gambar Simulasi dengan Gangguan

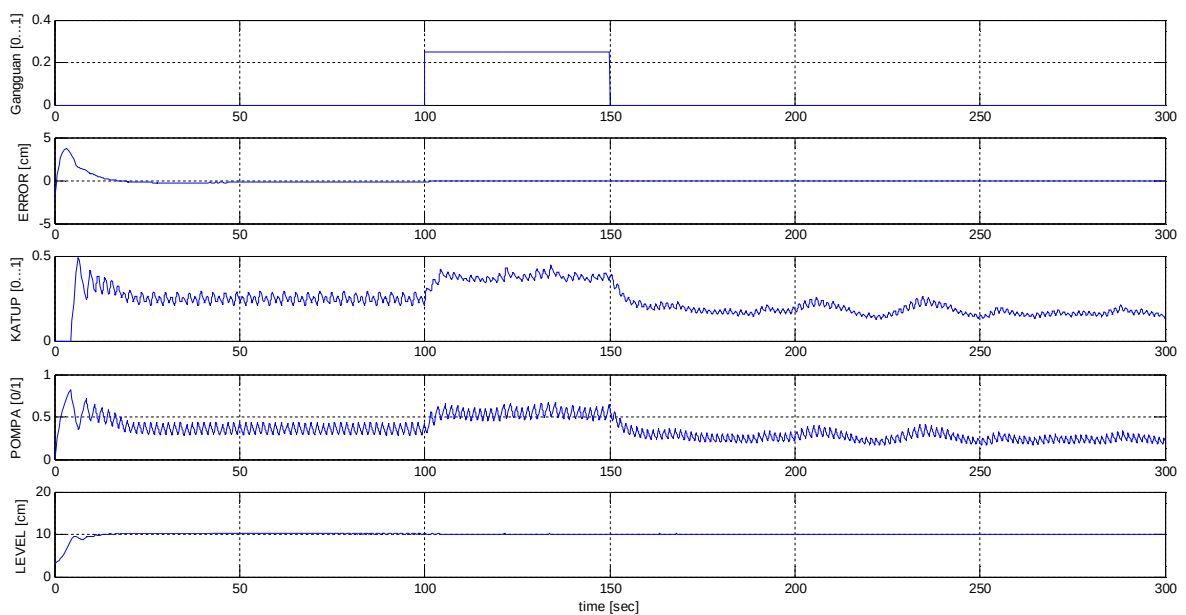


Gambar Subsistem Gangguan

Dari gambar di atas dapat dilihat bahwa gangguan mempengaruhi kinerja dari katup dan pompa. Gangguan inilah yang akan dicoba untuk ditangani dengan meminimalisir error menggunakan pengendali. Berikut hasil pengambilan data yang kami lakukan untuk mengendalikan gangguan yang ada.

Diatur $K_p = 10$; $K_i = 10$; $K_d = 50$

Pengaturan ini adalah pengaturan yang kami peroleh setelah memberikan nilai yang berubah-ubah pada nilai K_p , K_i , dan K_d sehingga diperoleh level ketinggian air pada tangki yang cenderung seimbang. Nilai K_d dibuat sangat besar. Hal ini agar kesalahan yang terjadi segera ditangani. Sehingga diperoleh kinerja sistem dari masing-masing variabel seperti gambar berikut ini.



E. KESIMPULAN

1. Sistem pengendali PID dapat menstabilkan system atau output mendekati atau sesuai dengan harapan walaupun terdapat gangguan
2. Pengendali P bersifat seperti penguat sehingga mempercepat respon. Pengendali I dapat memperbaiki sekaligus menghilangkan respon steady-state, namun pemilihan K_i yang tidak tepat dapat menyebabkan respon transien yang tinggi sehingga dapat menyebabkan ketidakstabilan sistem.
3. Pengendali tipe D memiliki respon terhadap laju perubahan kesalahan menghasilkan koreksi yang berarti sebelum kesalahan tersebut bertambah besar, jadi efeknya adalah menghasilkan tindakan pengendalian yang cepat.

BAB 2

SILO TO SILO DENGAN MENGGUNAKAN KONVEYER

A. Pendahuluan

Proses yang terjadi dalam industri manufaktur merupakan suatu sistem yang terdiri dari berbagai subsistem. Masing subsistem menjalankan fungsi sehingga sistem yang lebih besar mampu menyelesaikan suatu proses produksi. Terdapat banyak subsistem yang menyusun sistem yang lebih besar. Subsistem tersebut bekerja dengan fungsi dan bagian tertentu dari sistem.

Sistem silo ke conveyor dan conveyor ke silo merupakan sistem yang umum dalam sistem produksi. Sistem ini berfungsi sebagai sistem transportasi material dalam proses produksi. Material akan diangkut, baik sebelum diolah ataupun setelah diolah menggunakan sistem ini.

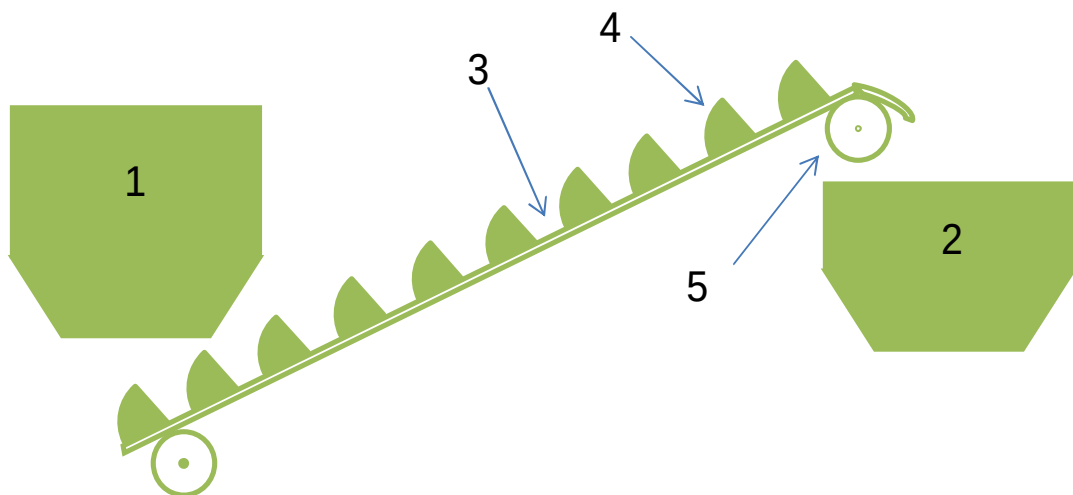
B. Tujuan Proyek

Proyek berupa simulasi Sistem kerja Silo ke Silo dengan menggunakan conveyor memiliki :

1. Dapat mengetahui Prinsip Kerja Sistem Silo ke Silo dengan menggunakan conveyor.
2. Mengetahui variabel yang berpengaruh pada pengiriman material.
3. Mampu membuat mendesain sistem dan melakukan simulasi.

C. Perancangan dan Pemodelan Sistem

1. Perancangan



Keterangan :

1. Silo 1
2. Silo 2
3. Konveyor
4. Belt / Cawang
5. Motor

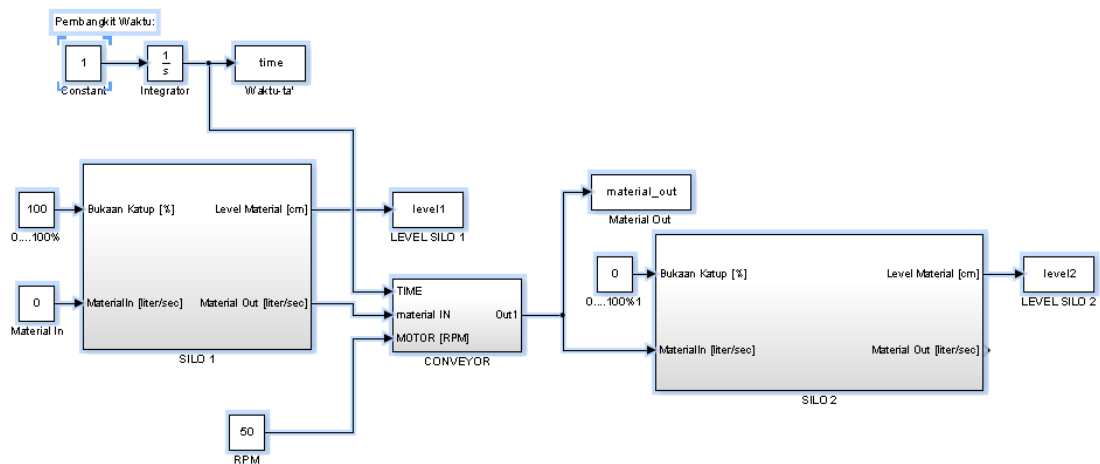
Dimensi – dimensi sistem :

- a. SILO 1
 - Diameter bagian Tabung = 26 cm

- Tinggi bagian tabung = 20,5 Cm
 - Tinggi Bagian kerucut = 12 cm
 - Diameter kerucut yang kecil = 4,5 cm
 - Diameter kerucut besar = diameter tabung
- b. Silo 2
- Diameter bagian Tabung = 26 cm
 - Tinggi bagian tabung = 10 cm
 - Tinggi Bagian kerucut = 12 cm
 - Diameter kerucut yang kecil = 4,5 cm
 - Diameter kerucut besar = diameter tabung
- c. Konveyor
- Konveyor terdiri dari 15 cawang antara Silo 1 dan Silo 2

2. Pemodelan

Sistem ini dimodelkan dengan menggunakan program Matlab 2013b, dengan tiga 4 bagian utama yaitu pembangkit waktu, Subsystem Silo1, Subsystem Silo2 dan Subsystem Conveyor. Bagian ini merupakan bagian penyusun sistem yang akan disimulasikan. Untuk melihat bentuk pemodelan dapat dilihat pada gambar berikut.

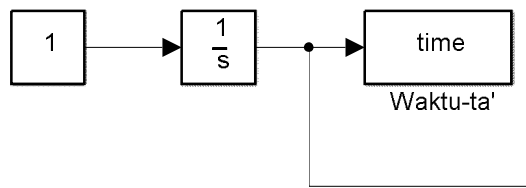


Gambar 2. Model Sistem Silo to Silo

Untuk mengamati secara lebih rinci, bagian tersebut perlu dijabarkan dalam sistem tersebut.

a. Pembangkit Waktu

Dalam sebuah sistem, pembangkit waktu dijadikan sebagai parameter waktu untuk melakukan perhitungan. Parameter ini digunakan oleh sistem pada besaran-besaran yang membutuhkan besaran waktu. Misalnya dalam perhitungan frekuensi, periode, atau dalam pembangkit sinyal sinus.

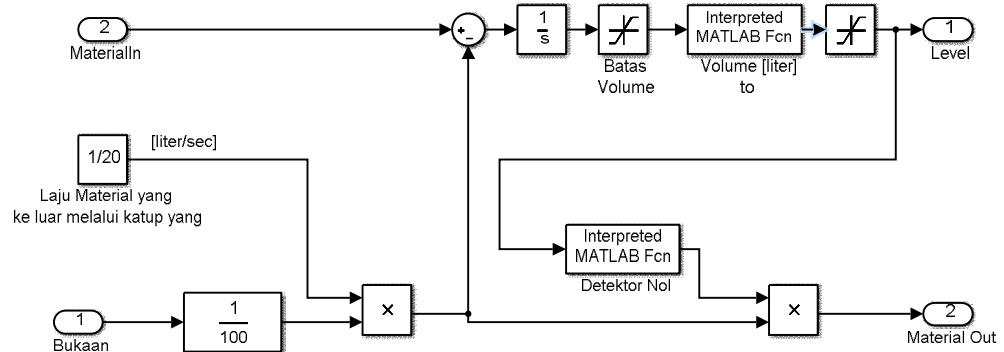


Gambar 3 : Pembangkit Waktu

b. Sub Sistem Silo 1 dan Silo 2

Pemodelan silo 1 sebagaimana yang terdapat pada gambar 4. Dalam model silo 1 terdapat 2 input yaitu material dengan satuan liter/sec dan bukaan katup dalam %. Untuk output ada 2, yaitu level material dalam (cm) dan material out dalam liter/second.

Saat input material berupa bahan padat masuk, untuk menghasilkan output berupa level (cm), data bahan tersebut perlu dilakukan konversi. Data Inputan dalam bentuk liter/second mesti dikonversi ke dalam bentuk centimeter untuk agar dapat menyatakan level material yang ada dalam silo.



Gambar 4. Pemodelan Silo 1 dan Silo 2

Cara kerja dari sistem yang terdapat pada silo 1 sama dengan sistem yang pada silo 2, yang membedakan keduanya adalah ukuran silo tersebut. Ukuran tinggi silo 1 lebi besar dari pada silo 2. Sebagai mana hasil pengukuran, silo 1 memiliki tinggi 33,5 cm dan silo 2 tingginya 20 cm.

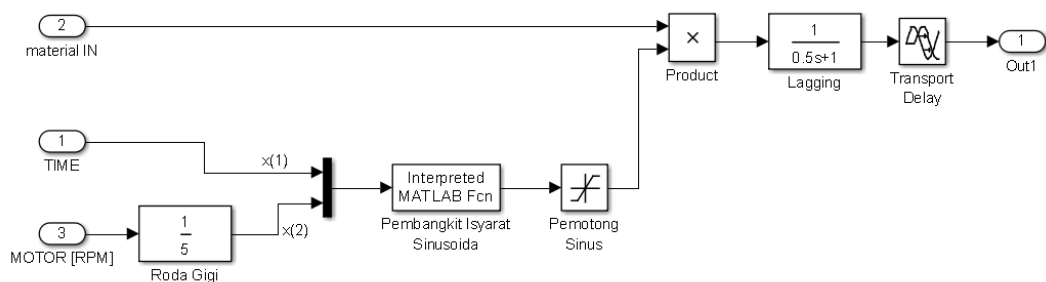
c. Subsystem Conveyor

Conveyor merupakan pengangkut material dari silo 1 ke silo 2. Conveyor akan membawa material dengan menggunakan cawang yang terpasang pada conveyor. Cawang ini sebagai penampung material yang akan ditumpahkan pada silo 2. Conveyor

sangat penting untuk menjaga sistem tetap bekerja dan dapat pula meningkatkan efisiensi sistem.

Untuk membentuk model yang dapat mewakili cara kerja dari conveyor, maka dibutuhkan 3 buah input, dan 1 output. Inputan terdiri dari input material, pewaktuan (time), dan kecepatan motor. Ketiga inputan inilah yang membuat sistem dapat bekerja. Sedangkan output hanya berupa material. Walaupun jika lebih diperinci, output dari sistem ini bisa lebih dari 1.

Untuk menjalankan conveyor dibutuhkan sebuah penggerak, maka dimasukkanlah inputan berupa kecepatan motor. Dalam pemodelan conveyor, waktu dibutuhkan sebagai besaran untuk membangkitkan gelombang sinus. Fungsi gelombang sinus untuk memodelkan bentuk cawang yang ada pada conveyor. Selanjutnya, material yang dibawah oleh conveyor. Untuk melakukan pemodelan sinyal sinus yang telah dipotong, kemudian dikalikan dengan material input. Sehingga dari sinilah akan didapatkan jumlah material yang terbawa pada setiap cawang yang ada pada conveyor. Berikut ini adalah blok diagram yang terdapat dalam model conveyor dengan menggunakan matlab.



Gambar 5. Model Sistem Conveyor

D. Pengujian dan Analisis

1. Pengujian

Pengujian Model akan dilakukan dengan merubah variabel yang ada pada koneyor. Perubahan dilakukan untuk mengamati sejauh mana pegraruh variabel tersebut terhadap efisiensi conveyor. Ada dua variabel yang diubah pada konveyer yaitu kecepatan motor dan tinggi cawang. Untuk mengubah tinggi cawang, maka amplitudo pada sintaks pembangkitit sinyal sinus yang akan diubah.

Dari pengujian diperoleh data sebagai berikut.

Untuk Kecepatan Conveyor :

Volumer mterial silo 1 = 5 liter

Level Silo 1 : 16.6053

No	Kecepatan motor	Tinggi Cawang (amplitudo pemb. Sinus)	Volume Silo2	Level Silo2
1	100	2	2.435	11.77
2	200	2	2.43	11.74
3	300	2	2.408	11.72
4	400	2	2.415	11.73

Untuk Amplitudo :

Volumer mterial silo 1 = 5 liter

Level Silo 1 : 16.6053

No	Kecepatan motor	Tinggi Cawang (amplitudo pemb. Sinus)	Volume Silo2	Level Silo2
1	250	2	2.419	11.74
2	250	2.5	2.543	11.98
3	250	3	2.624	12.13
4	250	3.5	2.681	12.24

Analisi Data

Data di atas merupakan hasil perhitungan menggunakan model simulasi matlab sebagaimana yang digunakan. Variabel yang diubah nilainya yaitu kecepatan motor yang menggerakkan conveyor. Hal tersebut dilakukan untuk mengamati perubahan level yang terjadi pada silo 2. Berdasarkan data di atas diperoleh adanya perubahan pada volume dan level material yang ada pada silo2.

Untuk perubahan pada variabel kecepatan, perubahan yang didapat tidak begitu signifikan. Jika kita amati data pada variabel tersebut, terdapat kecenderungan bahwa semakin kecepatan ditambahkan level material yang pada silo 2 semakin menurun. Sehingga yang terlihat bahwa semakin cepat putaran motor, maka akan menyebabkan material yang diangkut semakin banyak yang hilang. Pada kecepatan 100 rpm, level pada silo2 sebesar 11.77 cm. Saat kecepatan motor 200 rpm, nilai levelnya menurun sebesar 0.03 cm. Begitu pun saat 300 rpm, mengalami penurunan dengan nilai level 11.72. Namun pada kecepatan 400rpm nilai level meningkat 0.01. Walaupun perubahan berada pada rentang nilai yang kecil, namun jika pada material dengan jumlah yang besar nilai ini tentu sangat berpengaruh pada hasil produksi dan efisiensi Alat.

Untuk perubahan pada amplitudo sinyal sinus, atau dalam hal ini, nilai kemiringan cawang pada conveyor, data yang diperoleh sebagaimana yang diperlihatkan oleh tabel diatas. Dari data tersebut, terlihat ada perbedaan antara kecepatan dan amplitudo. Jika pada data kecepatan kecenderungan dari data tersebut berbanding terbalik, sedangkan pada amplitudo data cenderung berbanding lurus. Saat nilai amplitudo 2 cm, level silo2 11,74 cm, ketika 2,5 level meningkat 11,98, ketika nilai amplitudonya terus dinaikkan menjadi 3, level silo 2 menjadi 12.13, dan begitu pula pada amplitudo 3.5, kenaikan levelnya menjadi 12.24.

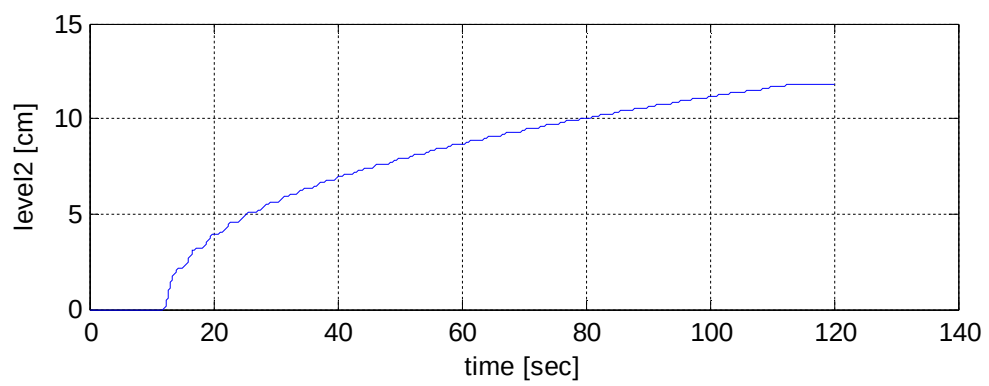
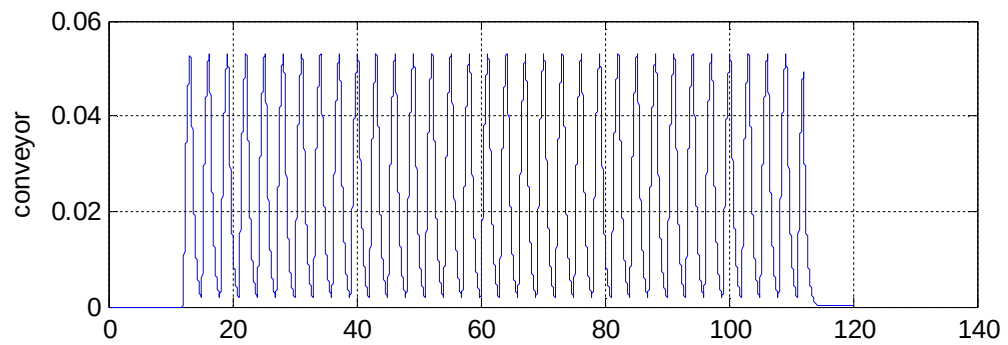
E. Kesimpulan

Dari data yang telah diambil pada simulasi silo ke silo dengan conveyor, ada beberapa hal yang dapat disimpulkan.

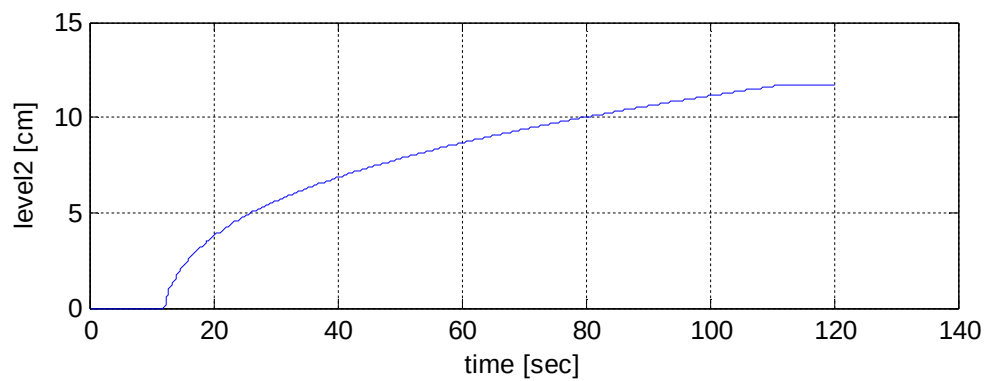
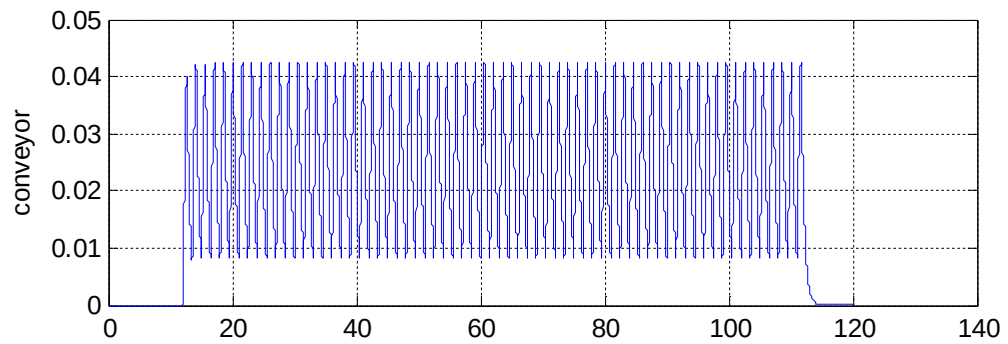
1. Prinsip kerja sistem ini sangat ditentukan oleh kualitas conveyor.
2. Kecepatan motor conveyor mengurangi efisiensi dari conveyor, sedangkan tinggi cawang dapat meningkatkan efisiensi conveyor.

Data Grafik

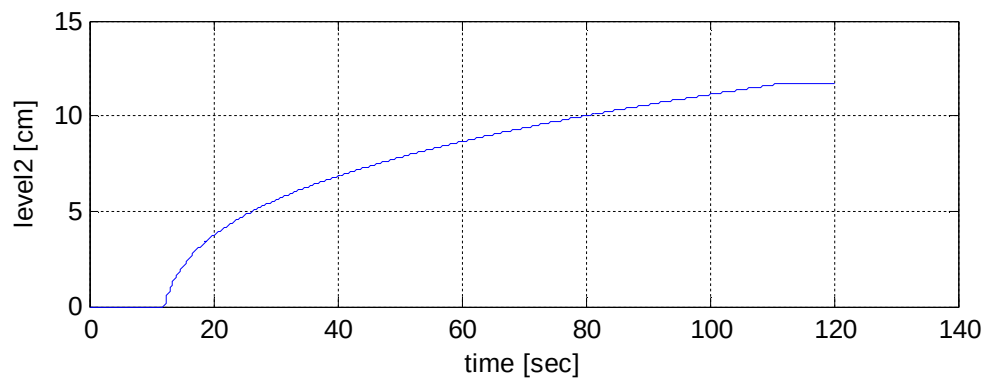
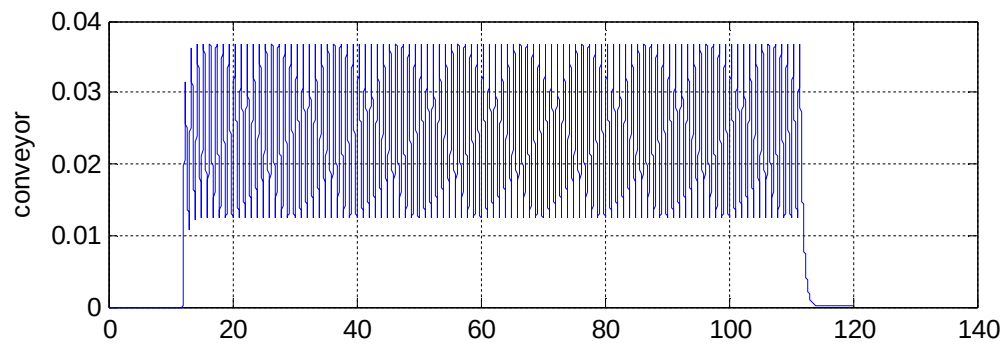
Kecepatan motor = 100 rpm



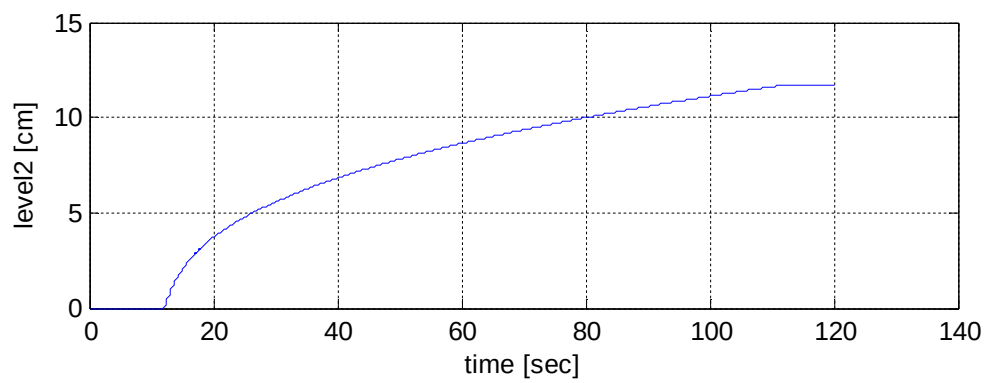
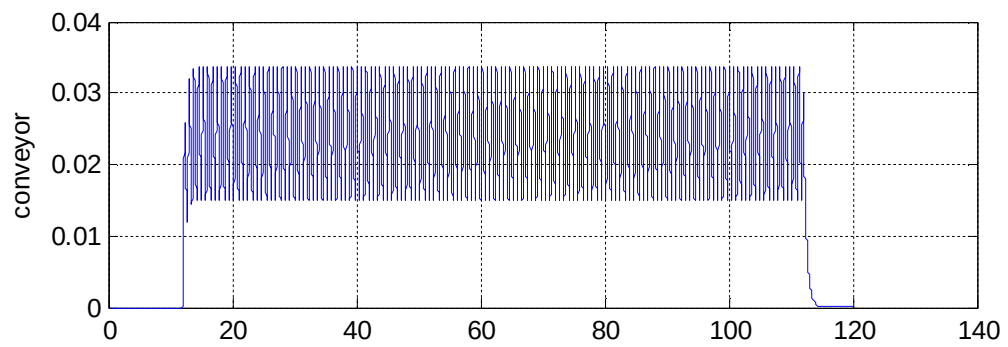
Kecepatan motor = 200 rpm



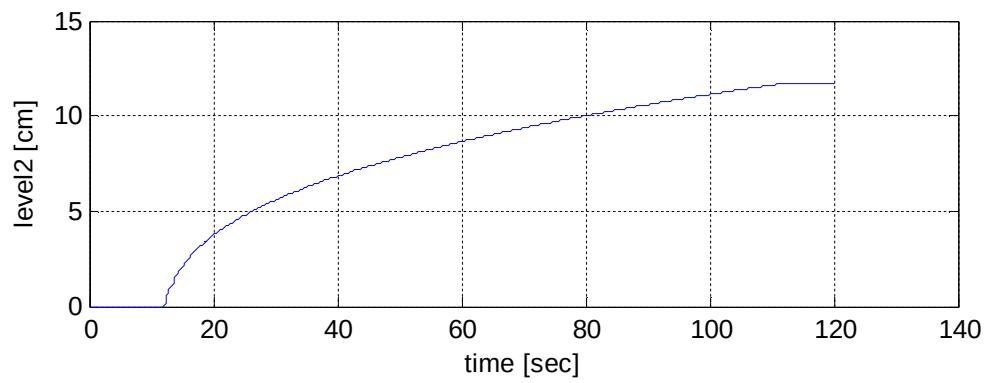
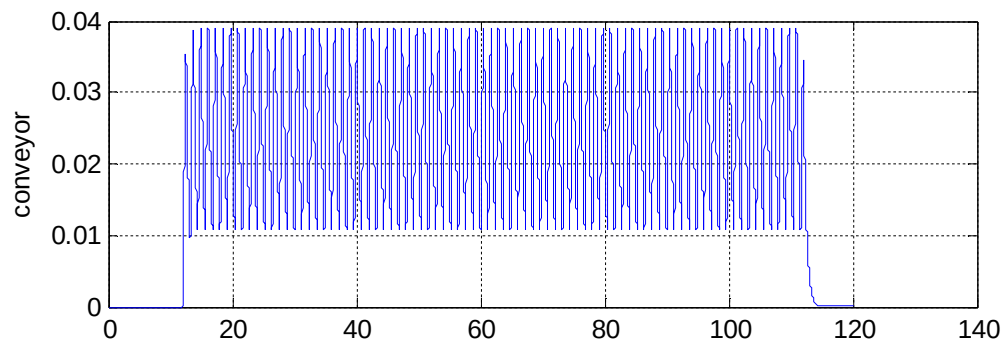
Kecepatan Motor = 300 rpm



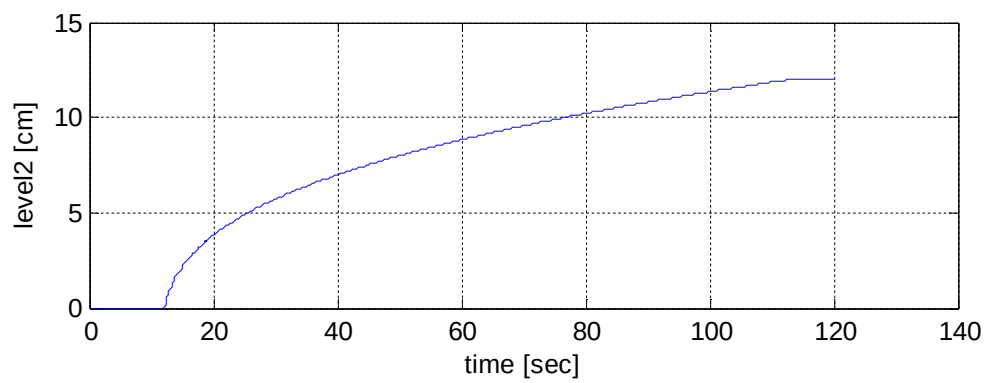
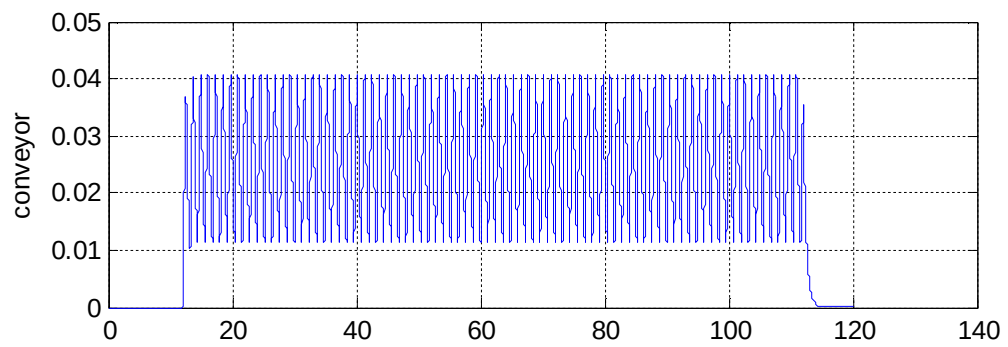
Kecepatan motor = 400 rpm



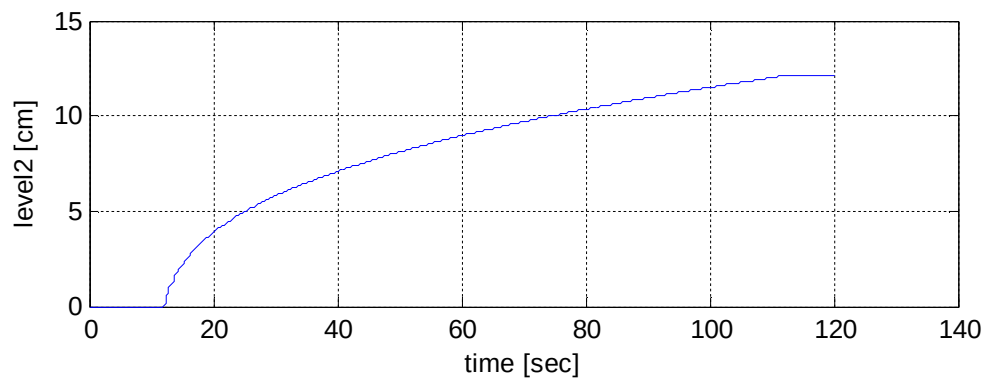
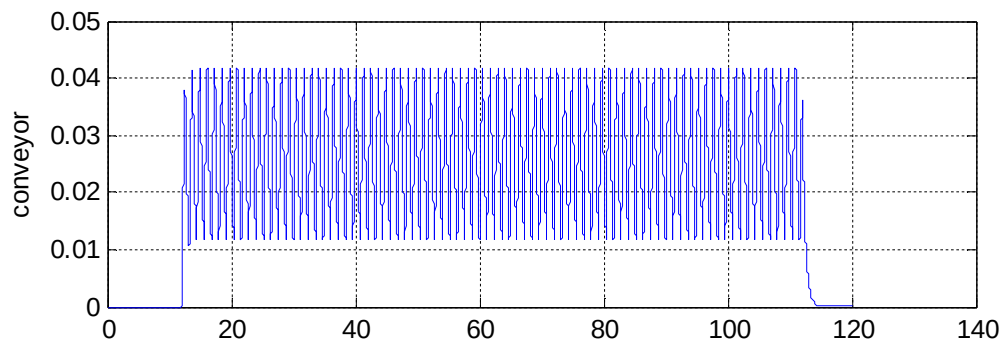
Amplitudo 2 cm



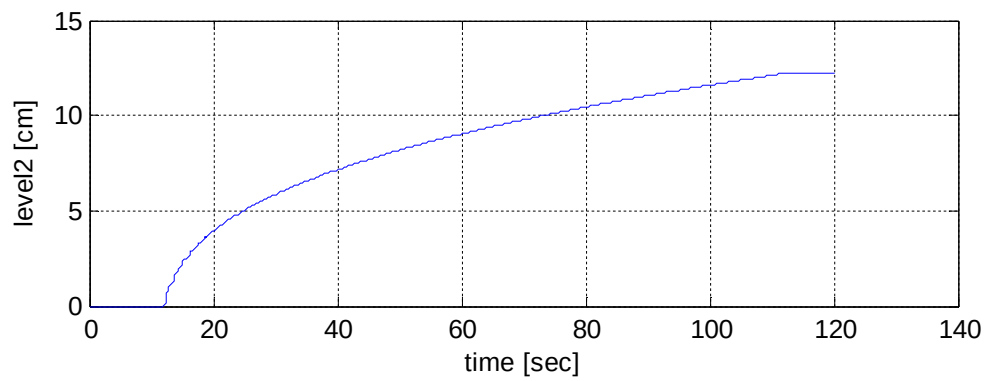
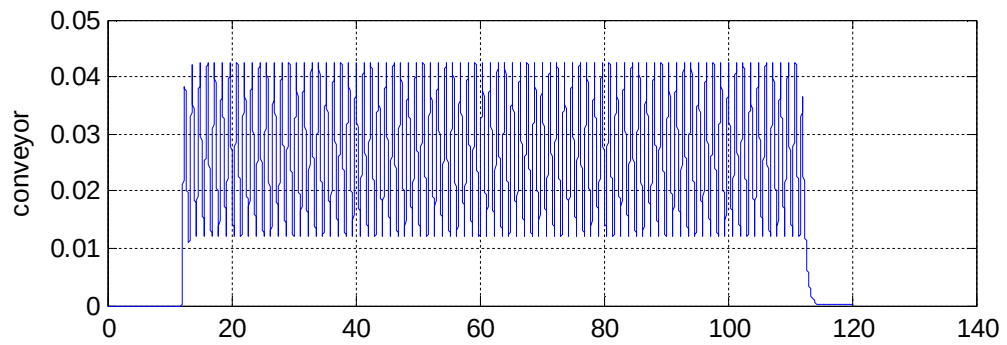
Amplitudo 2.5 cm



Amplitudo 3 cm



Amplitudo 3.5 cm



LAPORAN PERKULIAHAN TEKNOLOGI KENDALI PROSES



CHAIDIR ANWAR_D411 12 012

TEKNIK KOMPUTER KENDALI ELEKTRONIKA (TKKE)

FAKULTAS TEKNIK JURUSAN ELEKTRO

UNHAS

BAB I

PENDAHULUAN

1.1. LATAR BELAKANG

Dalam rangka memenuhi tuntutan zaman sebagai mahasiswa teknik elektro, mahasiswa kini perlu memiliki kemampuan untuk dapat menerapkan ilmu yang dipelajarinya kedalam bidang kerjanya masing-masing. Dengan berbagai perkembangan teknologi, tentu perlu pembelajaran yang lebih banyak. Bagi seorang mahasiswa teknik elektro dengan konsentrasi Teknik Kendali, perlulah mendapat pembelajaran lebih mengenai cara mengendalikan dan merancang sebuah sistem fisik kedalam bentuk matematis yang dapat dianalisis, dalam hal kaitannya dengan suatu sistem fisik yang melakukan suatu proses, baik berupa proses fisika maupun kimia. Untuk itu penting bagi kita dalam memahami cara-cara yang diperlukan untuk membuat perancangan suatu sistem fisik kedalam model yang mudah untuk dianalisis –dalam hal ini adalah kestabilannya dan respon keluarannya–.

1.2. TUJUAN

Mampu mengubah suatu model sistem fisik kedalam model yang dapat dianalisis, baik matematis maupun dalam bagan kotak, sehingga dapat dianalisis kestabilan dan respon keluarannya. Dalam hal ini akan dilakukan dengan bantuan software *Simulink MATLAB R2013a (8.1.0.604)*.

BAB II

PROYEK GUNT

Gunt merupakan sebuah perusahaan di Jerman yang membuat miniatur dari berbagai proses. Salah satunya adalah *level control*, *hsi*. Berikut adalah gambaran dari sistem fisik tersebut:



Level control merupakan sebuah proses yang berusaha menjaga kestabilan level air pada tangki penampungannya. Miniatur ini memiliki sebuah pompa untuk memompa air dari bawah ke dalam tangki penampungannya, sebuah katup proporsional untuk membuang kelebihan air dari yang diinginkan, serta sebuah sensor tekanan yang dapat mengukur ketinggian air di dalam tangki penampungan. Sistem ini juga dilengkapi dengan sebuah pipa pembuangan ketika terjadi kelebihan tampungan pada tangki. Berikut ini adalah data-data teknis yang dibutuhkan dalam perancangan model proses:

1. Level-controlled tank

- capacity: 1,2L

2. Storage tank

- capacity: 3,7L

3. Pump

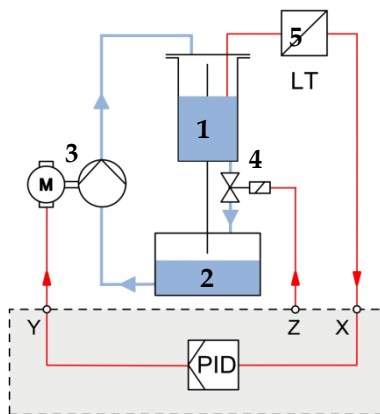
- power consumption: 18W

- max. flow rate: 8L/min

- max. head: 6m

4. Proportional valve: Kvs: 0,7m³/h

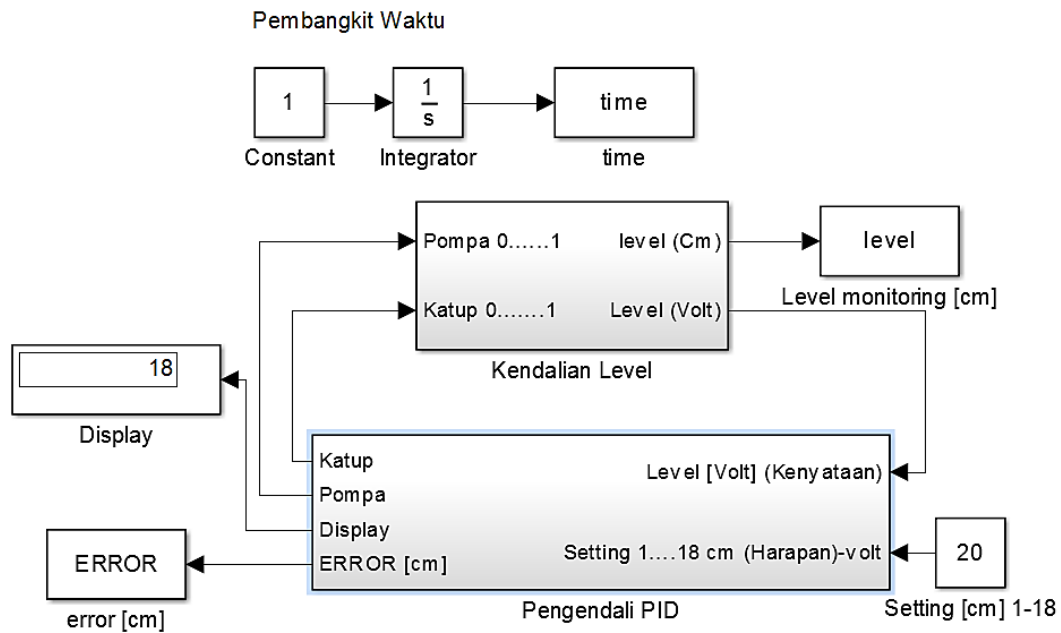
5. Pressure sensor: 0...30mbar (0...300mm)



Bagan skematik proses dari miniatur diperlihatkan pada gambar disamping. Sebagai informasi tambahan yang tidak diketahui, diasumsikan bahwa tinggi tangki adalah 20 cm dengan tinggi pipa pembuangan *overflow* adalah 18 cm.

Dari skematik proses diatas, dapat dilihat bahwa ada dua blok utama, yakni pengendali dan kendalian. Pengendali akan mendapat isyarat *setpoint*, yang berupa nilai pengaturan ketinggian air yang diinginkan. Pada pengendali, dibutuhkan dua buah masukan sebagai parameter acuannya, yaitu *setpoint* dan level ketinggian air yang telah dicapai, yang diambil dari pembacaan sensor yang sudah dalam satuan volt. Sebagai keluaran dari pengendali adalah isyarat kendalian untuk katup dan pompa. Keluaran lain dari pengendali disini sebagai tambahan adalah *display* yang akan menunjukkan hasil pembacaan dari *sensor* (lebih tepatnya dikatakan sebagai *transducer*) dalam satuan *centimeter*. Keluaran lainnya adalah galat (*error*) yang disimpan pada sebuah variabel “*ERROR*” (*disebut to workspace*) yang akan menyimpan kondisi galat setiap waktu selama selang waktu simulasi sehingga respon dari sistem dapat teramati. Untuk kemudahan, selanjutnya untuk dilakukan *monitoring* akan digunakan bentuk variabel seperti ini.

Sedangkan untuk kendalian, akan menerima masukan berupa isyarat kendalian dari pengendali, yaitu isyarat kendalian pompa katup. Satu-satunya keluaran dari kendalian adalah hasil dari pembacaan sensor ketinggian dalam satuan *volt*. Namun sebagai tambahan, dipasang sebuah variabel “*level*” –yang dimaksud adalah *to workspace*, bertipe *array*– untuk level ketinggian air pada tangki. Untuk kemudahan *monitoring* terhadap waktu, dibuatlah pembangkit waktu dengan persamaan $\int 1 dt = t$ untuk membantu respon sistem terhadap waktu (dengan nama variabel “*time*”). Berikut ini adalah model umum yang telah kita capai hingga saat ini, dengan Kendalian Level dan Pengendali PID berupa s.



1. KENDALIAN LEVEL

Seperti yang telah dijelaskan sebelumnya, Kendalian atau *Plant* memiliki dua masukan, yaitu isyarat kendalian untuk pompa serta katup, serta dua buah keluaran, yakni umpan balik atau *feedback* dari sensor serta untuk *monitoring* ketinggian air pada tangki.

1.1.POMPA

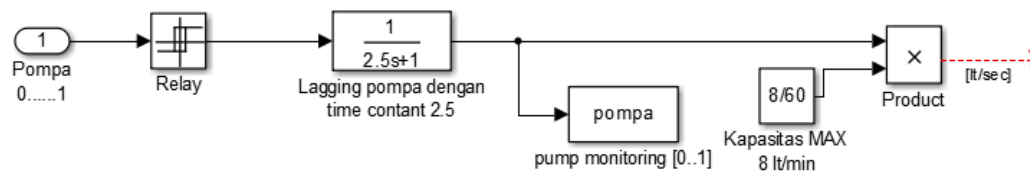
Pompa memiliki isyarat kendalian yang diasumsikan dari nol sampai satu [0..1], dimana nol adalah kondisi saat pompa dimatikan, sedangkan satu adalah saat pompa dinyalakan dengan kecepatan maksimum. Diketahui bahwa pompa yang digunakan memiliki kecepatan aliran maksimum 8 liter/menit. Untuk itu dibuat sebuah blok *constant* yang digunakan sebagai pembanding terhadap sinyal kendalian. Sebagai catatan, pompa juga dapat diberi isyarat kendalian kurang dari 0 karna sifat *hysteresis* dari pompa yang harus diberi isyarat kendalian kurang dari 0 untuk memmatikannya.

Karena pompa tentunya diputar oleh motor yang dikontrol oleh sebuah *relay*, maka pada pompa akan berlaku suatu model *hysteresis*, yakni kondisi dimana pompa tidak memiliki titik yang sama untuk on atau off dari keadaan sebelumnya. Untuk itu, dari input pompa dipasang sebuah blok *relay* yang memberikan efek *hysteresis* tersebut. Sebagai parameter pada blok relay, diberikan titik pergantian *on* pada 0.2, dan titik pergantian *off* pada -0.2, dengan keluaran adalah 1 saat *on* dan 0 saat *off*.

Setelah *on*, motor pompa tidak akan serta merta berputar dengan kecepatan maksimum sehingga aliran pompa maksimum tidak mungkin akan langsung tercapai. Untuk motor orde 1, motor pompa akan *lagging* terhadap respon masukannya dengan *time constant* (τ) tertentu. Untuk itu, digunakan sebuah blok *transfer function* untuk membuat *lagging* dengan *time constant* τ dengan persamaan $\frac{1}{\tau \cdot s + 1}$.

Keluaran dari blok *transfer function* ini dapat dipasang variabel *monitoring* “pompa” untuk mengetahui respon dari kecepatan motor pompa. Hasil dari *lagging* berupa kecepatan motor terhadap putaran maksimumnya[0..1], sehingga berdasarkan putaran motor maksimum, kita dapat membandingkan dengan kecepatan aliran maksimum –blok *constant* yang dibahas sebelumnya– untuk mendapatkan kecepatan aliran air sesaat yang dipompa ke dalam tangki.

Untuk membandingkan, kita dapat menggunakan sebuah blok *product* dengan keluaran berupa kecepatan aliran yang dipompa kedalam tangki dalam satuan *liter/detik*. Jika diasumsikan $\tau=2.5$, maka secara keseluruhan blok diagram pompa dapat digambarkan sebagai berikut:



1.2. KATUP

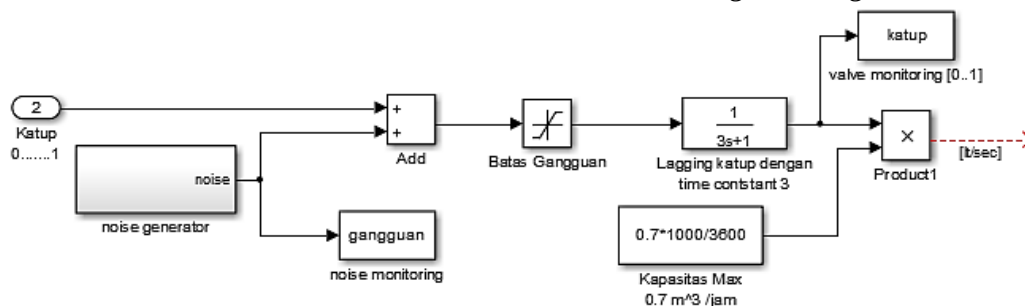
Katup yang digunakan adalah katup proporsional, artinya tidak hanya tertutup atau terbuka secara penuh, tapi memiliki *range*, seperti setengah tertutup, atau seperempat terbuka. Untuk itu, maka isyarat kendalian dapat berupa nilai-nilai dari nol sampai satu, dengan asumsi bahwa 0 adalah saat pompa tertutup penuh dan 1 saat pompa terbuka penuh. Sebagai catatan, katup tidak akan diberikan keadaan isyarat kendalian kurang dari 0 atau lebih dari 1.

Katup akan beroperasi dengan rentang isyarat kendalian dari 0 sampai 1. Oleh sebab itu, isyarat kendalian diberi blok *saturation*, sehingga memberi batasan apabila ada gangguan yang muncul dan menyebabkan isyarat kendalian yang diterimanya bernilai tidak pada rentang tersebut.

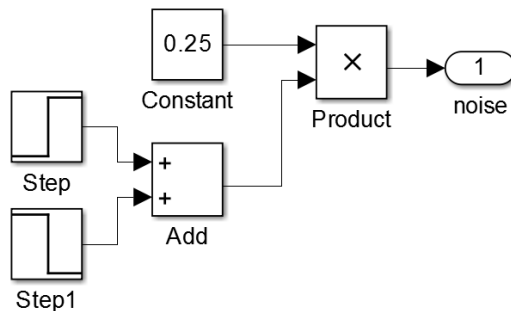
Seperti pada motor, katup juga tidak dapat langsung menutup dan membuka sekat berdasarkan isyarat kendaliannya, melainkan akan mengalami *lagging*. Oleh sebab itu, setelah blok *saturation* kembali dipasang sebuah blok *transfer function* yang memberikan sifat *lagging* tersebut pada katup. Setelah blok ini akan diberikan sebuah variabel *monitoring* dengan nama “katup” untuk memantau status dari katup.

Karena katup dapat membuang fluida dengan debit maksimum $0.7 \text{ m}^3/\text{hours}$, dan isyarat kendalian yang telah keluar dari blok *lagging* akan menghasilkan isyarat dari 0 sampai 1, maka kita dapat kembali membandingkan besarnya aliran yang keluar dari katup sesaat terhadap debit maksimumnya dengan kembali menggunakan blok *product*. Tentu saja dengan mengkonversi satuan dari debit maksimum dalam *liter/sec* terlebih dahulu. Keluarannya berupa debit aliran air yang dibuang dari tangki melalui katup dalam satuan *liter/second*.

Sebagai tambahan, sebelum melewati blok *saturation*, isyarat kendalian dapat diberikan gangguan untuk melihat respon sistem apabila terdapat gangguan. Akan ada sebuah yang dibuat dapat menghasilkan gangguan pada waktu tertentu sesuai keinginan, yang kemudian akan ditambahkan dengan blok *add* sehingga menghasilkan suatu isyarat kendalian yang terganggu. penghasil gangguan dapat dipantau dengan memasang variabel *monitoring* “gangguan”. Sehingga apabila keseluruhan sistem digambarkan, dengan *time constant* diasumsikan bernilai 3 detik, akan membentuk susunan blok diagram sebagai berikut:



Untuk *noise generator*, yakni penghasil gangguan, misalnya dapat dibuat dengan blok diagram di bawah ini. Tujuannya adalah menghasilkan gangguan berbentuk pulsa yang dapat diaktifkan selama waktu tertentu, misalnya dari 100 sampai 150 detik dengan amplitudo 0.25.

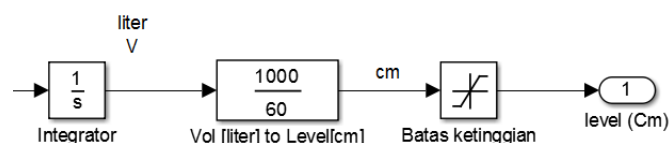


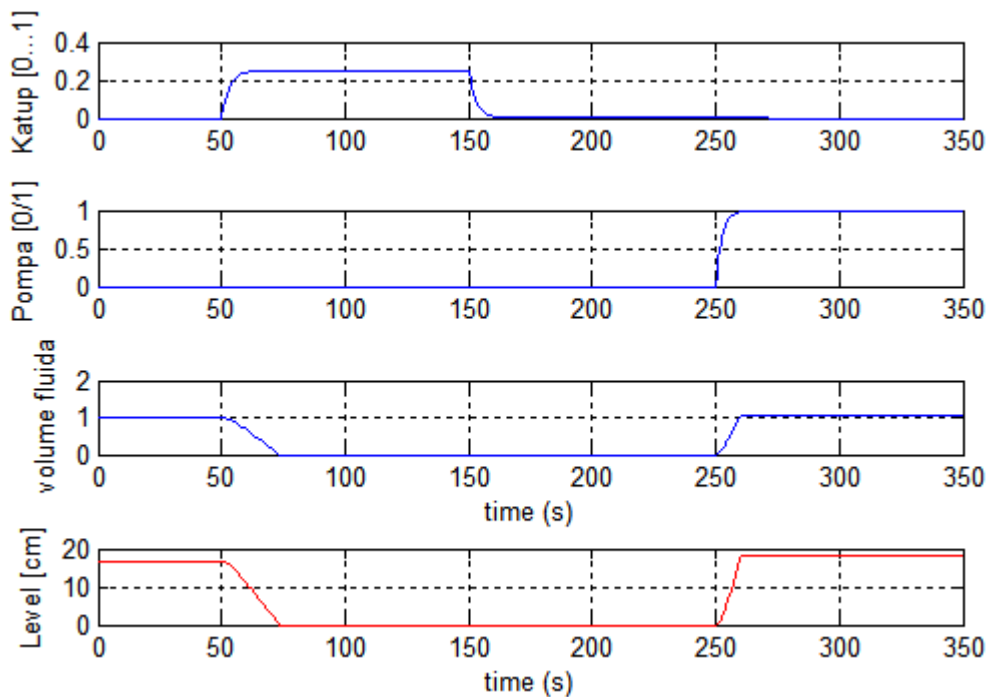
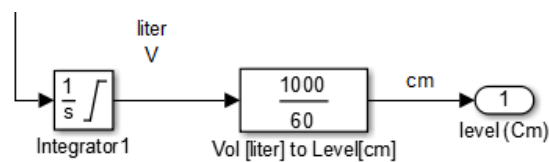
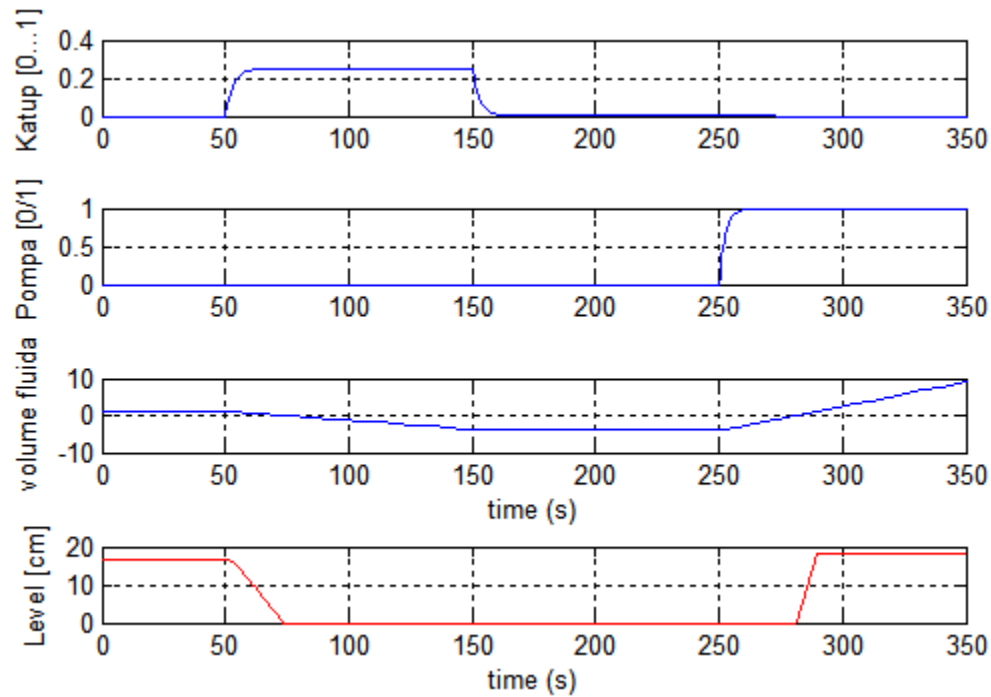
1.3. LEVEL CONTROLLED TANK

Level controlled tank merupakan tangki penampungan yang akan diatur ketinggiannya. Tangki ini akan mendapat tambahan fluida dari pompa, dan kelebihan fluida dapat dibuang melalui katup. Karena pompa bersifat menaikkan level fluida pada tangki, dan katup bersifat mengurangi aliran level fluida, maka aliran sesaat pada tangki adalah kecepatan aliran pada pompa dikurangi kecepatan aliran pada katup. Untuk itu dipasanglah suatu blok yang mengurangi aliran pompa terhadap aliran katup tersebut, yakni blok *sum*.

Hasil dari penjumlahan ini memiliki satuan *liter/detik* –turunan volume terhadap waktu–. Sehingga untuk mendapatkan volume sesaat pada tangki maka dilakukan pengintegralan sehingga menghasilkan satuan volume. Dengan asumsi luas penampang tangki adalah 60 cm^2 , dengan tinggi pipa pembuangan *overflow* 18 cm –18.1 cm untuk toleransi 0.1cm– maka batas dari volume ini adalah dari 0 sampai 1.086 liter. Volume merupakan integrasi dari kecepatan aliran fluida setiap waktu. Untuk itu perlulah menggunakan suatu blok *integrator* yang dilengkapi *saturation* agar volume yang ada di dalam tangki tidak bernilai di luar rentang ini. Mengapa tidak menggunakan blok *satursasi* saja setelah diubah menjadi satuan cm agar lebih mudah perhitungannya? Sebab, apabila dilakukan setelah blok *integrator*, maka pada kondisi katup dibuka maksimum tanpa tambahan fluida dari pompa, hasil *integrator* tetap akan bernilai negatif walaupun pada *monitoring* nilai ini tetap terbaca sebagai level 0cm akibat pemasangan blok *saturation* setelah *integrator*. Namun, begitu pompa dinyalakan, level air tidak akan langsung naik, karena pompa perlu kembali menolak volume fluida yang bernilai negatif tersebut. Itulah mengapa pembatasan volume fluida dilakukan langsung pada blok *integrator*, sehingga apapun yang terjadi, volume fluida akan bernilai kurang dari nol. Mari membandingkan grafik respon sistem terhadap perbedaan peletakan *saturation*.

Pada blok diagram yang menggunakan *saturation* setelah blok *integrator*, volume akan tetap bernilai negatif, dan level air akan mulai bertambah saat volume ini telah melewati titik nol. Berbeda dengan blok *integrator* yang telah dilengkapi dengan batas (saturasi) untuk volumenya. Saat pompa dinyalakan, maka saat itu juga level ketinggian fluida di dalam tangki akan naik. Secara fisik, model yang kedua inilah yang masuk akal. Blok *integrator with saturation* akan diberikan batas bawah 0 dan batas atas 1.086. Pada blok ini juga dapat diisi *initial condition*, yakni kondisi awal dari tangki dalam satuan liter (misalnya 1).



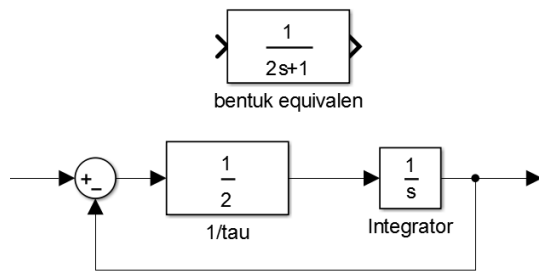


Karena keluaran yang akan dimonitoring dari ini adalah level ketinggian tangki dalam satuan cm, perlu dilakukan perubahan dari volume menjadi ketinggian. Ketinggian dalam cm adalah volume dalam cm^3 dibagi luas penampangnya dalam cm^2 . Untuk itu digunakan sebuah blok *transfer function* yang mengkonversi nilai ini sebagai keluaran. Sebuah *sensor* akan dipasang pula pada keluaran ini untuk mengawasi level ketinggian fluida dalam cm sehingga menjadi

level ketinggian dalam *volt*.

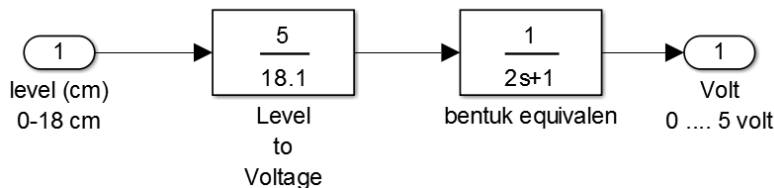
1.4. SENSOR

sensor bertujuan untuk membaca ketinggian air dalam cm, dan memberikan keluaran dalam satuan *volt*. Untuk ketinggian maksimum adalah 18.1 cm, dan akan dikonversi dalam satuan *volt* dari 0 sampai 5, maka dari satuan cm dapat diubah ke *volt* dengan mengalikannya dengan sebuah blok *transfer function* $\frac{5}{18.1}$.

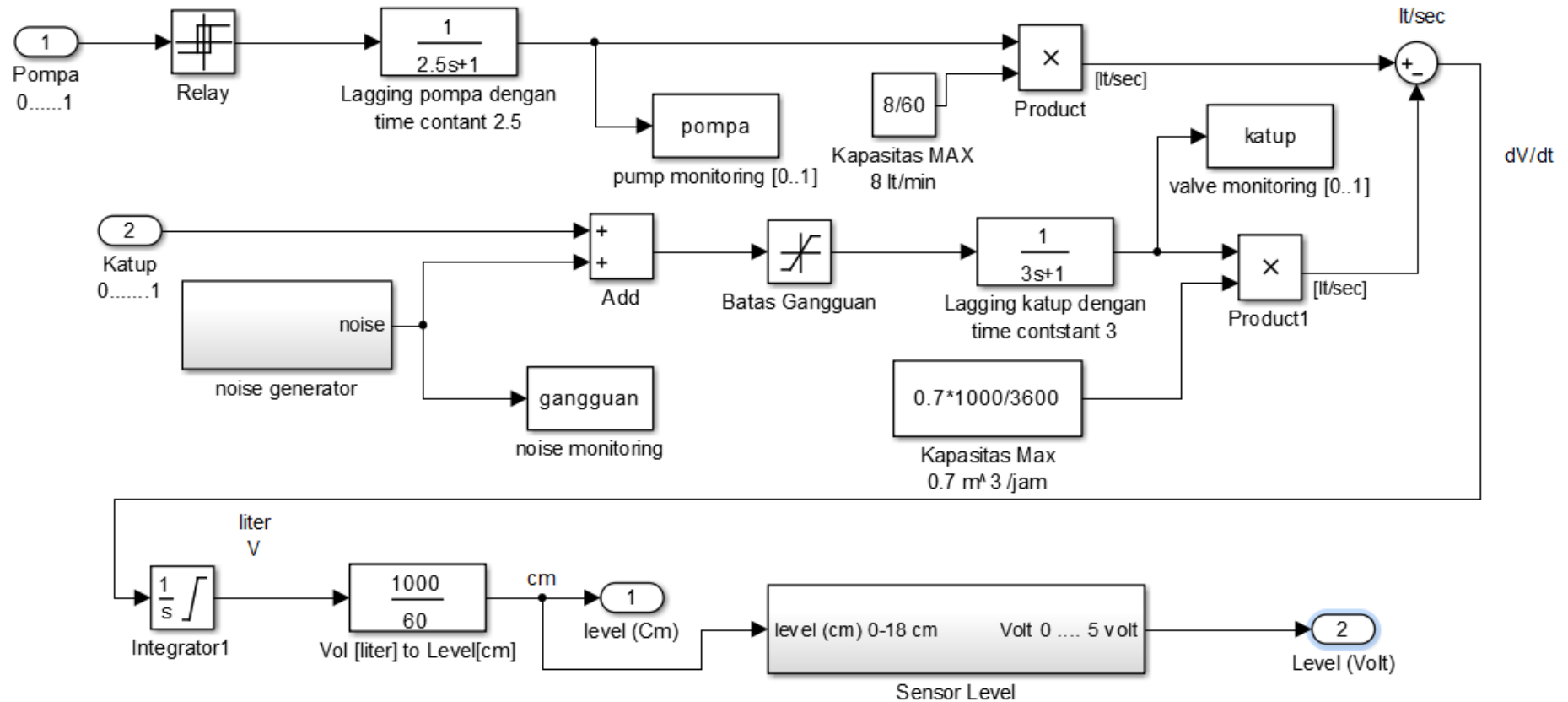


Namun, seperti sensor pada umumnya, tentu pembacaan akan mengalami *lagging*. Misalnya saja dengan *time constant* 2, maka blok *lagging* dapat berbentuk seperti sebelumnya (gambar atas), atau dengan penjabaran (gambar bawah). Kelebihan dari gambar di bawah, kita dapat menentukan nilai awal pada blok *integrator*. Yang perlu diketahui adalah penambahan nilai awal pada sensor akan merusak hasil pembacaan sensor, dan dapat menyebabkan masalah kemudian. Misalnys saja

tidak akan pernah di dapatkan ketinggian fluida yang tepat jika dilihat dari pembacaan sensornya, sehingga output akan terus berosilasi atau ketinggian air tidak akan sesuai dengan *setpoint* yang diberikan. Untuk itu, pemberian nilai awal pada sensor tidak perlulah diberikan, karena sensor akan membaca nilai awal dari ketinggian fluida dengan sendirinya, walaupun tanpa memberikan nilai awal yang sama pada sensor. Secara lengkap, blok diagram sensor ditunjukkan pada gambar berikut:



Secara lengkap, blok kendalian level diberikan pada gambar di bawah ini:



2. PENGENDALI PID

Pengendali PID adalah yang memberikan isyarat kendalian pada kendalian agar level ketinggian fluida pada tangki tetap stabil sesuai dengan nilai *setpoint* yang diberikan. Masukan dari pengendali ada dua, yakni *setpoint* atau harapan dan hasil pembacaan sensor dalam *volt*, atau kenyataan. Sedangkan galat atau *error* adalah selisih antara harapan dan kenyataan.

Yang pertama adalah *display* yang menunjukkan hasil pembacaan sensor dalam satuan *centimeter*. Karena rentang keluaran sensor adalah 0 sampai 5 volt untuk pembacaan 0 sampai 18.1 cm, perlu diberikan blok pembalik dari volt ke cm, yakni *transfer function* dengan nilai $\frac{18.1}{5}$, sehingga dari blok ini akan langsung didapatkan level ketinggian fluida dalam satuan cm.

Karena hasil pembacaan sensor adalah dalam *volt*, maka untuk mencari galat, perlu dilakukan konversi *setpoint* terlebih menjadi satuan yang sama, *volt*. Namun, sebelum dilakukan konversi, perlu dilakukan pembatasan pada *setpoint*, agar nantinya *setpoint* berlebih, misalnya lebih besar dari tinggi pipa pembuangan *overflow*, maka akan mengakibatkan pompa terus *on* karena tinggi fluida pada tangki tidak akan mungkin melebihi tinggi pipa pembuangan *overflow* tersebut. Untuk itu, dari masukan *setpoint* dipasang sebuah blok *saturation* yang nilainya dibuat dari 0 sampai 18.1 cm.

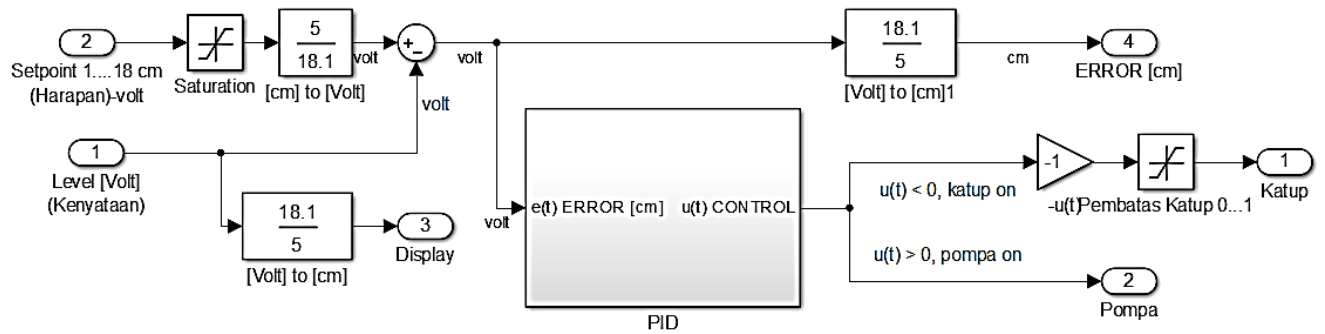
Barulah setelah itu sebuah blok *transfer function cm-to-volt* $\left(\frac{5}{18.1}\right)$ diletakkan.

Sekarang, keduanya –harapan dan kenyataan– telah dalam satuan yang sama. Barulah *error(volt)* bisa didapatkan dengan mengurangkan antara harapan dan kenyataan. Jika hasil bernilai negatif, artinya ketinggian fluida melebihi harapan sehingga katup perlu dibuka, dan apabila bernilai positif, artinya ketinggian fluida kurang dari harapan, sehingga pompa yang perlu di jalankan. Untuk memantau *error* dalam *centimeter* sebagai salah satu keluaran dari pengendali, kembali setelah *error* dalam *volt* didapatkan, diletakkan sebuah blok *transfer function volt-to-cm* $\left(\frac{18.1}{5}\right)$ yang keluarannya sudah merupakan *error* dalam satuan *centimeter*.

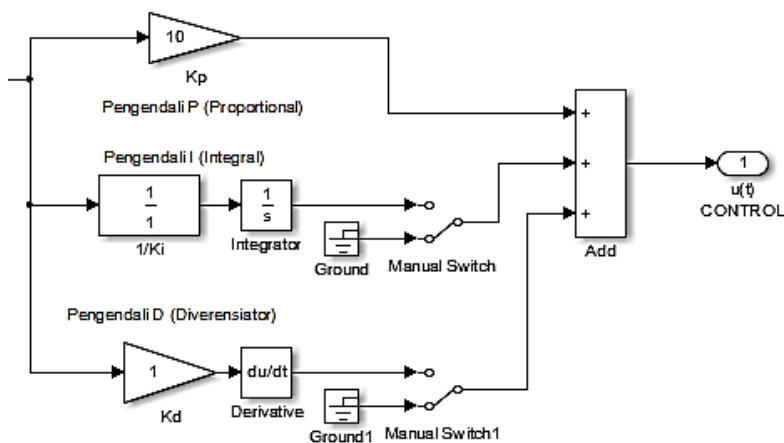
Pengendalian pompa dan katup dilakukan dengan menambahkan sebuah *PID* yang berisi pengendali PID berdasarkan *error* yang diterimanya. Oleh sebab itu, keluaran dari pengendali apabila bernilai positif akan dibuat untuk menggerakkan pompa dan apabila bernilai negatif akan dibuat untuk menggerakkan katup. Untuk itu dari keluaran *PID* langsung disambungkan ke keluaran yang menuju pompa, sehingga keluaran *PID* tersebut akan langsung mengendalikan pompa. Sedangkan untuk katup saat keluaran *PID* bernilai negatif, perlu dilakukan pembalikan dan pembatasan terlebih dahulu. Dilakukan pembalikan karena yang mengaktifkan katup adalah keluaran negatif dari *PID*, sedangkan katup akan menerima isyarat kendalian dari 0 sampai 1 (positif). Untuk itulah keluaran *PID* dilakukan pembalikan terlebih dahulu. Sedangkan dilakukan pembatasan dari 0 sampai 1 dilakukan karena agar isyarat kendalian tidak bernilai negatif ataupun bernilai diatas 1. Namun, untuk membuat katup tidak terlalu rentan terhadap gangguan, batasan atas yang dimasukkan bukanlah 1, melainkan 5-10. Batasan atas juga tidak boleh terlalu tinggi, sebab efek *lagging* pada katup dapat menyimpan isyarat yang terlalu besar sehingga dapat menyebabkan sistem malah menjadi lebih tidak stabil. Pembatasan diatas 1 ini tidaklah menjadi masalah, sebab pada kendalian sendiri sudah diberikan batasan sehingga isyarat yang diterimanya tidak melewati rentang 0..1.

Lalu, mengapa pompa tidak diberikan pembatasan yang sama? Pompa memiliki sifat *hysteresis*, sehingga butuh nilai isyarat kendalian negatif yang cukup untuk mematikan pompa. Jika diberikan batasan yang sama pada pompa, isyarat kendalian pompa tidak akan pernah bernilai negatif, sehingga pompa akan terus-menerus *on*. Inilah sebab mengapa pompa tidak diberikan batasan isyarat kendalian.

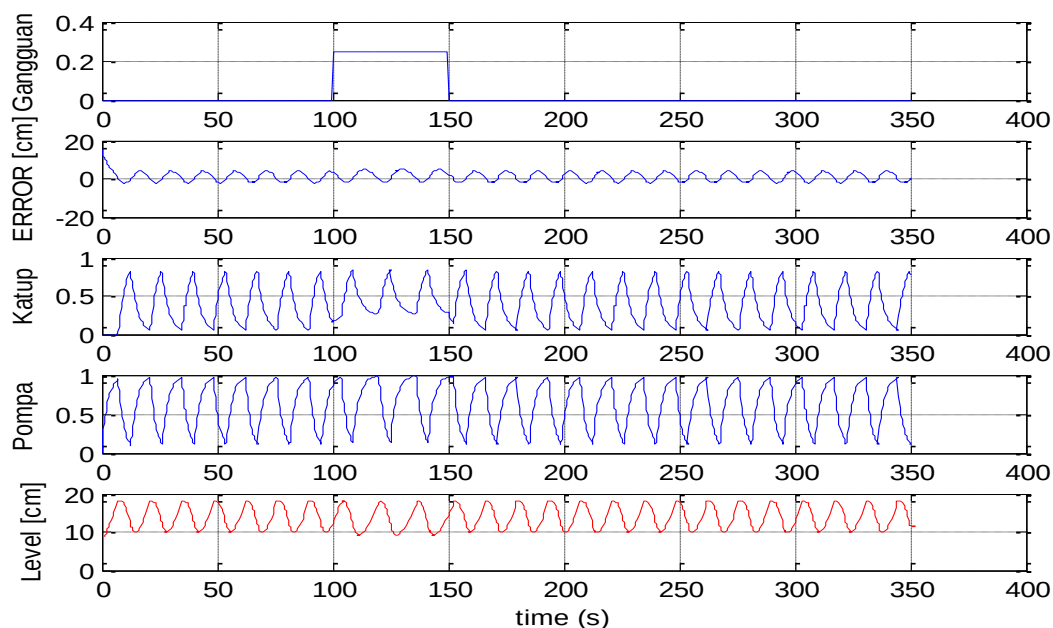
Berikut ini adalah blok diagram dari *PID* secara lengkap:



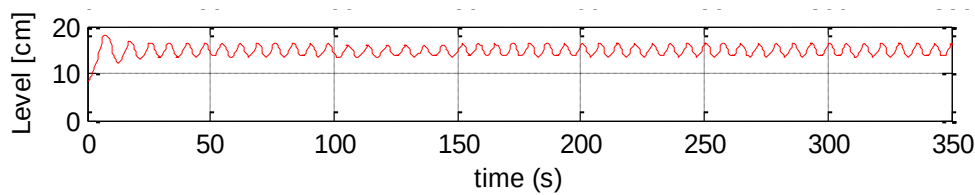
Selanjutnya adalah *PID*. Pendengali *PID* terdiri atas tiga buah pengendali; pengendali proportional (*P*), pengendali Differensial (*D*) dan pengendali Integral (*I*). Pengendali proportional bersifat memperbesar error sehingga akan lebih cepat dilakukan koreksi. Pengendali differensial bersifat meredam osilasi pada respon sistem. Sedangkan pengendali integral bersifat mempercepat respon sistem mencapai keadaan stabil, namun bersifat memberikan efek osilasi pada respon sistem tersebut. Berikut ini adalah blok diagram pengendali *PID* secara umum:



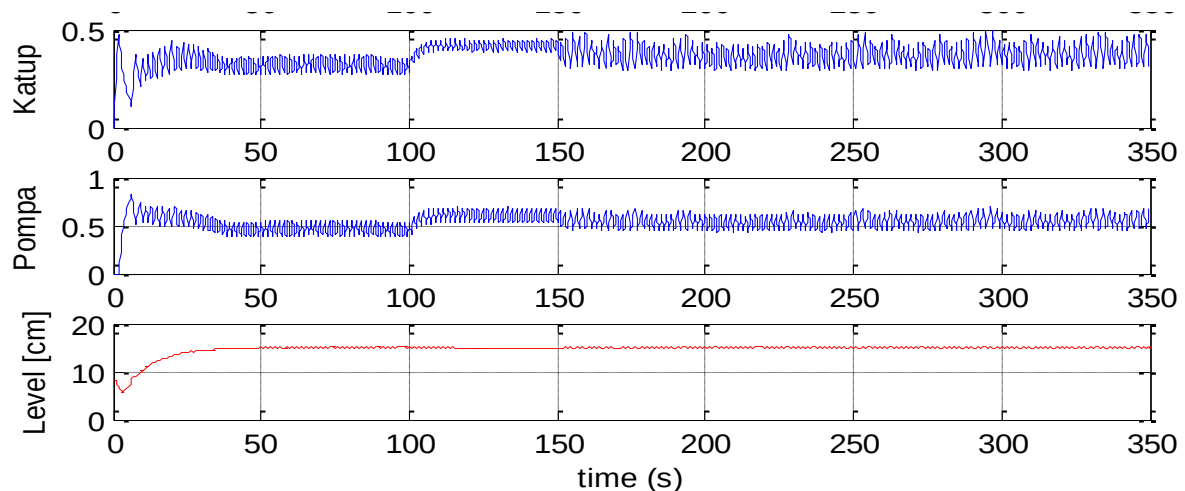
Terlihat bahwa keluaran pengendali *PID* $u(t)$ merupakan akumulasi dari pengendali *P*, *I* dan *D*. Grakfik berikut akan menunjukkan bagaimana respon sistem (ketinggian fluida) dengan hanya menggunakan pengendali proportional dengan $K_p=10$ terhadap *setpoint* 15 cm (dengan volume awal 0.5):



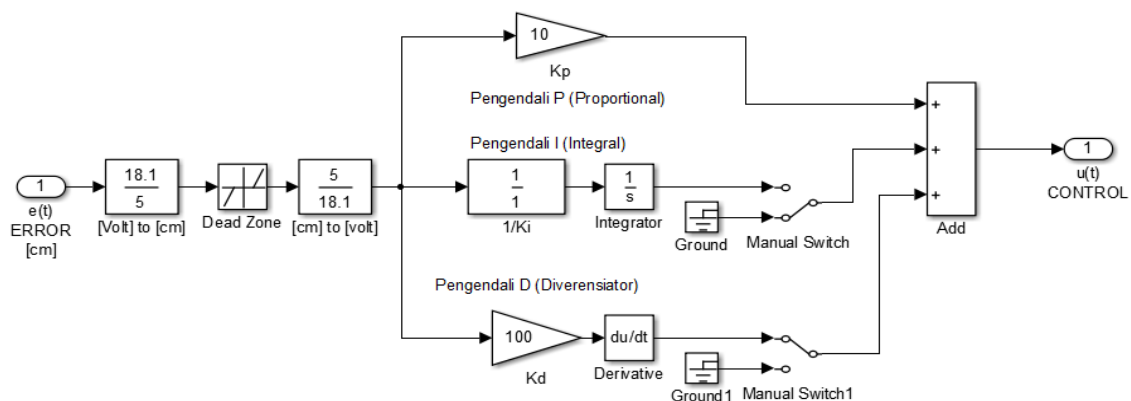
Dari respon sistem terlihat bahwa level ketinggian fluida hanya berosilasi di sekitar *setpoint*. Untuk meredam osilasi maka dipasanglah kombinasi pengendali PD ($K_d=10$), sehingga respon sistem menjadi seperti grafik di bawah ini:

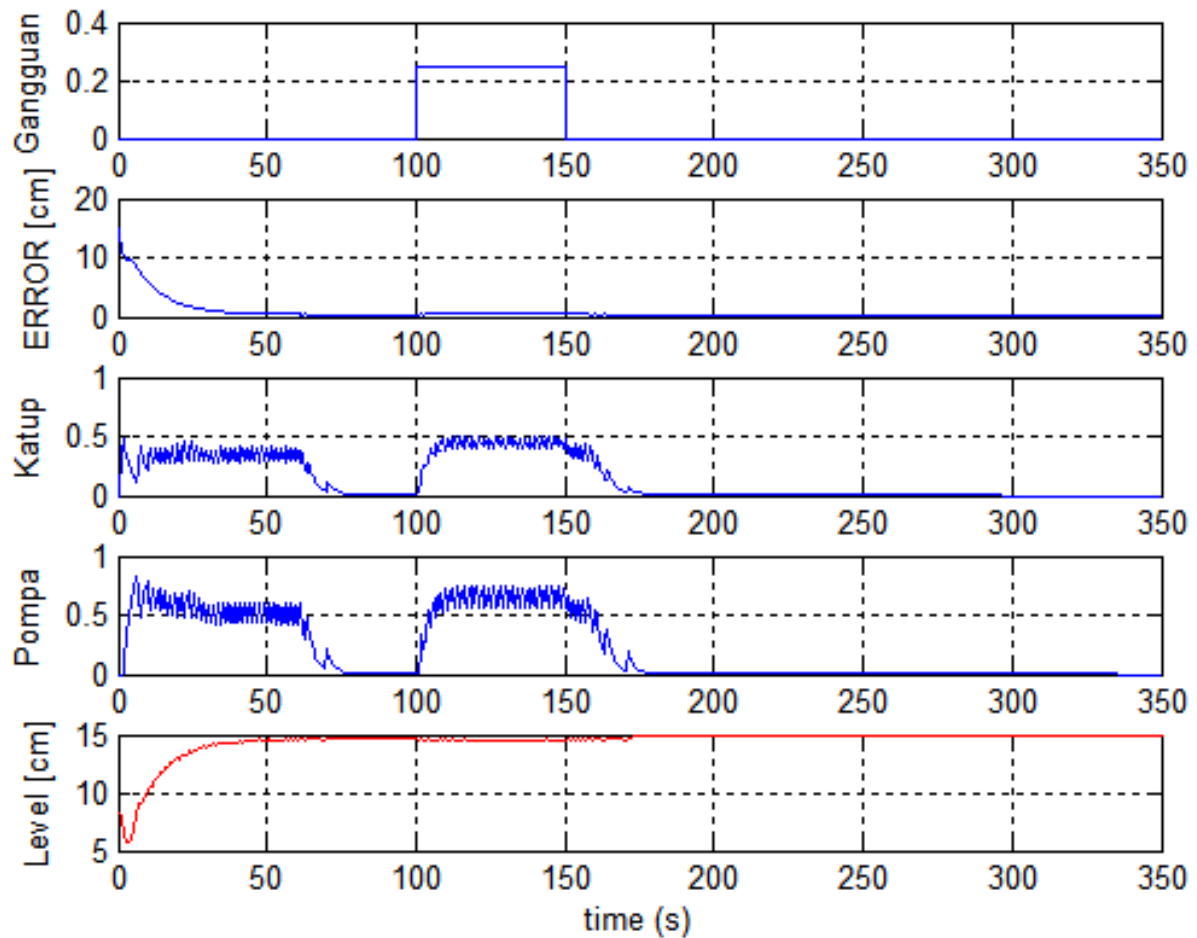


Jelas terlihat bahwa osilasi yang dihasilkan sudah cukup teredam. Untuk lebih meredam respon level ketinggian fluida pada tangki, kita hanya perlu menaikkan nilai koefisien differensial (K_d) menjadi cukup besar untuk meredam osilasi yang terjadi, misalnya $K_d=50-100$. Berikut adalah contoh respon sistem dengan nilai $K_d=100$:



Sekarang, respon output sudah sangat stabil. Masalah berikutnya yang muncul adalah walaupun telah mencapai ketinggian fluida yang sangat mendekati *setpoint*, katup dan pompa terus bekerja selama ada *error* yang tercipta, walaupun nilainya sangat kecil. Ini bisa diatasi dengan memberikan koreksi *error*, berupa toleransi error yang masih dapat diterima. Misalnya untuk toleransi *error* yang baik untuk kondisi *hysteresis* relay dari -0.2 s/d 0.2 adalah ± 0.4 cm. Sedangkan untuk kondisi *hysteresis* relay dari -0.1 s/d 0.1 toleransi ± 0.1 cm sudah cukup baik. Berikut ini blok diagram lengkap dari PID beserta grafik respon sistem terhadap kondisi *hysteresis* relay dari -0.2 s/d 0.2 dengan toleransi ± 0.4 cm:

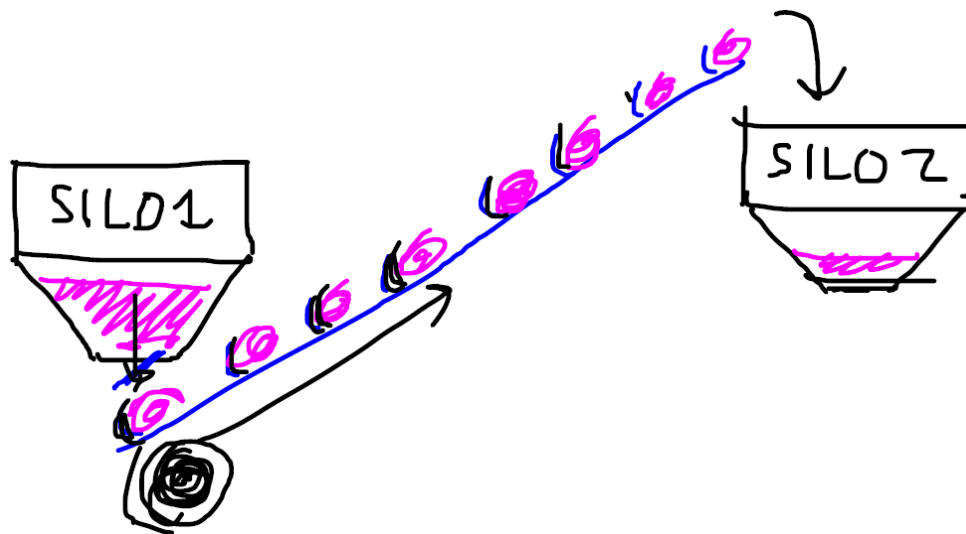




Respon sistem seperti inilah yang dicari. Level ketinggian fluida dalam tangki akan tetap stabil (dengan toleransi *error* tertentu) walaupun ada gangguan yang muncul pada katup. Sedangkan apabila telah berada pada level ketinggian fluida yang ditentukan pada *setpoint*, maka katup dan pompa akan berhenti aktif, dan mencegah pompa serta katup beroperasi secara berlebihan.

BAB III

PROYEK SILO-TO-SILO



Proyek ini adalah membuat suatu model proses untuk memindahkan benda berupa pasir antara dua tempat penampungan (disebut silo) menggunakan –lebih tepat disebut *elevator*– . ini dilengkapi cawan-cawan sebagai penahan tiap gundukan pasir. Silo yang digunakan berbentuk tabung pada bagian atas, dan kerucut terpancung terbalik pada bagian bawah. Ada dua silo yang digunakan, disebut Silo1 dan Silo2. Konfigurasi instalasi dapat dilihat pada gambar di atas.

Silo1 adalah tempat tampungan awal pasir. Pada bagian bawah Silo1 terdapat sebuah katup proporsional. Mulut katup akan langsung mengarah ke sebuah *conveyor*. *Conveyor* ini dilengkapi dengan cawan-cawan yang akan menahan gundukan pasir yang jatuh dari katup sehingga tidak jatuh selama perjalanan. *Conveyor* akan digerakkan oleh motor –dipisahkan oleh roda gigi–, sehingga cawan tersebut akan membawa setiap gundukan ke bagian ujung lainnya dari *conveyor* untuk ditumpahkan pada Silo2. Silo2 inilah yang akan kembali menampung setiap gundukan pasir yang telah dikeluarkan oleh Silo1. Pada Silo2, mulut katup akan tetap dibuat tertutup.

Berikut ini adalah data-data teknis yang dibutuhkan dalam perancangan model proses:

1. Silo1

-**Katup** (proporsional; debit maks 1liter/20detik)

-**Penampung**

Diameter tabung 26 cm

Tinggi tabung 20.5 cm

Diameter atas kerucut terpancung terbalik 26 cm

Diameter bawah kerucut terpancung terbalik 4.5 cm

Tinggi kerucut terpancung terbalik 12 cm

2. Silo2

-**Katup** (proporsional; debit maks 1liter/20detik)

-**Penampung**

Diameter tabung 26 cm

Tinggi tabung 10.5 cm

Diameter atas kerucut terpancung terbalik 26 cm

Diameter bawah kerucut terpancung terbalik 4.5 cm
Tinggi kerucut terpancung terbalik 12 cm

3. Conveyor

Kecepatan Motor 250rpm, (*adjustable*)

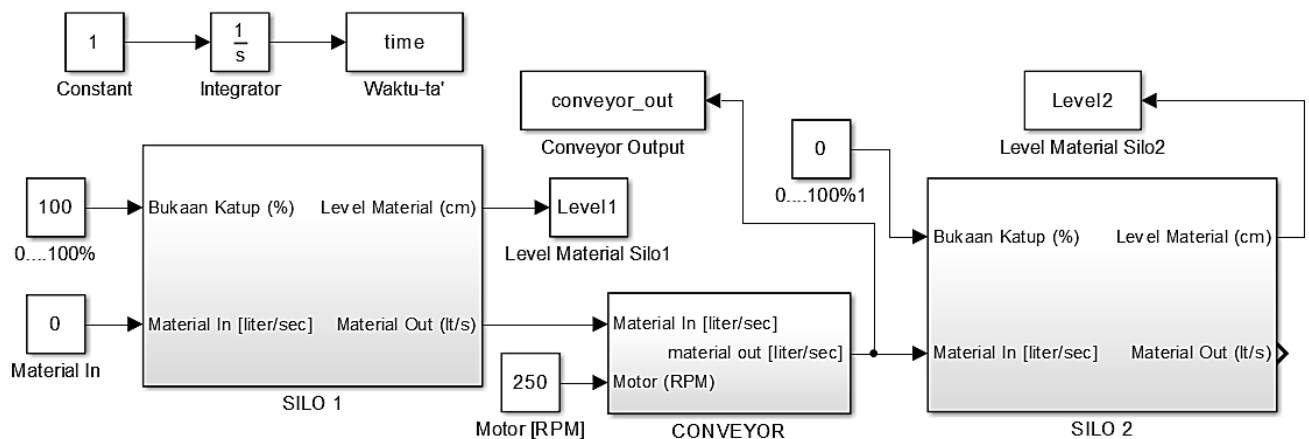
Roda gigi (kecepatan *conveyor*) 1:5 motor (artinya sebanyak 250/5 cawan ditumpahkan setiap menit)

Jumlah cawan di atas *conveyor* 10 cawan

Dari model di atas jelas terlihat bahwa ada tiga *subsystem* yang harus dibangun, Silo1, *conveyor* dan Silo2. Silo1 akan dapat diisi dengan material dari luar. Selain itu, perlu pula diatur seberapa besar katup pada Silo1 akan dibuka. Sebagai keluaran dari silo1, adalah material yang keluar dari katup. Untuk mengamati level material (dalam cm) pada Silo1, dapat pula ditambahkan sebuah variabel *monitoring* “*Level1*” pada keluaran lain dari Silo1.

Material keluaran dari Silo1 akan jatuh pada *conveyor* dan membentuk gundukan-gundukan pasir yang berjalan di atasnya. Selain banyaknya pasir dari Silo1 yang menjadi masukan dari *conveyor*, kecepatan motor perlu pula diatur untuk menentukan seberapa cepat gundukan material akan dibawa dan ditumpahkan ke Silo2. Untuk itu pada *conveyor* dibuat dua parameter masukan, yaitu *material_in* dan kecepatan motor untuk *conveyor*. Sedangkan hanya akan dibuat keluaran tunggal untuk *conveyor*, yaitu *material_out*, yakni berupa gundukan-gundukan pasir yang dibawa dan dijatuhkan pada Silo2. Akan dipasang sebuah variabel *monitoring* dengan nama “*_out*” untuk *memonitoring* bentuk keluaran dari *conveyor*.

Sedangkan untuk Silo2, secara blok diagram akan sama saja dengan Silo1, hanya saja akan ada sedikit perbedaan terkait dengan parameter-parameter saturasinya. Seperti halnya Silo1, pada Silo2 juga akan ditambahkan sebuah variabel *monitoring* dengan nama “*Level2*” untuk memantau ketinggian sesaat pasir pada Silo2. Sebagai catatan, katup pada Silo2 ini akan selalu dibuat tertutup, agar tidak ada material yang terbuang dengan percuma. Secara lengkap, model umum yang telah kita capai saat ini ditunjukkan pada gambar berikut:



1. SILO1

Telah dibahas sebelumnya bahwa bukaan katup merupakan salah satu parameter masukan dari Silo1. Selain bukaan katup, *Material_In* merupakan parameter masukan lainnya. Jika bukaan katup dinyatakan dalam persen[0..100], maka nilai bukaan ini perlu terlebih dahulu dikonversi ke nilai yang sebenarnya, yaitu dengan membagi nilai bukaan tersebut dengan 100. Untuk membagi, kita dapat menggunakan sebuah *transfer function*. Dari data teknis diketahui bahwa katup memiliki debit 1 liter / 20 detik untuk bukaan penuh, atau setara dengan 0.05liter/detik. Seperti biasanya, kita akan membandingkan nilai bukaan katup terhadap nilai maksimum untuk mendapatkan debit yang sebanding dengan bukaan katup tersebut. Untuk membandingkan, kita dapat kembali menggunakan sebuah blok *product*. Hasil dari membandingkan dengan blok ini akan memiliki keluaran berupa debit keluarnya material dari Silo1 dalam liter-per-detik, sebanding dengan nilai bukaan katup yang diberikan.

Jika *material_in* dalam liter/detik adalah debit masuknya material baru pada Silo, sedangkan bukaan katup menunjukkan debit keluarnya material dari Silo, maka debit pertambahan material pada Silo adalah *material_in* dikurangi debit bukaan katup. Untuk itu, dari parameter masukan *material_in* pada Silo langsung dapat dikurangkan dengan debit bukaan katup dalam satuan yang sama. Digunakan sebuah blok *sum*. Sekarang kita telah mendapatkan debit pertambahan material rata-rata sesaat dalam liter/detik pada Silo.

Seperti teknik pada proyek sebelumnya, jika debit adalah turunan volume terhadap waktu, maka volume merupakan integral dari debit terhadap waktu. Untuk itu, keluaran dari blok *sum* akan dimasukkan pada sebuah blok *integrator* untuk mendapatkan volume dari Silo dalam liter. Blok *integrator* yang digunakan adalah blok yang dilengkapi *saturation*. Alasannya masih sama, agar volume pada Silo tidak akan pernah bernilai negatif, sekalipun dilakukan pembatasan tinggi material pada bagian akhir. Penjelasan lebih rinci tentang blok ini dapat dilihat pada pembahasan sebelumnya (halaman 6 sub-bab 1.3. *level controlled tank*).

Sebagai referensi, untuk kerucut terpancung berlaku rumus $V = \frac{1}{3}\pi h(R^2 + Rr + r^2)$, dengan h adalah tinggi kerucut terpancung, R jari-jari tutup besar, dan r jari-jari tutup kecil. Dengan rumus ini ditambah dengan rumus volume tabung, didapatkan bahwa volume maksimum dari Silo1 adalah 13.4389 liter. Untuk itu pada batasan blok *integrator* yang digunakan diberi batasan saturasi dari 0 sampai 13.4389. Sedangkan *initial condition* merupakan volume awal dari Silo1 (misalnya 5 liter).

Variabel *monitoring* “*Level1*” yang dipasang bertujuan untuk memantau level ketinggian material pada Silo1 dalam satuan cm. Untuk itu perlu dilakukan konversi dari volume material dalam liter menjadi ketinggian material dalam cm. Dalam melakukan tugas ini, pilihan paling mudah adalah dengan menggunakan sebuah blok “*Interpreted MATLAB Fcn*”, yakni sebuah blok yang memiliki respon keluaran berdasarkan *listing program* yang diketikkan oleh pengguna. Untuk melakukan tugas ini, berikut *listing program* yang digunakan:

```
function y = volume_to_level(x)
V = 1000*x; % mendefinisikan volume sebagai masukan dalam liter
R = 26/2; % Jari-jari silinder/tutup besar kerucut
r = 4.5/2; % Jari-jari katup /tutup kecil kerucut
h = 12; % Tinggi kerucut terpancung tersisa
Vk = (1/3)*(pi)*(R^2 + (R*r) + r^2)*h; % Volume kerucut terpancung
```

```

ho = (r/(R - r))*h; % Tinggi potongan kerucut (yang terpotong)
if V >= Vk %Jika kerucut telah penuh terisi material
    y = h + ((V-Vk)/(pi*(R^2))); % level material adalah tinggi kerucut
    ditambah tinggi volume pada tabung
else % Jika kerucut belum penuh
    y = (((3*V*(ho^2)) + (pi*(r^2)*(ho^3)))/(pi*(r^2)))^(1/3) - ho;
end

```

Untuk kerucut yang belum penuh, digunakan dari persamaan $t = \sqrt[3]{\frac{\pi h_0^3 r^2 + 3 V h_0^2}{\pi r^2}} - h_0$.

Keluaran dari blok fungsi terprogram ini dapat langsung menjadi keluaran dari *subsystem* Silo1 untuk dipantau melalui variabel *monitoring* “Level1”. Namun, masalah yang muncul adalah bagaimana seandainya bila katup terus terbuka sedangkan material pada Silo1 telah habis. Pada *integrator* mungkin tidak akan menjadi masalah, karena telah diberi batasan sebelumnya. Masalah yang sebenarnya adalah pada material keluaran (*material_out*) dari Silo1 yang akan diterima oleh *conveyor*. Perlu ada sebuah blok pembatas yang me-nolkan material keluaran saat material di dalam Silo telah habis. Untuk itu, digunakan sebuah blok *Interpreted MATLAB Fcn* (disebut *detektor_nol*) dari level ketinggian material (dalam cm), yang keluarannya dikalikan dengan debit keluaran Silo1. Jika material belum habis, blok *Interpreted MATLAB Fcn* akan memberikan isyarat 1, sehingga tidak akan mengganggu nilai debit keluaran dari Silo1. Sedangkan apabila material sudah habis, akan diberikan isyarat 0, sehingga debit hasil perkalian akan ikut menjadi nol. Dari hasil perkalian inilah yang benar-benar sudah dapat menjadi keluaran dari *subsystem* Silo1 untuk ditujukan ke *conveyor*.

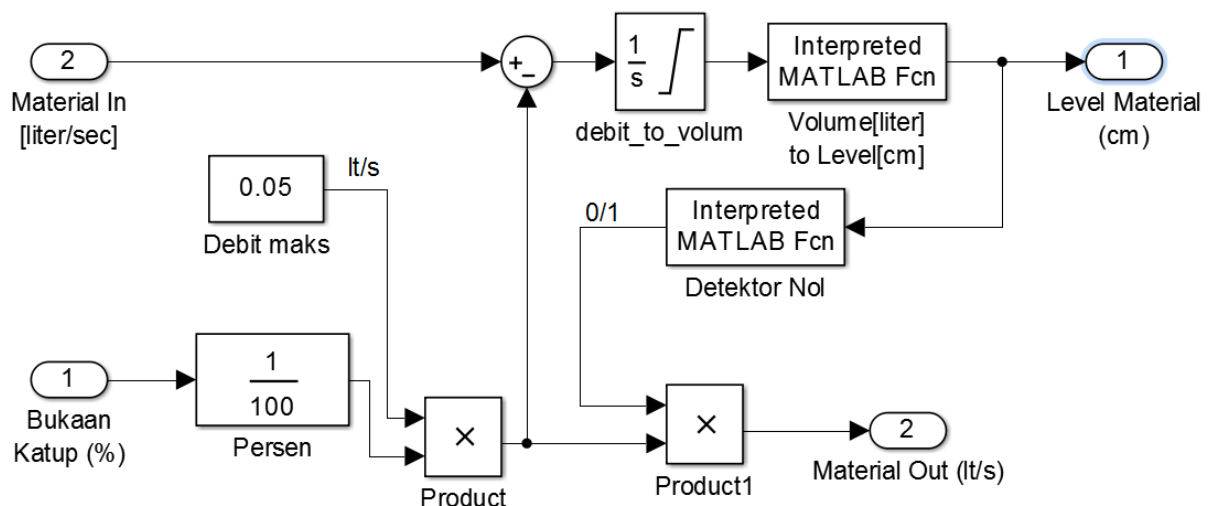
Berikut adalah *listing program* dari blok *detektor_nol*:

```

function y=detektor_nol(x)
if x==0
    y=0;
else
    y=1;
end

```

Berikut adalah blok diagram dari *subsystem* Silo1 secara utuh:



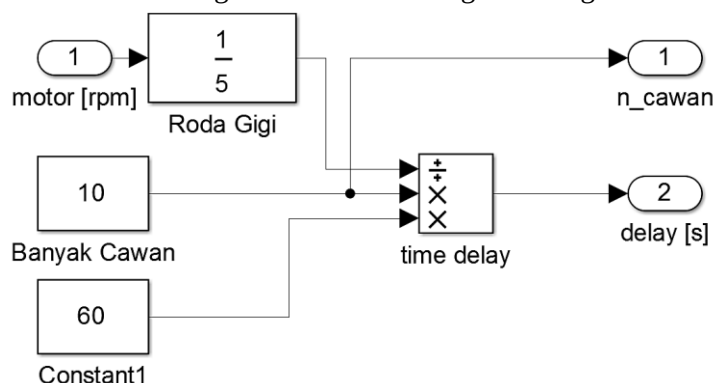
2. CONVEYOR

Subsystem ini bertujuan untuk mengubah volume yang masuk akan mengalami perubahan bentuk menjadi gundukan-gundukan material pada setiap cawannya, disertai dengan *delay* akibat dari proses memindahkan material. Kedengarannya sederhana, tapi *subsystem* inilah yang paling rumit.

Akan ada dua masukan dari *subsystem* ini, yakni *material_in* dan kecepatan motor *conveyor* dalam rpm. Akan kembali dibuat dua buah *subsystem*, yakni *subsystem* yang menerangkan data teknis dari *conveyor*, serta *subsystem* yang membentuk material masukan menjadi gundukan-gundukan material.

2.1. KETERANGAN SISTEM

Bagian inilah yang akan menerangkan data-data teknis dari *conveyor* sehingga menjadi hal berguna untuk bagian lainnya. Jika diketahui kecepatan motor sebagai parameter masukan adalah x rpm, maka *conveyor* akan menjatuhkan sebanyak $x/5$ cawan dalam setiap menit, atau setiap cawan membutuhkan $60 : \frac{x}{5}$ detik untuk dijatuhkan. Karena ada 10 buah cawan di atas *conveyor*, maka dari mulai mengisi cawan sampai mulai menjatuhkannya pada ujung satunya, dibutuhkan waktu selama 10 kali dari waktu menjatuhkan 1 cawan, sebagai waktu dalam detik. Jika keluaran dari blok ini adalah banyak cawan (n_cawan) dan , maka sistem ini dapat direalisasikan dengan susunan blok diagram sebagai berikut:



2.2. PEMBENTUK GUNDUKAN

Subsystem inilah yang paling rumit. *Subsystem* ini dibuat dengan tiga buah parameter masukan; *material_in*, n_cawan (jumlah cawan), dan .

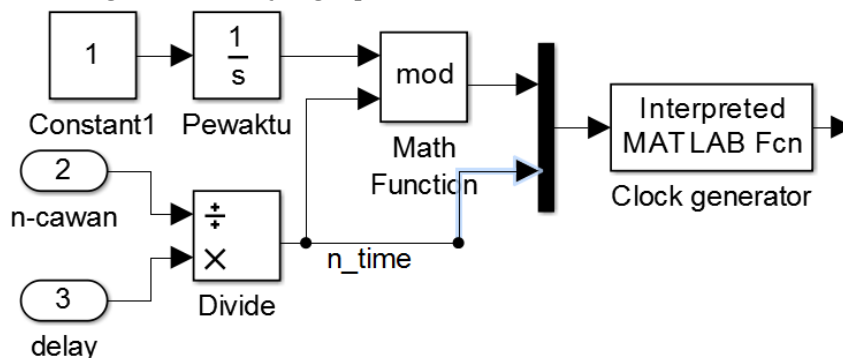
Pertama-tama, kita akan membutuhkan sebuah nilai yang menunjukkan waktu untuk mengisi setiap cawan dari tumpahan katup Silo1 (dinamakan n_time). n_time dapat dengan mudah didapatkan dengan membagi waktu total terhadap jumlah cawan yang ada (n_cawan). Waktu ini akan menjadi salah satu keluaran *subsystem*, karena setara dengan waktu yang dibutuhkan untuk memproses 1 buah gundukan pada setiap cawan, sehingga disebut juga *process time*.

Rangkaian pembentuk gundukan akan dibuat bekerja berdasarkan jumlah volume dari Silo1 yang diterima setiap waktu pengisian 1 cawan. Artinya yang kita butuhkan sekarang adalah sebuah pewaktu yang menentukan kapan volume dari setiap gundukan material akan mulai

dihitung, menahan volume tersebut sampai gundukan berikutnya, dan mengulang parameter waktu sebagai parameter fungsi untuk pembentuk gundukan –akan dibahas kemudian.

Sebagai pembuka, kita akan menghitung volume material yang diterima *conveyor* untuk mengisi satu buah cawan dan menjadi sebuah gundukan. Volume ini dapat kembali kita dapatkan dengan blok *integrator*. Namun, kita harus terus mengulang menghitung untuk gundukan-gundukan berikutnya. Untuk itu kita membutuhkan sebuah blok *integrator* yang dilengkapi dengan kemampuan *external*. Jika kita akan melakukan setiap selesai menghitung volume untuk satu gundukan, dan yang kita gunakan adalah mode reset tepi turun (*falling*), maka yang kita butuhkan adalah memiliki tepi turun setiap pergantian satu periode *clock*. Artinya *clock* yang akan dibuat adalah bernilai nol selama setengah periode awal *clock*, dan bernilai satu pada setengah periode berikutnya. Untuk melakukannya, kita dapat menggunakan sebuah blok *Interpreted MATLAB Fcn*, namun dengan beberapa parameter masukan.

Kita dapat menggunakan blok diagram beserta *listing program* berikut untuk membangkitkan *clock* yang tepat:



```
function y=fclock(x)
waktu=x(1); %waktu dari sisa bagi waktu terhadap waktu n_time
delay=x(2); %waktu untuk memproses setiap cawan (n_time)
banding=delay/2; %setengah periode n_time
if waktu<banding
    y=0; %output 0 selama setengah siklus pertama clock
else y=1; %output 1 selama setengah siklus berikutnya
end
end
```

Pertama kita akan membutuhkan sebuah pewaktu yang terus membangkitkan waktu selama simulasi dijalankan. Sisa hasil bagi waktu terhadap waktu setiap cawan (*n_time*) akan menghasilkan nilai dari 0 sampai *n_time*, yakni lamanya waktu setiap cawan dari mulai diisi. Bisa didapatkan dengan menggunakan fungsi “mod” dari blok *Math Function*. *n_time* juga menunjukkan lamanya waktu untuk 1 periode *clock*. Dari range nilai ini, kita dapat membuat keluaran akan bernilai 0 selama sisa hasil bagi lebih kecil dari setengah siklus *clock*, dan akan bernilai 1 selama setengah siklus berikutnya, yakni jika sisa hasil bagi telah lebih besar dari setengah siklus *clock*. Akhirnya kita telah mendapatkan sebuah sistem *clock* yang memberikan tepi turun setiap volume untuk 1 cawan telah selesai dihitung.

Setelah selesai dihitung volume untuk setiap cawan, perlulah membentuk gundukan material dengan volume yang tepat sama dengan nilai tersebut. Kita dapat membentuk gundukan

material namun dalam satuan liter/detik, karena sebagai masukan untuk silo berikutnya, dapat digunakan sebuah blok *Interpreted MATLAB Fcn*. Jika keluaran berbentuk debit yang dibentuk seperti gundukan-gundukan (seperti grafik positif fungsi sinus), namun dengan tetap memperhatikan volume yang diterima, dengan ilmu kalkulus dapat dibentuk persamaan berikut:

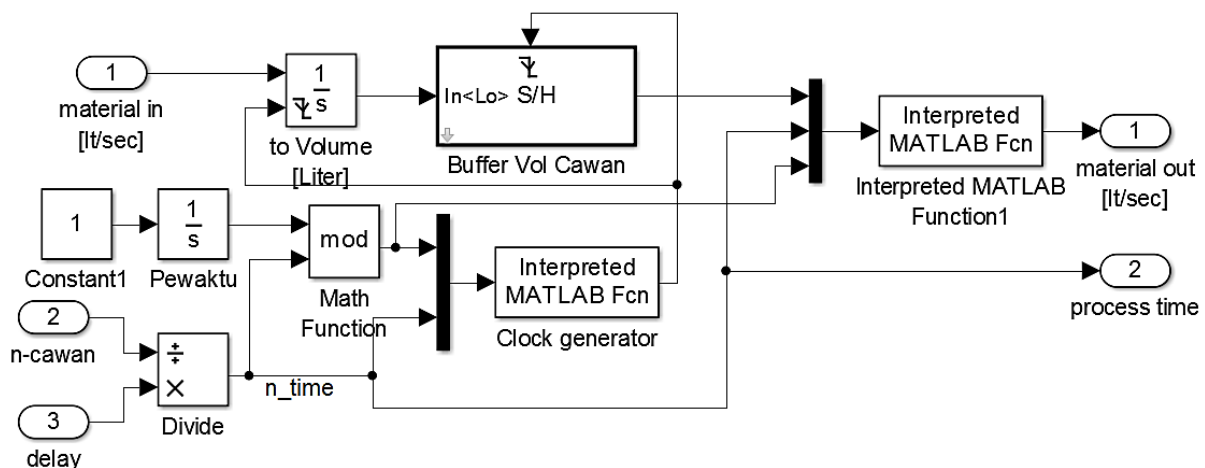
$$V_{in} = \int_0^{n_{time}} A \cdot \sin(\omega \cdot t) dt ; \text{ dengan } \omega = \frac{\pi}{n_{time}}$$

$$A = \frac{\pi \cdot V_{in}}{2 \cdot n_{time}}$$

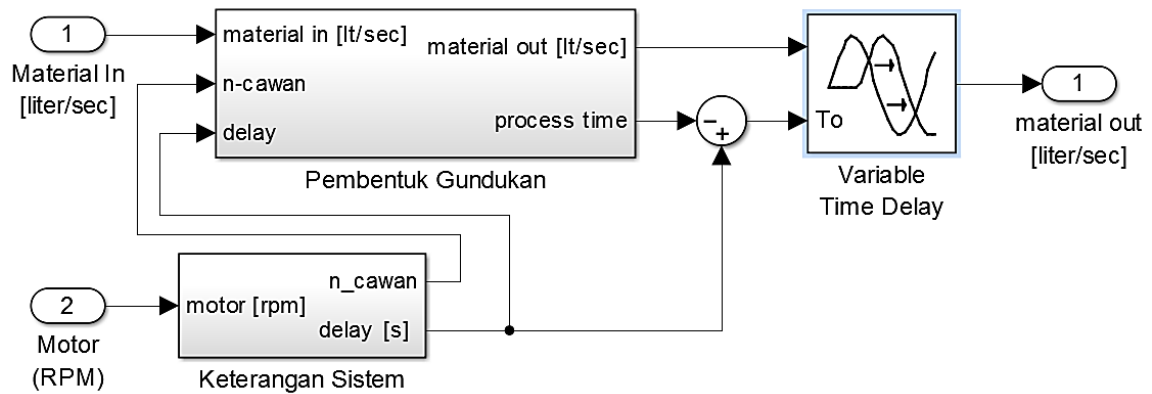
Dari persamaan ini, kita dapat membuat sebuah fungsi yang mendefinisikan volume yang tepat dari bentuk-bentuk gundukan yang dihasilkan, walau dalam satuan liter/detik. Namun, variabel n_{time} pada *listring program* berikut akan digantikan dengan variabel “waktu” yang ukurannya sengaja dibuat hanya 80% dari n_{time} , agar jarak antar satu gundukan dan gundukan berikutnya tetap terlihat. Berikut ini adalah *listring program*-nya:

```
function y=pembentuk_cawan(x)
vol=x(1); %volume untuk setiap cawan
n_time=x(2); %waktu pengisian tiap cawan
ftime=x(3); %parameter waktu
waktu=n_time-(n_time/5); %panjang gundukan dalam detik
omega=pi/waktu; %kecepatan sudut fungsi
A=0.5*pi*vol/waktu; %menentukan amplitudo yang tepat agar volume sesuai
if ftime<=waktu
    y=A*sin(omega*ftime);
else y=0;
end
end
```

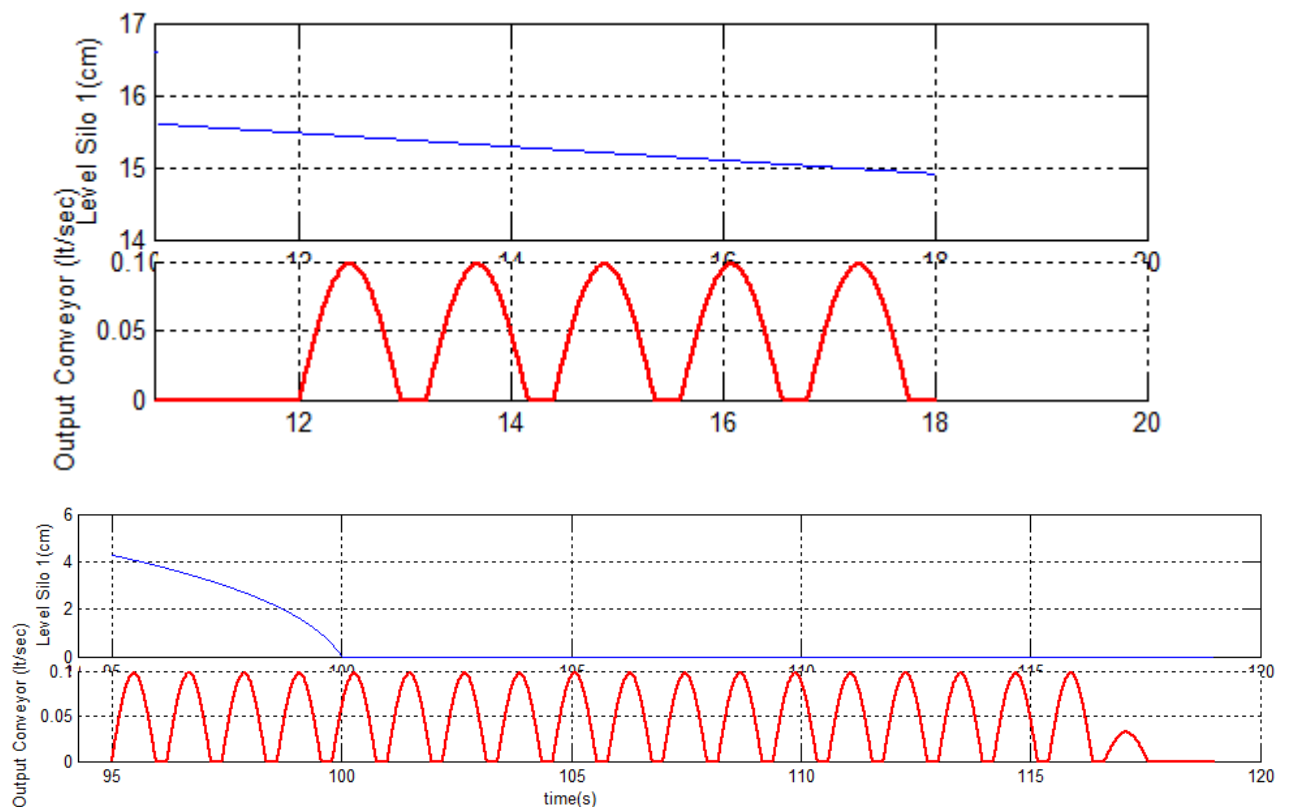
Namun, masalahnya adalah kita membutuhkan sebuah nilai masukan dari fungsi berupa volume yang akan dibuat gundukan berdasarkan hitungan dari blok *integrator*. Artinya, kita harus menahan hasil hitungan blok *integrator* tepat sebelum nilainya di- selama satu periode *clock*. Untuk melakukannya, kita membutuhkan sebuah blok yang tepat, *Sample and Hold*, dengan *clock* yang dibuat sama dengan *clock* reset dari blok *integrator* tersebut. Secara lengkap, berikut ditunjukkan blok diagram dari *subsystem* Pembentuk Gundukan:



Subsystem ini akan memakan waktu sebesar waktu untuk mengisi 1 cawan. Artinya, kita masih membutuhkan waktu sebesar 9 kali proses seperti ini lagi. Nilai *delay* sebesar itu adalah sama saja mengurangi waktu *delay* dari *subsystem* “*Keterangan Sistem*” dengan waktu proses yang dipakai oleh blok *subsystem* “*Pembentuk Gundukan*”. Untuk memberikan *delay*, kita dapat menggunakan sebuah blok *Variable Time* sebelum gundukan-gundukan material dibawa keluar, dengan nilai *delay* dari hasil pengurangan tersebut. Berikut ini adalah gambaran blok diagram *conveyor* secara lengkap:



Berikut ini respon keluaran dari *conveyor* dengan kecepatan motor 250rpm:

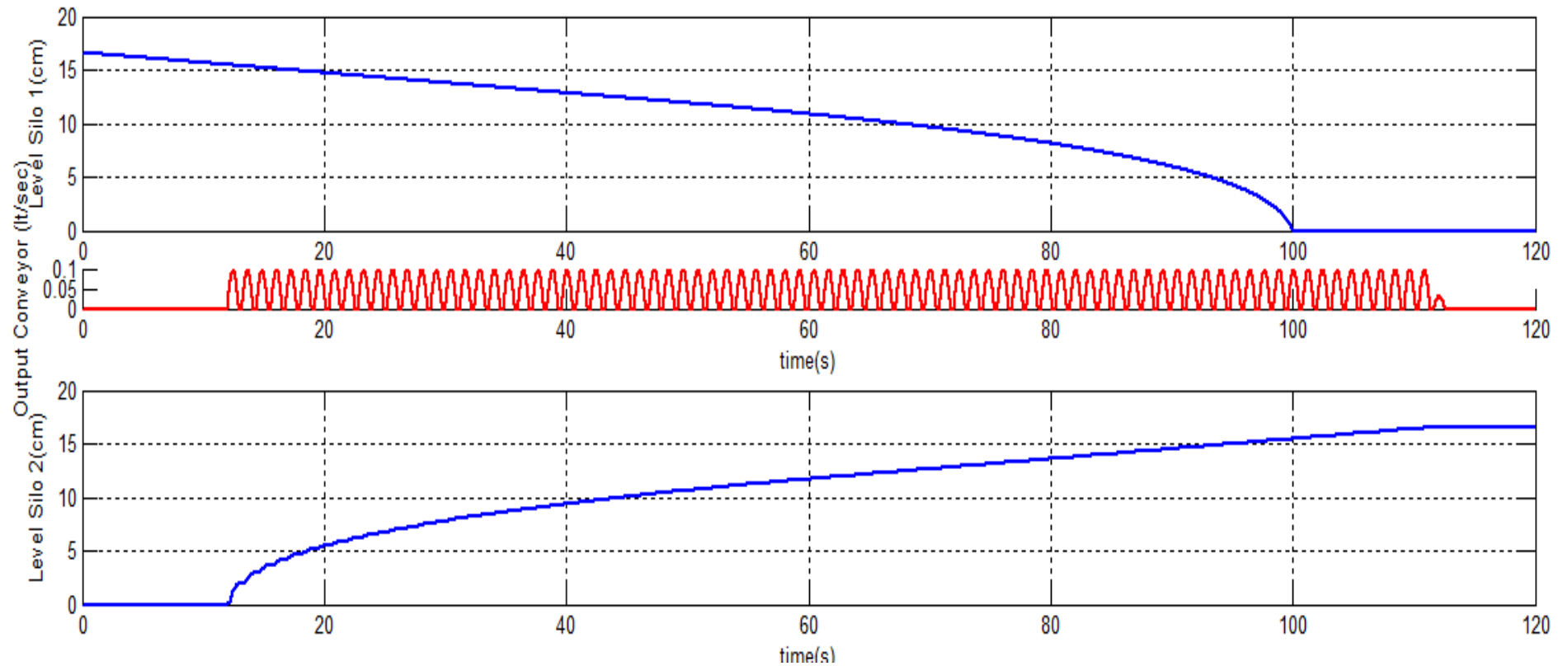


Terlihat bahwa respon keluaran dari *conveyor* berbentuk gundukan-gundukan material yang amplitudonya menyesuaikan dengan jumlah material yang masuk.

3. SILO2

Secara umum, blok diagram Silo2 akan sama saja dengan blok diagram pada Silo1, hanya saja Silo2 hanya dapat menampung material yang lebih sedikit dari Silo1. Untuk tinggi tabung pada Silo2, hanya mampu menampung material hingga 8.1296 liter. Untuk itu, perbedaan dari *subsystem* pada Silo2 hanya terletak pada batasan saturasi dari blok *integrator*-nya.

Berikut adalah grafik respon pemindahan material dari model Silo_to_Silo ini:



Respon yang baik adalah apabila ketinggian awal pada Silo1 akan sama dengan ketinggian akhir pada Silo2.

BAB IV

PENUTUP

KESIMPULAN

Sebagai mahasiswa elektro di bidang teknik kendali, mempelajari pemodelan sistem akan sangat penting. Selain untuk memonitoring sistem, pemodelan dapat membantu dalam melakukan analisis terhadap kesalahan-kesalahan sistem yang mungkin terjadi untuk meminimalisir kerugian dalam suatu proses. Namun, dibutuhkan keahlian yang baik dalam memodelkan sistem secara benar, karena pemodelan yang kurang tepat tentu saja dapat mengakibatkan kesalahan dalam memonitoring suatu proses.

“Laporan Hasil Proyek Simulasi Menggunakan Matlab”



Oleh:

SAHRUM NURHABIB YUSUF D411 12 267

MUHAMMAD IQBAL D411 12 283

SUB PRODI TEKNIK KOMPUTER KENDALI ELEKTRONIKA

PRODI S1 TEKNIK ELEKTRO

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

2014/2015

BAB 1 PROJECT 1

G.U.N.T Level Control

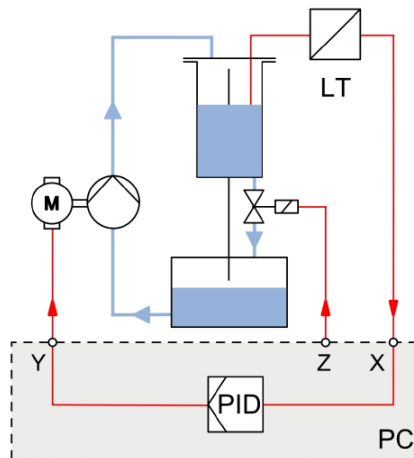
1.1 PENDAHULUAN

Salah satu dari kebutuhan manusia dalam kehidupan sehari-hari tak luput dari bahan-bahan yang berbentuk cair. Namun bahan cair seperti air di daerah tertentu bukan menjadi barang yang mudah untuk didapatkan. Kadang hanya ada pada waktu-waktu tertentu saja. Oleh karena itu dibutuhkan sebuah wadah yang digunakan untuk menampung bahan-bahan cair sehingga pada saat waktu tertentu saat air tersedia, air tersebut ditampung untuk digunakan pada saat tidak ada air yang tersedia, air dari penampungan tersebut dapat dimanfaatkan.

1.2 TUJUAN

Mengetahui bagaimana karakteristik dari pengendali PID kinerja Gunt Level Control menggunakan Matlab

1.3 TEORI DASAR



Gunt level control adalah sebuah alat yang digunakan untuk mengontrol volume dari sebuah tangki yang menampung barang cair. Pada gunt level control, terdiri dari sebuah tangkin, pompa, katup, selang atau pipa, sensor pengukur ketinggian, dan sebuah pengendali PID. Pengontrollan ini bertujuan agar volume air dalam tangki tetap pada volume yang diinginkan

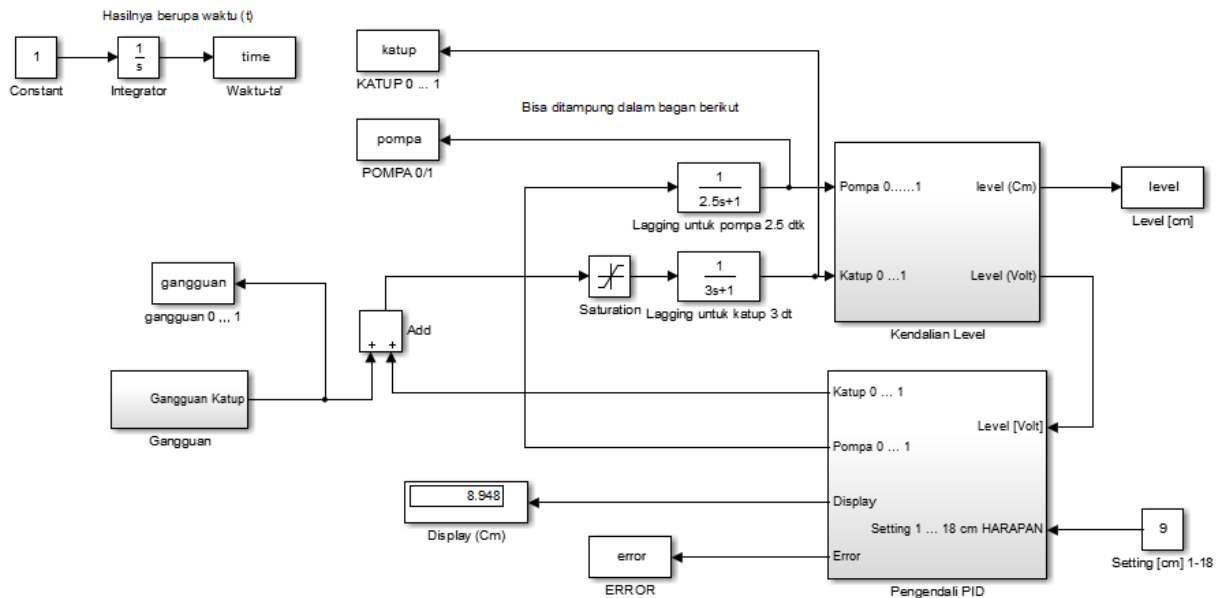
Pada gambar diatas merupakan ilustrasi dari sebuah gunt control dimana:

1. Garis Merah sebagai ilustrasi arah aliran listrik

2. Garis Biru sebagai ilustrasi aliran air aliran air

Sensor (LT) pada gambar akan membaca berapa ketinggian dari air pada bak penampung. Keluaran dari sensor ini sebagai sinyal listrik yang digunakan sebagai informasi inputan pengontrol PID. Pengontrol PID akan mengaktifkan pompa pada saat air dibawah titik normal dan akan membuka katub pada saat air diatas titik normal.

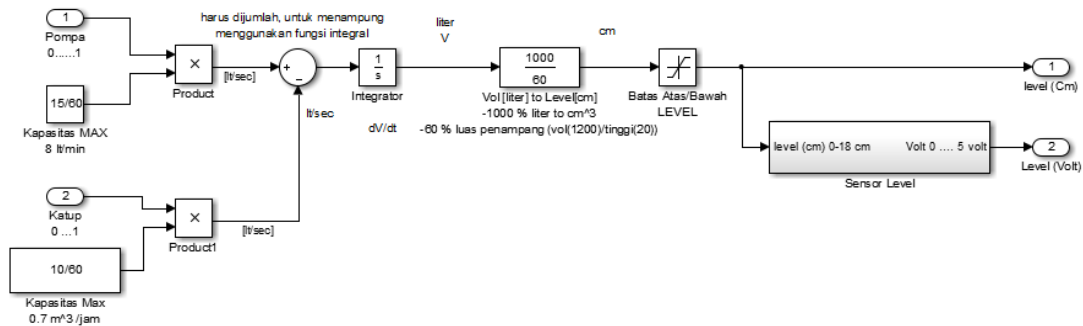
1.4 Analisa Dan Hasil Pengamatan



Gambar diatas merupakan simulasi sistem dari pengontrol. Pada pengontrol diset ketinggian dari air pada tangki yaitu 9 cm. Sedangkan ketinggian air pada tangki secara aktual yaitu pada 8.948 sedangkan errornya yaitu selisih antara nilai yang diset dengan nilai aktual yaitu 0.052.

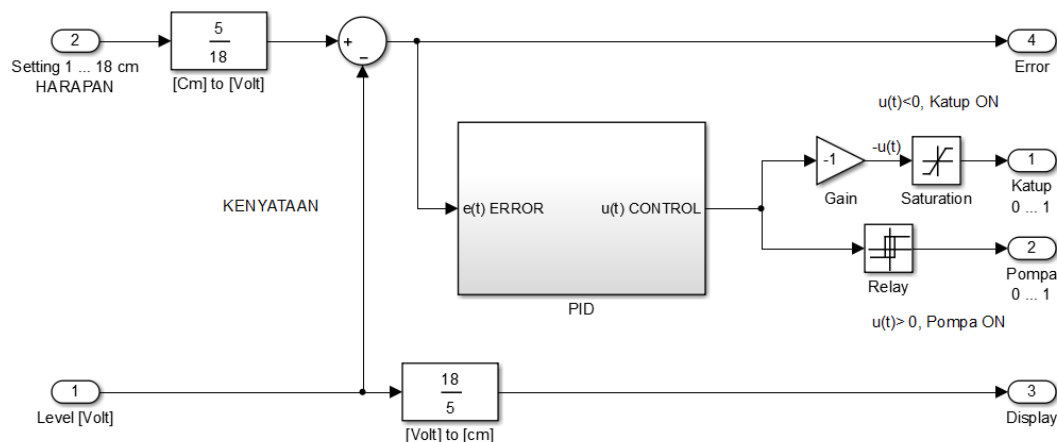
Pada simulasi diatas, terdapat 3 subsistem yang akan dijelaskan sebagai berikut:

1.4.1 Kendalian Level



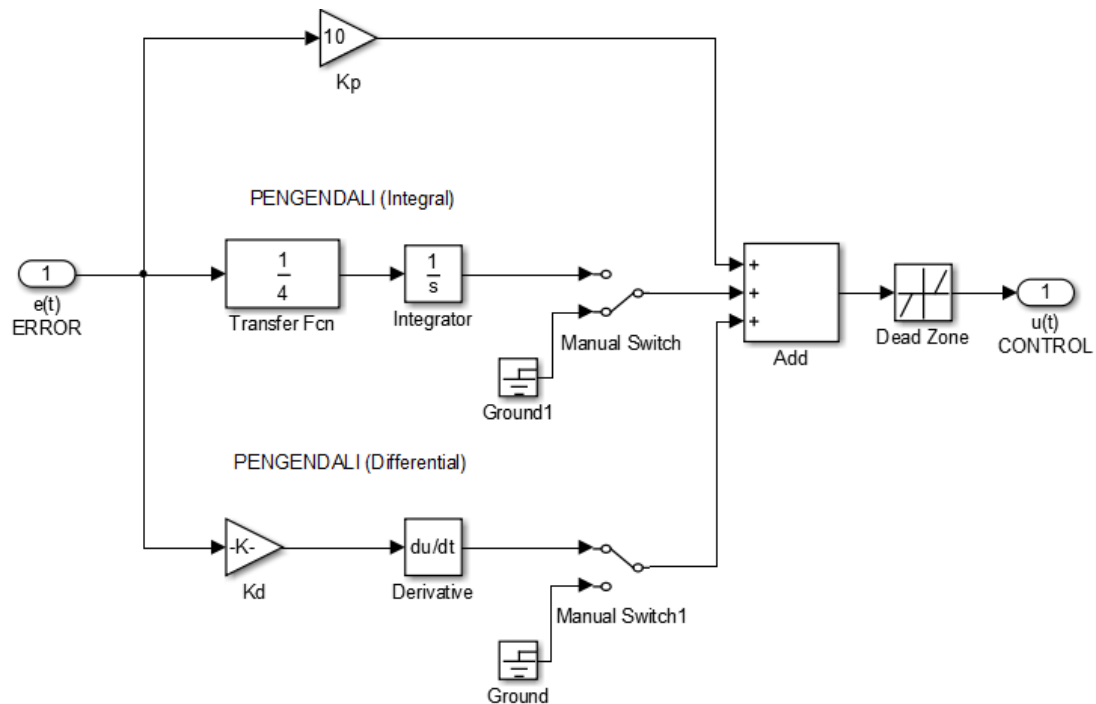
Pada gambar diatas, terdapat 2 input yaitu input 1 yang diisi oleh pompa. Dimana pompa berguna untuk mengalirkan air ke dalam tangki. Kemudian input ke 2 diisi oleh katup. Jika pompa bersifat menambahkan air dalam tangki, maka katup bertujuan untuk mengurangi air dalam tangki. Pada keluaran terbagi menjadi 2 bagian yaitu bagian output nyata dimana disini hasilnya dapat dilihat dengan mata dalam ukuran cm (centimeter) dan keluaran lainnya berupa sinyal listrik. Sinyal listrik sendiri sebelum keluar menjadi sinyal listrik sebelumnya mengalami proses perubahan dari hasil ukuran nyatanya dalam cm. Proses ini berlangsung pada subsistem sensor level.

1.4.2 Pengendali PID

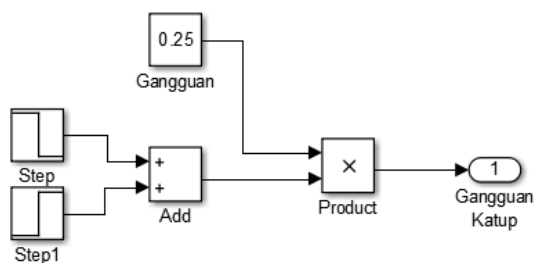


Subsistem Pengendali PID memiliki 2 inputan berupa settingan dari pengguna dan keluaran dari sensor yang berupa sinyal listrik. Agar kedua inputnya memiliki satuan yang sama, maka input dari settingan pengguna diubah terlebih dahulu sebelum dikurangkan dengan hasil dari sensor. Nilai dari harapan yang dikurangkan dengan nilai dari sensor kemudian disebut error. Nilai dari error sendiri ini kemudian akan ditampilkan sebagai keluaran 4. Keluaran 3 merupakan konversi dari sinyal listrik menjadi tampilan dalam bentuk besaran ukuran nyata yang kemudian akan ditampilkan pada display.

Error sendiri akan digunakan sebagai inputan dari Subsistem PID. Dalam subsistem PID sendiri berisi pengendali Proporsional, Integrator dan Differensial. Setiap pengendali ini memiliki inputan yang sama yaitu error. Output dari semua pengendali kemudian dijumlahkan. Deadzone pada output keluaran bertujuan agar keluaran pompa dan katub stabil dengan cara memberi batas toleransi kesalahan pada error



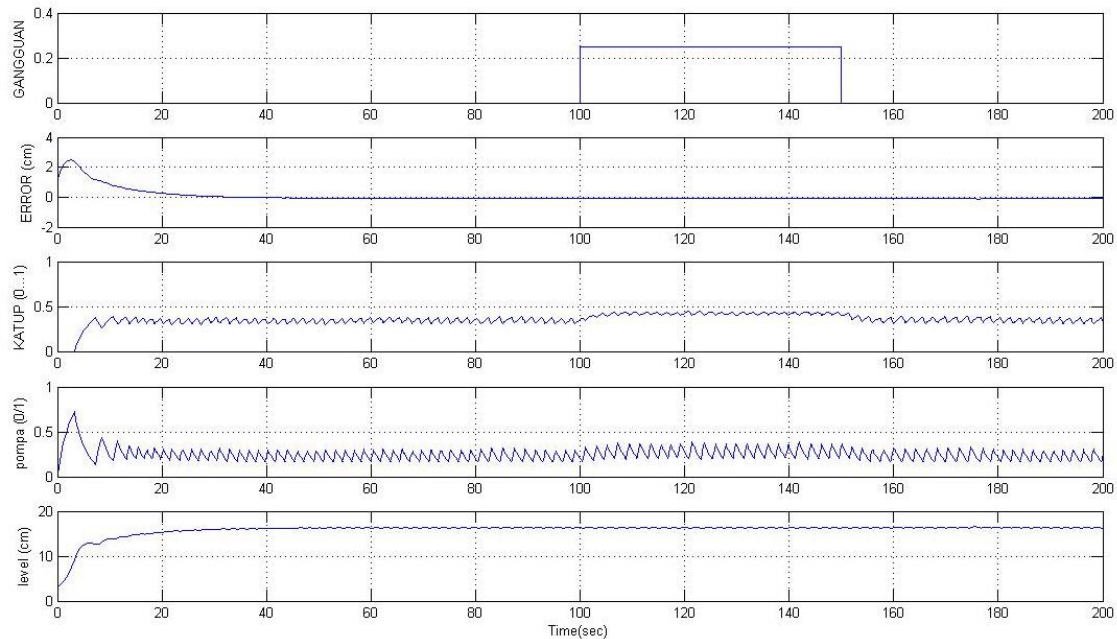
1.4.3 Gangguan



Subsistem gangguan bertujuan untuk mensimulasikan adanya gangguan sebagai bentuk realisasi terhadap kejadian pada kehidupan nyata sehingga simulasi ini semakin mendekati kejadian asli di dunia nyata. Pada subsistem ini disimulasikan adanya gangguan pada katub dimana pada saat tertentu katub tiba tiba terbuka tanpa ada perintah dari pengendali.

1.5 Analisa Hasil Simulasi

Untuk memperlihatkan bekerjanya sistem dan keadaan dari dari tangki dan pompa, ditampilkan dalam bentuk grafik sebagai berikut



Dari grafik di atas terlihat bahwa untuk mempertahankan ketinggian tangki, pompa dan katup bekerja bergantian hingga error mencapai titik 0. Pada saat ada gangguan, katup terbuka, tetapi pompa juga ikut bekerja sehingga keadaan air tetap pada ketinggian yang diinginkan.

1.6 Kesimpulan

Dari proyek ini diperlihatkan bagaimana cara mengontrol ketinggian sebuah tangki pengisian air dengan pengendali PID pada GUNT level control. Dengan menggunakan pengendali maka level air pada tangki tetap dapat dipertahankan bahkan pada saat ada gangguan pada pengontrol.

BAB 2 PROYEK 2

Conveyer Sebagai Alat Transpor Material Padat Dari Silo ke Silo

3.1 PENDAHULUAN

Pada project kali ini, kita mengamati perpindahan material dari silo 1 ke silo 2, dengan menggunakan *conveyer*. Silo yaitu sebuah wadah yang digunakan untuk menampung atau tempat penyimpanan sementara material padat. *Conveyer* adalah sebuah alat yang berfungsi untuk memindahkan material padat dari suatu tempat ke tempat yang lain. Dalam proyek ini kita menggunakan matlab sebagai simulasi karena keterbatasan waktu dan tempat. Kita tidak simulaskannya secara langsung dengan device, meskipun device ada di lab energi jurusan elektro. Proyek ini sendiri merupakan bagian alat pengering material padat yang dirancang Bapak Andani Achmad sebagai dalam rangka menyelesaikan disertasi sebagai syarat untuk mendapatkan gelar Doktor.

3.2 TUJUAN

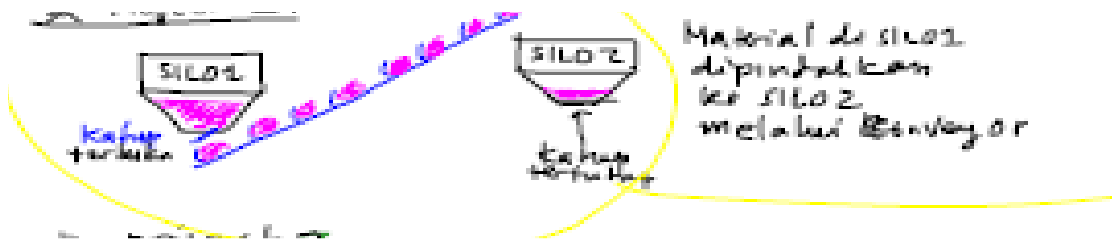
Mengetahui dan mengatur parameter-parameter setiap silo dan menentukan parameter *conveyer* sehingga perpindahan material dari silo ke silo tidak terjadi peurunan produk.

3.3 TEORI DASAR

Sejak zaman dahulu manusia selalu membuat alat yang dapat membantu manusia dalam membatasi kekurangannya dalam hal ruang dan waktu. Salah satu masalah utama yang dihadapi manusia adalah bagaimana cara yang ampuh dalam memindahkan suatu benda dari suatu tempat ke tempat lain dengan efisien.

Pada dunia industri maupun non industri terutama yang mengolah material padat atau barang jadi sifatnya padat membutuhkan sebuah alat yang berguna untuk mengangkut barang padat dari satu tempat ke tempat lain. Salah satu alat ciptaan manusia untuk memindahkan barang yang digunakan di industri sendiri adalah *conveyer*. *Conveyer* ini digunakan dalam memindahkan material padat dari suatu tempat ke tempat lain dalam jarak yang dekat.

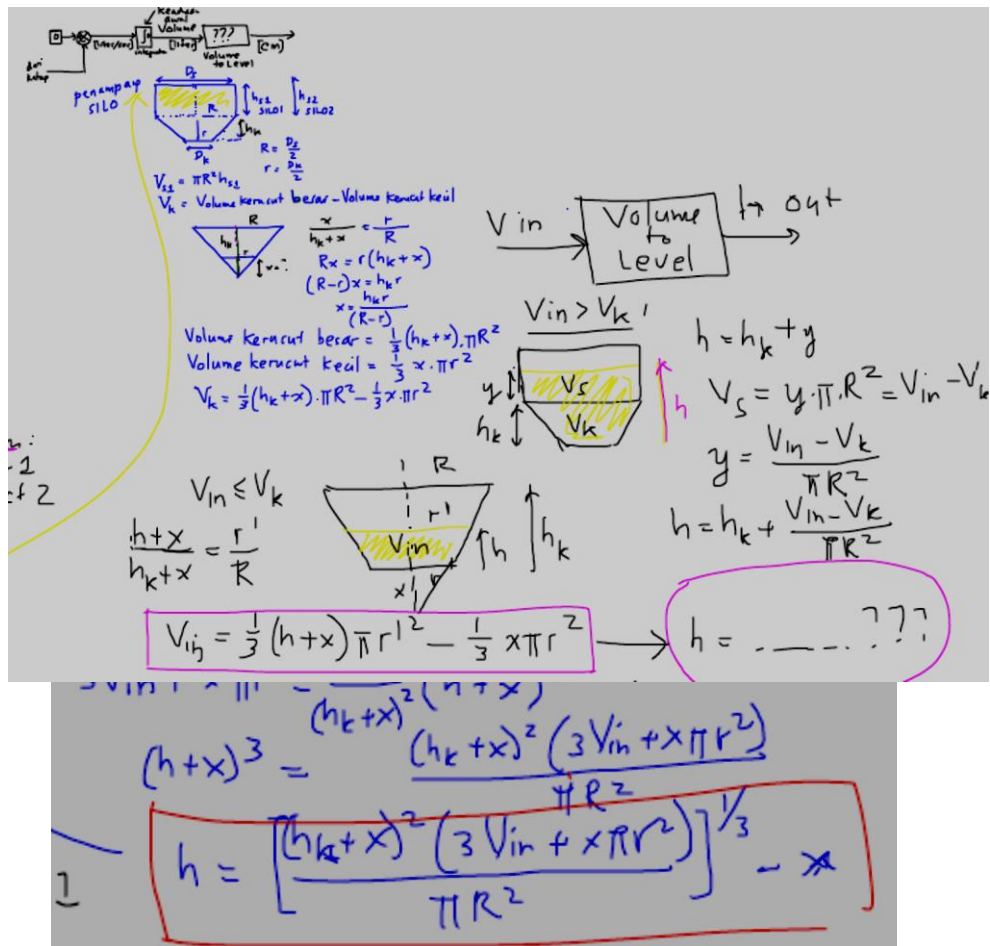
3.4 PEMBAHASAN



Gambar diatas merupakan simulasi mengenai project yang kita akan bahas, prinsip kerjanya material yang ada di silo 1 di pindahkan ke silo 2. Permasalahan disini memindahkan material dari silo 1 ke silo 2 sebanyak-banyaknya sehingga material di silo 1 semuanya dipindahkan ke silo2.

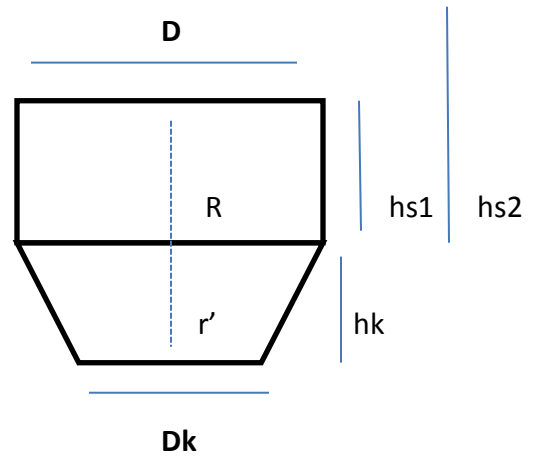
➤ Menghitung Penampang Silo

Untuk menghitung penampang silo dapat kita lihat rumus perhitungan dibawah ini :



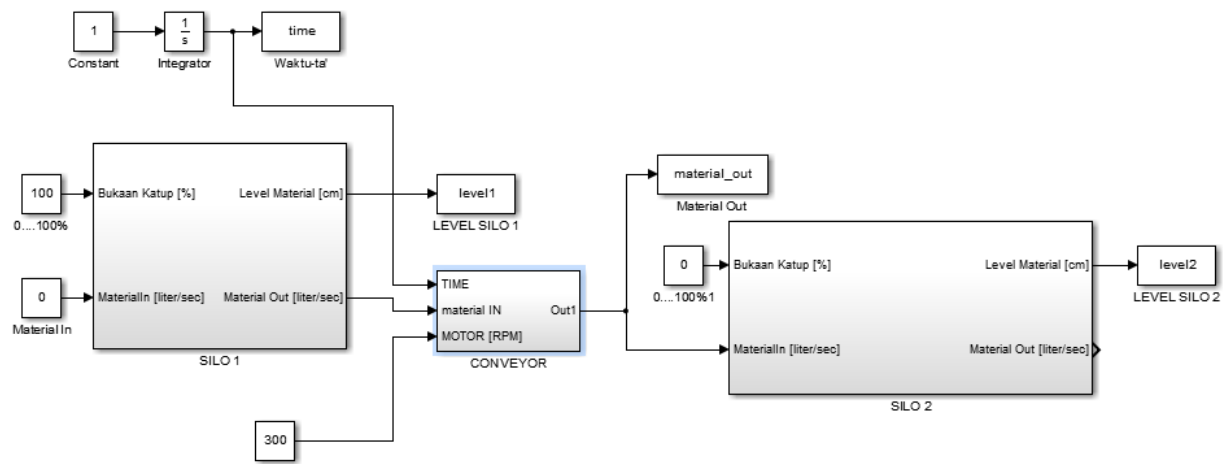
3.4.1 Hasil Pengukuran Silo

- a. Diameter Silo = 28 cm
- b. Jari-jari silo (R) = 14 cm
- c. Hk = 13 cm
- d. HS1 = 10 cm
- e. HS2 = 20 cm
- f. r' = 2,5 cm

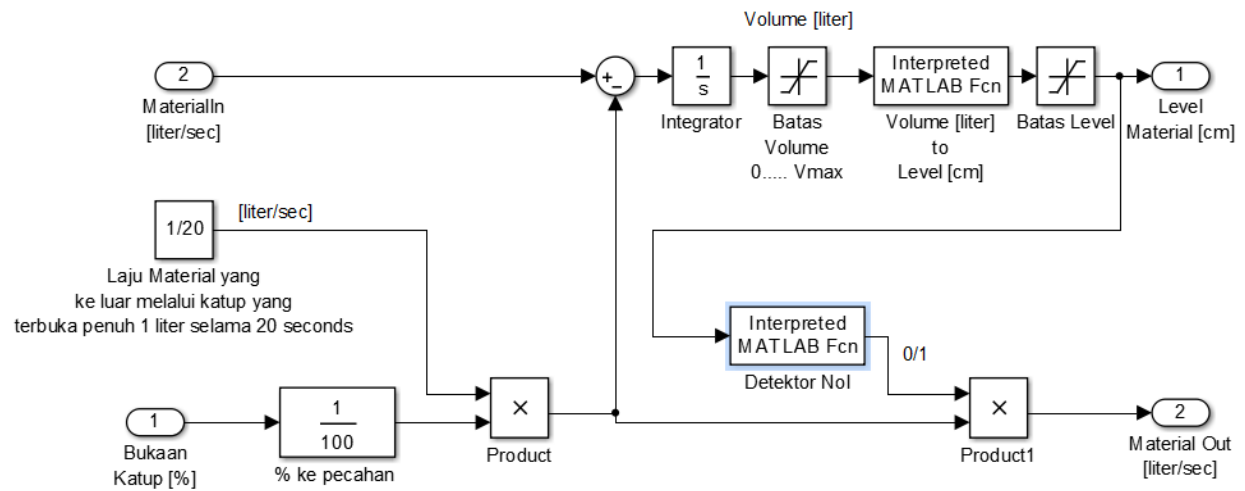


3.4.2 Analisa Dan Hasil Pengamatan

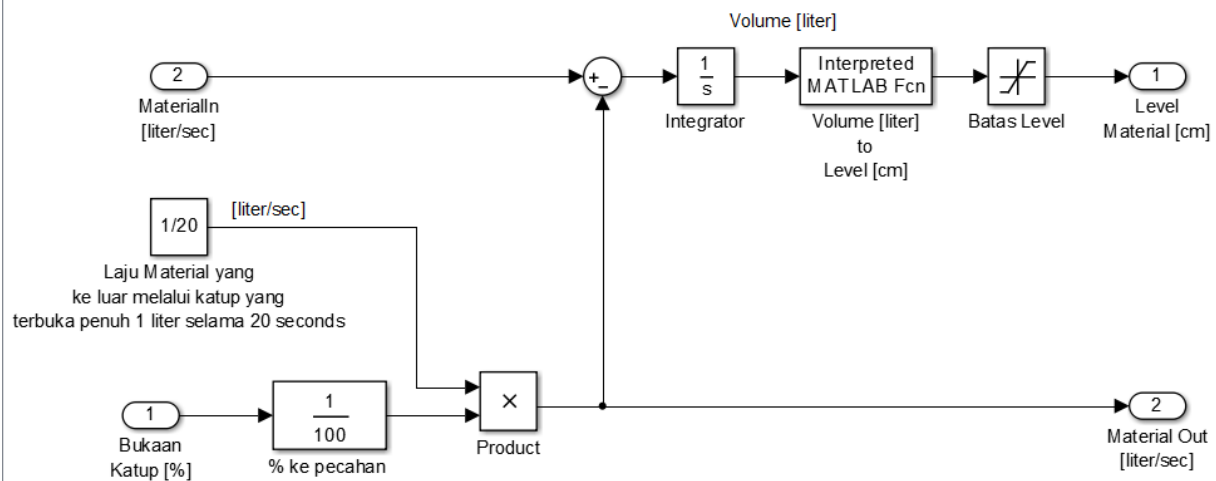
a. Gambar Simulasi Matlab



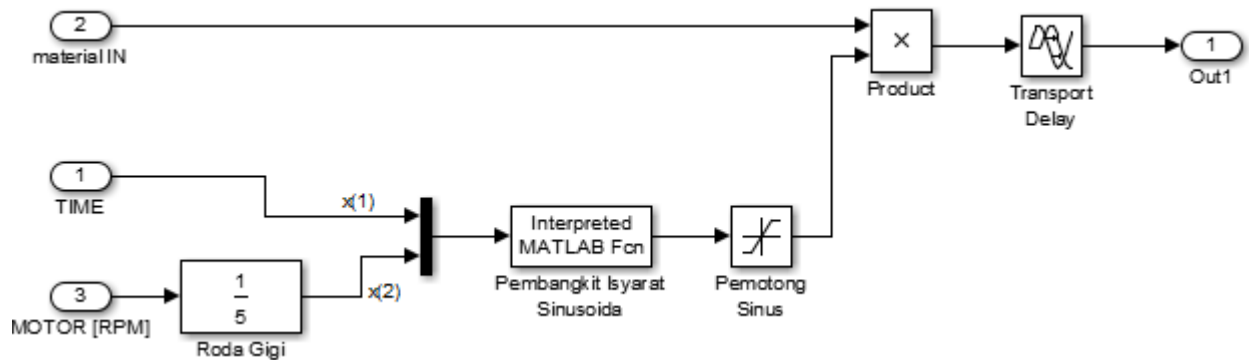
➤ **GAMBAR SUBSISTEM SILO 1**



➤ **GAMBAR SUBSISTEM SILO 2**



➤ GAMBAR SUBSISTEM CONVEYER



b. Scrip

➤ Volume to level

```
function y = volume_to_level(x)
V = 1000*x; % 1 liter = 1000 cm3
R = 13; % Jari-jari silinder = diameter/2
r = 2.5; % Jari-jari katup di bawah = diameter/2
hk = 12; % Tinggi kerucut terpotong
Vk = (1/3)*(pi)*(R^2 + (R*r) + r^2)*hk; % Volume kerucut
ho = (r/(R - r))*hk; % Tinggi potongan kerucut
if V >= Vk % Jika level di atas hk
y = hk + ((V-Vk)/(pi*(R^2))); % level material di silo 1 [cm]
else % Jika level di bawah hk
y = (((3*V*(ho^2)) + (pi*(r^2)*(ho^3)))/(pi*(r^2)))^(1/3) - ho;
end
```

➤ Pembangkit Sinus

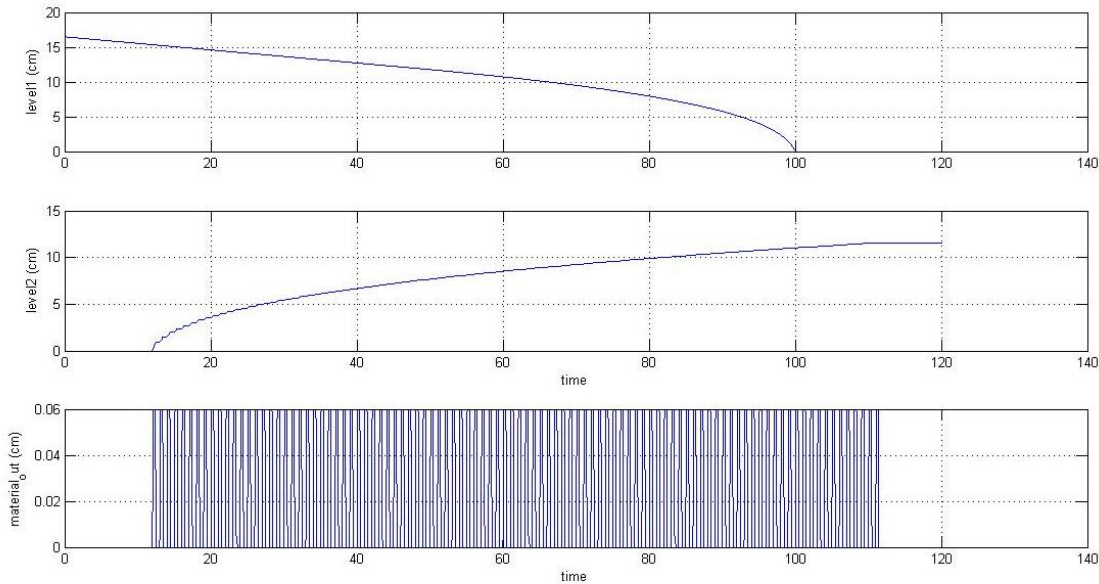
```
function y = pembangkit_sin(x)
t = x(1); % waktu
RPM = x(2); % Kecepatan MOTOR + roda gigi
D = 10*(60/RPM); % waktu membawa material dari Silo 1 ke Silo 2
T = D/10; % Periode isyarat sinus [sec]
f = 1/T; % Frekuensi [Hertz]
w = 2*pi*f; % Frekuensi sudut [rad/sec]
y = 2*sin(w*t); % isyarat sinusoida
```

➤ Detektor Nol

```
function y = detektor_nol(x)
level = x;
if level < 1
y = 0;
else
y = 1;
end
```

3.4.3 Analisa Hasil Simulasi

Hasil simulasi ditampilkan dalam bentuk grafik yang menampilkan keadaan dari silo 1, silo 2, dan keluaran dari conveyer.



Terlihat dari hasil simulasi diatas terjadi penurunan level .Level silo 1 berbeda dengan silo 2 ,pada silo 1 levelnya +-25 cm³ menjadi +-11 cm³ pada silo 2. Terjadi penurunan yang sangat drastis material produksi. Terlihat dari tumpahan cawan terjadi delay, sebesar 20 sec saat silo menumpahkan cawan ke conveyer. Banyak faktor yang menyebabkan hal ini karena pemberian value (nilainya) pada conveyer yang tidak sesuai dengan valuenya yang diinginkan.

3.5 Kesimpulan

Dari hasil project kita dapat mengetahui parameter-parameter dari setiap silo sehingga kita dapat merancang simulasinya dengan matlab. Dari tujuan diatas yakni memindahkan material cawan dari silo 1 ke silo 2 dengan conveyer, kita mendapatkan parameter value (nilai) tetapi kita tidak mendapatkan parameter value pada conveyer yang sesuai yang diharapkan , hasil yang didapatkan terlihat pada figure. Pada figure terjadi penurunan level material dari silo 1 da silo 2.

LAPORAN PROJECT
MATA KULIAH TEKNOLOGI KENDALI PROSES

**“Project Water Level Control dan
Project Silo to Silo Menggunakan Conveyor”**



DISUSUN OLEH :
MUH FAKHRI / D411 12 286
FAIZAL NAHRIR / D411 12 107

JURUSAN ELEKTRO FAKULTAS TEKNIK
UNIVERSITAS HASNUDDIN
2014/2015

KATA PENGANTAR

Puji syukur atas kehadiran kehadiran Allah SWT atas berkat rahmat dan karunianya laporan project mata kuliah Teknologi Kendali Proses dapat terselesaikan sesuai dengan waktu yang telah ditetapkan.

Kami mengucapkan terima kasih kepada semua pihak yang telah membantu dalam pengerjaan laporan ini. Semoga dengan adanya laporan ini dapat memberi pemahaman kita tentang teknologi kendali proses khususnya di dunia industri.

Perlu disadari bahwa penyusunan makalah ini jauh dari kata kesempurnaan oleh karena itu kami meminta masukan dan kritik bersifat membangun bukan menjatuhkan agar kedepannya pengerjaan laporan dimasa mendatang dapat lebih baik lagi.

Sekali lagi kami mengucapkan terima kasih dan penghargaan yang sebesar-besarnya sekian.

Penyusun

BAB 1

PROJECT WATER LEVEL CONTROL

PENDAHULUAN

Dunia industri terus berkembang dengan sistem-sistem yang baru dalam bidang manufaktur maupun energi, khususnya sistem kontrol. Saat ini banyak sekali ditawarkan suatu metode kontrol yang efektif dan mudah untuk diimplementasikan, salah satunya kontrol dengan sistem PID (Proporsional Integral Derivative).

Kontrol level air merupakan salah satu dari sekian banyak system yang ada dalam dunia industri. Selain sederhana, sistem tersebut banyak sekali digunakan dalam dunia industri, misalnya: industry kimia, proses produksi minyak dan gas, dan lain-lain.

Pada project ini dilakukan proses kontrol level air dari suatu prototype Water Level Control dari GUNT. Kontrol yang digunakan menggunakan sistem PID. Dalam kontrol level air, terjadi ketidakstabilan dalam hal level air. Oleh karena itu, untuk mengatasi hal tersebut digunakan sistem PID untuk menstabilkan. Pada project ini, dengan bantuan matlab dilakukan simulasi dari model Water Level Control GUNT dengan menggunakan sistem PID

TUJUAN

- a. Mengamati output dari sistem yang tidak memiliki pengendali, diberikan pengendali, dan diberikan gangguan
- b. Mengamati pengaruh tiap-tiap pengendali pada kinerja control sistem
- c. Menentukan karakteristik pengendali PID (Proporsional, Integral, Derivatif) sehingga didapatkan level air yang stabil

DASAR TEORI

1. Kontrol PID

Kontrol PID merupakan alat standar bagi otomasi industri. Fleksibilitas pada kontroler membuat kontrol PID digunakan pada banyak situasi. Kontroller juga dapat digunakan pada selective control maupun konfigurasi kontroller yang lain. Algoritma PID dapat didefinisikan sebagai berikut:

$$u(t) = K_c \left(e(t) + \frac{1}{T_I} \int_0^t e(t) dt + T_D \frac{de}{dt} \right)$$

dimana,

$u(t)$ = sinyal kontrol

$e(t)$ = error

K_c = gain kontroller

T_I = integral time

T_D = derivative time

Pengendali Proporsional (P)

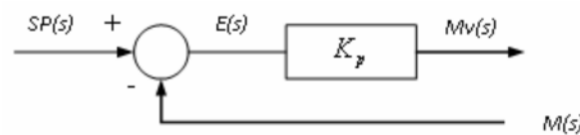
Kontroler proporsional memiliki 2 parameter, pita proporsional (*proportional band*) dan konstanta proporsional. Daerah kerja kontroler efektif dicerminkan oleh pita proporsional sedangkan konstanta proporsional menunjukkan nilai factor penguatan terhadap sinyal kesalahan, K_p . Hubungan antara *proportional band* (PB) dengan *konstanta proporsional* (K_p) ditunjukkan secara oleh Persamaan berikut:

$$PB = \frac{100\%}{K_p}$$

Dimana : PB = *Proportional Band*

K_p = *Gain Proses*

Diagram blok pengendali proporsional ditunjukkan seperti pada gambar berikut:



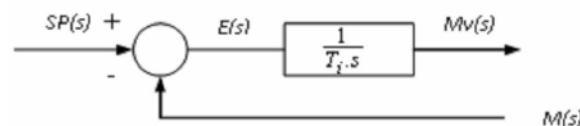
Penggunaan mode kontrol proporsional harus memperhatikan hal – hal berikut :

- jika nilai K_p kecil, mode kontrol proporsional hanya mampu melakukan koreksi kesalahan yang kecil, sehingga akan menghasilkan respon sistem yang lambat.
- jika nilai K_p dinaikkan, respon sistem menunjukkan semakin cepat mencapai keadaan stabilnya.
- Namun jika nilai K_p diperbesar sehingga mencapai harga yang berlebihan, akan mengakibatkan system bekerja tidak stabil, atau respon sistem akan berosilasi.

Kontrol *P (Proportional)* selalu sebanding dengan besarnya input. Bentuk *transfer function* dari kontrol P adalah $U = K_c \cdot e$ dimana ; K_c = gain kontrol proporsional

Pengendali Integral (I)

Kontroller integral memiliki karakteristik seperti halnya sebuah integral. Keluaran kontroller sangat dipengaruhi oleh perubahan yang sebanding dengan nilai sinyal kesalahan. Keluaran kontroller ini merupakan jumlahan yang terus menerus dari perubahan masukannya. Kalau sinyal kesalahan tidak mengalami perubahan, keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. Diagram blok mode kontrol *integral* ditunjukkan oleh gambar berikut:



Kontroler integral mempunyai beberapa karakteristik berikut ini:

- Keluaran kontroler butuh selang waktu tertentu, sehingga kontroler integral cenderung memperlambat respon.
- Ketika sinyal kesalahan berharga nol, keluaran kontroler akan bertahan pada nilai sebelumnya.

- Jika sinyal kesalahan tidak berharga nol, keluaran akan menunjukkan kenaikan atau penurunan yang dipengaruhi oleh besarnya sinyal kesalahan dan nilai K_i .
- *Konstanta integral* K_i berharga besar, *offset* akan cepat hilang. *Saat* nilai K_i besar akan berakibat peningkatan osilasi dari sinyal keluaran controller.

Transfer function dari unit *control integral* adalah :

$$U = \frac{1}{T_I} K_c \int e \cdot dt$$

Dimana :

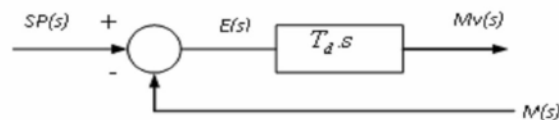
T_I = *integral time*

e = *error* (input dari unit *control*)

K_c = *gain* dari *controller*

Pengendali Differensial (D)

Keluaran controller *differensial* memiliki sifat seperti halnya suatu operasi *derivatif*. Perubahan yang mendadak pada masukan controller, akan mengakibatkan perubahan yang sangat besar dan cepat. Diagram blok pengendali difrensial ditunjukkan oleh gambar berikut:



Karakteristik dari controller *differensial* adalah sebagai berikut:

- Kontroler ini tidak dapat menghasilkan keluaran bila tidak ada perubahan atau *error* sebagai sinyal kesalahan untuk masukannya.
- Jika sinyal *error* berubah terhadap waktu, maka keluaran yang dihasilkan controller tergantung pada nilai T_d dan laju perubahan sinyal kesalahan.
- Controller *differensial* mempunyai karakter untuk mendahului, sehingga controller ini dapat menghasilkan koreksi yang signifikan sebelum pembangkit *error* menjadi sangat besar. Jadi controller *differensial* dapat mengantisipasi pembangkit *error*, memberikan aksi yang bersifat korektif, dan cenderung meningkatkan stabilitas sistem.

Transfer function dari unit *control differential* adalah :

$$U = K_c \cdot T_D \frac{de}{dt}$$

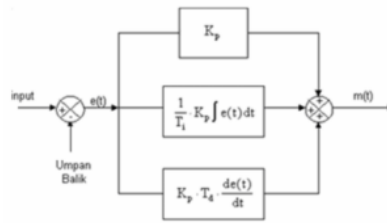
dimana,

K_C = *gain*

e = *error*

T_D = *derivative time*

Keluaran controller PID merupakan penjumlahan dari keluaran controller proporsional, controller integral dan controller differensial. Gambar diatas menunjukkan hubungan input dan output pada mode control PID. Karakteristik controller PID sangat dipengaruhi oleh kontribusi besar dari ketiga parameter P, I dan D. Penngaturan nilai konstanta K_p , T_i , dan T_d akan mengakibatkan penonjolan sifat dari masing-masing elemen.

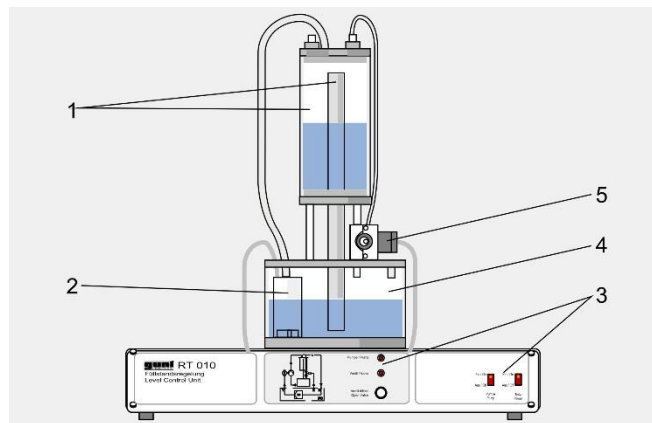


Satu atau dua dari ketiga konstanta tersebut dapat *disetting* lebih menonjol dibanding yang lain sehingga konstanta yang menonjol itulah akan memberikan kontribusi lebih dominan pada respon sistem secara keseluruhan. Pengaruh nilai K_p , T_i dan T_d pada respon sistem adalah :

- K_p yang kecil akan membuat pengendali menjadi sensitif dan cenderung membawa loop berosilasi, sedangkan K_p yang besar akan meningkatkan *offset* yang besar juga.
- T_i yang kecil bermanfaat untuk menghilangkan *offset* tetapi juga cenderung membawa sistem menjadi lebih sensitif dan lebih mudah berosilasi, sedangkan T_i yang besar belum tentu efektif menghilangkan *offset* dan juga cenderung membuat sistem menjadi lambat.
- T_d yang besar akan membawa unsur D menjadi lebih menonjol sehingga respon cenderung cepat, sedangkan T_d yang kecil kurang memberi nilai ekstra pada saat – saat awal.

2. Water Level GUNT

Pada proyek ini, dibuatkan model dari sistem berikut pada simulink matlab:



Level control diatas terdiri atas beberapa bagian yaitu:

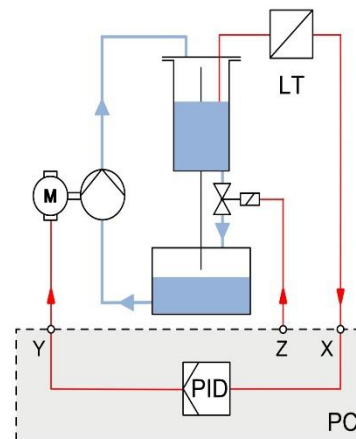
1. Level-controlled tank with overflow
2. Pump
3. Displays and controls
4. Storage tank
5. Proportional valve

Dengan spesifikasi sebagai berikut:

- Level-controlled tank
- capacity: 1200mL

- Storage tank
 - capacity: 3700mL
- Pump
 - power consumption: 18W
 - max. flow rate: 8L/min
 - max. head: 6m
- Proportional valve: Kvs: 0,7m³/h
- Pressure sensor: 0...30mbar (0...300mm)
- Software controller configurable as P, PI, PID and switching controller
- Software
 - process schematic with controller type selection (manual, continuous controller, two- or three-point controller, programmer)
 - time functions
 - simulation function
 - disturbance variable input
- Dimensions and Weight
 - LxWxH: 600x450x800mm
 - Weight: approx. 22kg

Prinsip kerja dari sistem ini adalah: Air yang berada di storage tank (tangki penyimpanan) dengan menggunakan pompa yang terkendali dipompa untuk mengisi level-controlled tank (tangki level) ketika ketinggian air sudah lebih besar atau mencapai level yang di setting maka ketinggian tersebut dibaca sensor ketinggian dan pengendali akan membuka katup sehingga dihasilkan ketinggian sesuai settingan. Ketika level-controlled tank melebihi kapasitasnya maka secara otomatis air akan kembali ke storage tank menggunakan pipa overflow. Sinyal output dari sensor mempengaruhi kecepatan motor dan laju aliran air.

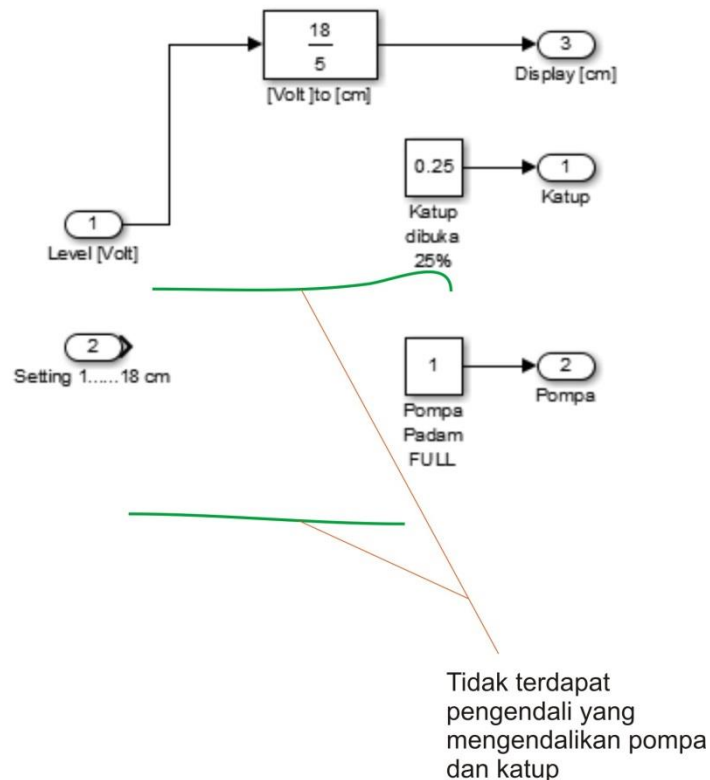


ANALISA DAN HASIL PENGAMATAN

Simulasi dilakukan menggunakan fasilitas simulink matlab water level control GUNT dibuatkan modelnya di matlab. Pada proyek ini, simulasi yang dilakukan yaitu simulasi tanpa pengendali, dengan pengendali dan dengan gangguan.

Simulasi dengan pengendali adalah simulasi water level control yang memanfaatkan pengendali PID dalam mengendalikan ketinggian air pada tangki. Hal-hal yang berpengaruh dalam kendalian tersebut adalah pompa dan katup yang akan diatur agar menghasilkan level ketinggian air pada tangki yang sesuai dengan yang disetting atau yang diinginkan.

Gambar Subsistem Kendalian Level



Gambar Subsistem Pengendali PID

Pada gambar **subsistem Pengendali PID** terlihat bahwa pengendali ini blum memiliki sistem PID. Sehingga pengaturannya dilakukan secara manual. Katup dari water level control diset pada nilai 0.25 dengan demikian katup dibuka 25 % dan pompa diset 1 dengan demikian pompa padam.

Pada gambar **subsistem Kendalian Level** terlihat bahwa output dari susbsitem Pengendali yang berupa nilai pompa dan katup dimasukkan ke Kendalian Level. Nilai pompa dikalikan dengan kapasitas pompa. Ketika nilai pompa 0 maka hasil kalinya adalah nol , berarti pompa padam. Nilai katup juga dikalikan dengan kapasitas bukaan katup. Kedua hasil tersebut di kurangkan lal dimasukkan ke intgrator. Outpu intgator berupa liter ehingga dikonersi ke cm. Selanjutnya dilakukan pembatasan. Sehingga didapatlah nilai levelnya. Nilai level tersebut diubah ke volt oleh sistem pembacaan sensor untuk diumpan ke subsistem Pengendali PID ntuk diolah.

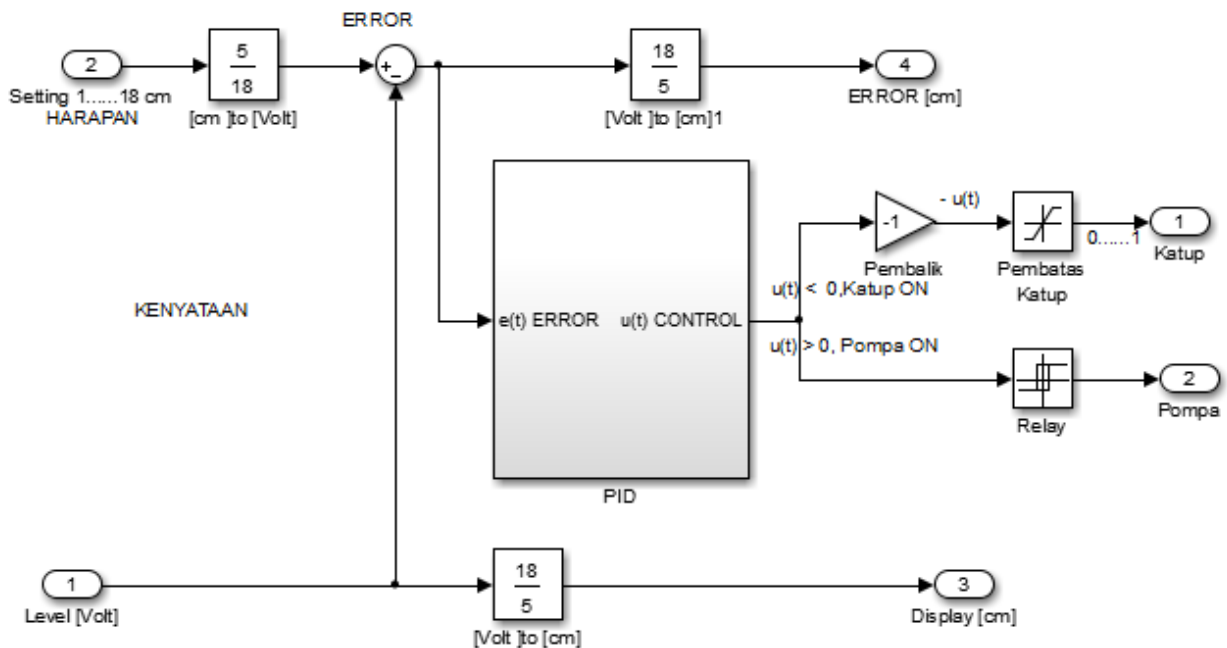
Hasil simulasi menunjukkan level ketinggian air pada tangki akan selalu berada pada level overflow atau maksimum yaitu pada ketinggian 19 cm. Seperti yang dapat dilihat pada gamba berikut.



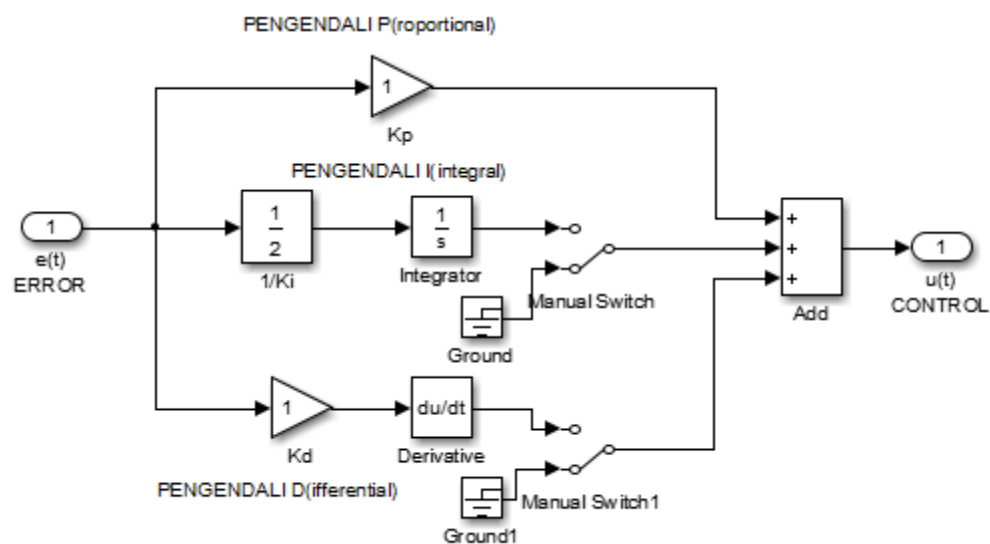
- **SIMULASI WATER LEVEL CONTROL DENGAN PENGENDALI**

Gambar Simulasi Water Level Control dengan Pengendali

Dari gambar di atas terlihat bahwa dilakukan penambahan variabel output yaitu ERROR untuk melihat error yang terjadi.. Setting dalam sistem kendalian ini adalah 10 cm. Pada pompa dan katup diberikan lagging. Pada pompa sebesar 2,5 detik dan katup sebesar 2,5 detik juga.



Gambar Simulasi Subsistem Pengendali PID



Gambar Simulasi Subsistem PID

Pada **subsistem Pengendali PID** di atas dapat dilihat bahwa pompa dan katup di atur oleh keluaran yang dihasilkan oleh PID.

Gambar **subsistem PID** diatas terlihat bahwa nilai error yang terjadi diatasi oleh pengendali PID denga melakukan pengontrolan sesuai pengaturan berikut:

Ketika $\text{Error} +$ berarti bahwa Harapan (ketinggian yang diinginkan) $>$ Kenyataan (ketinggian yang dihasilkan), maka Pompa On sehingga akan dialirkan air untuk mengisi tangki sampai pada ketinggian yang diinginkan.

Ketika $\text{Error} -$ berarti bahwa Harapan (ketinggian yang diinginkan) $<$ Kenyataan (ketinggian yang dihasilkan), maka Katup On sehingga tangki air akan dikuras sampai pada ketinggian yang diinginkan.

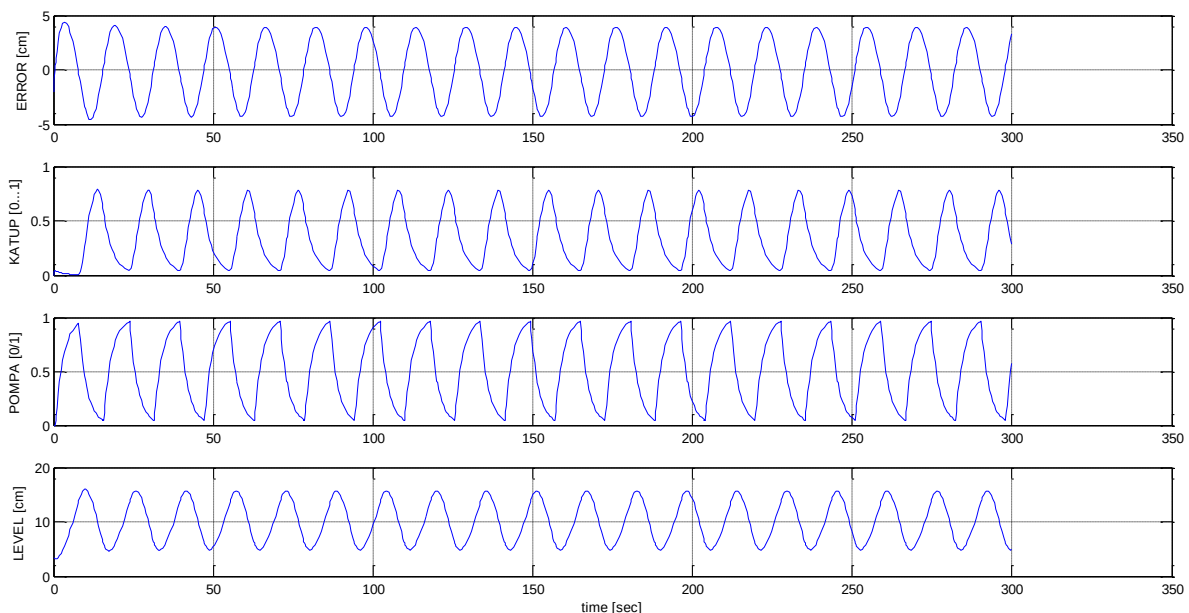
Berikut ini hasil keluaran yang diperoleh dengan memberikan pengendali yang berbeda-beda dengan besar nilai tiap pengendali yang berbeda-beda pula. Besar nilai yang digunakan dalam tiap pengendali menggunakan metode Heuristik.

Pengendali P

Pada simulasi pengendalinya hanya P. Dari hasil simulasi dapat dilihat bahwa pengendali P bersifat penguat sehingga mempercepat kecepatan respon. Semakin besar nilai K_p semakin cepat pula respon kerja yang dihasilkan seperti yang dapat dilihat pada hasil data berikut ini.

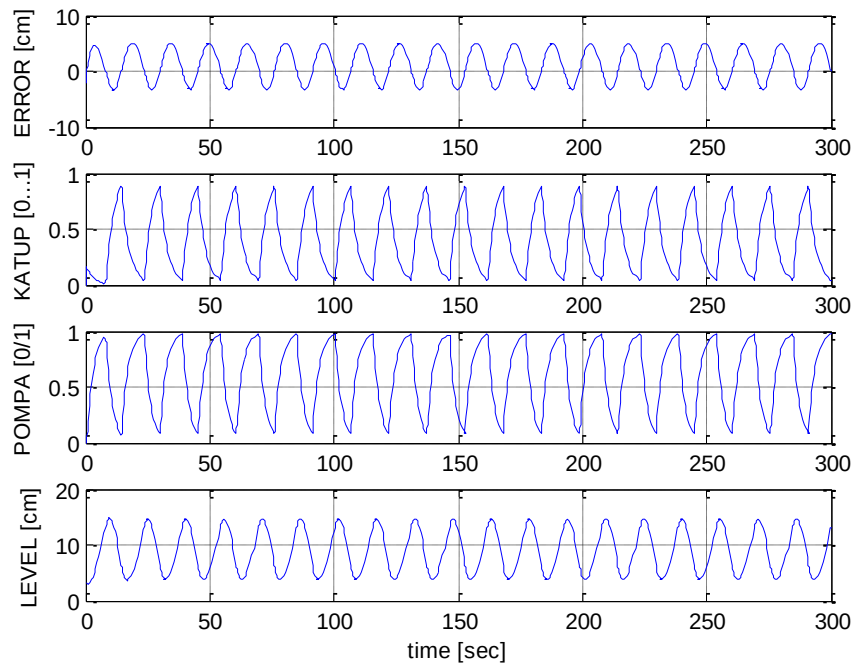
- Ketika $K_p = 1$

Ketika hanya pengendali P yang mempengaruhi dalam pengendalian. Dengan besar $K_p = 1$ maka akan diperoleh hasil keluaran dari tiap variabel keluaran yang tidak stabil. Seperti yang dapat dilihat pada gambar berikut ini. Error yang terjadi cukup besar. Sehingga mempengaruhi kontrol katup dan pompa sehingga berakibat pada ketinggian (level) air.



- $K_p = 98$

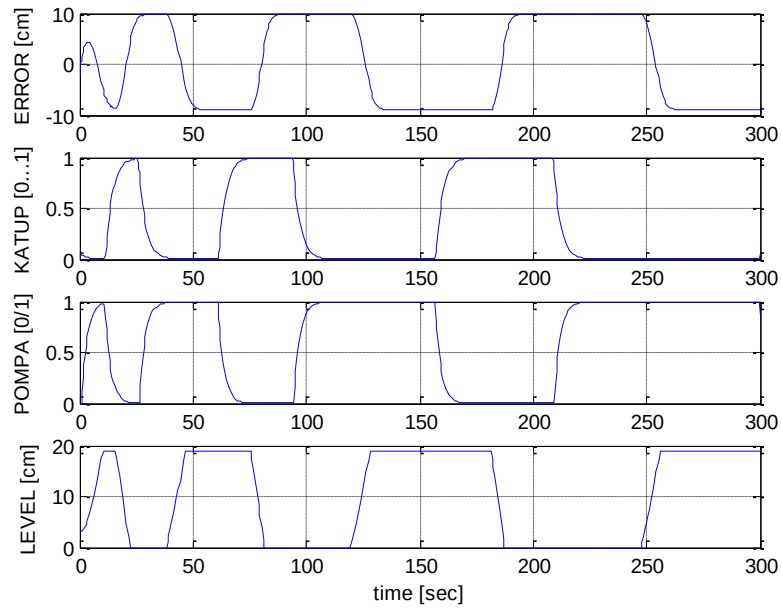
Hasil yang masih tidak stabil dan cenderung sama dengan kondisi sebelumnya. Pada kondisi ini pompa dan katup mengalami kondisi on off yang begitu tajam. Hal ini karena sifat dari K_p yang semakin memperngaruhi sistem yaitu kecepatan respon sehingga diperoleh hasil seperti gambar berikut ini. Namun terlihat bahwa error yang terjadi lebih kecil ketika $K_p=1$.



Pengendali PI

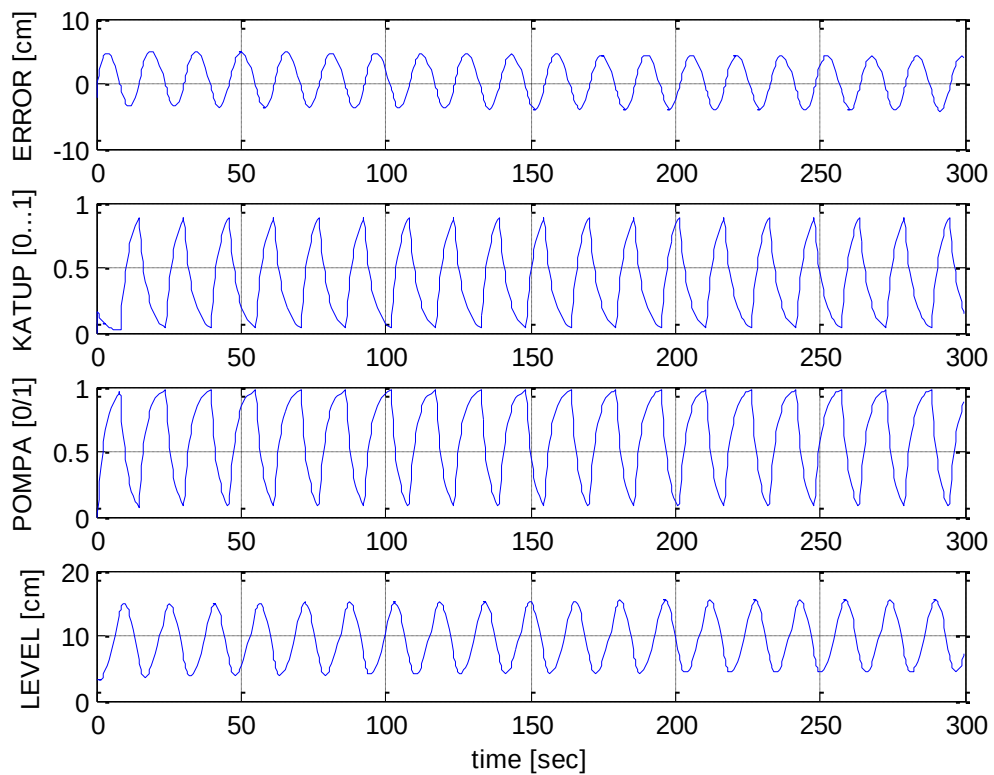
Pada simulasi ini ditambahkan pengendali I. Dari hasil simulasi diperoleh hasil bahwa pengendali ini cenderung mengalami Instability. Seperti yang dapat dilihat pada hasil data berikut ini.

- Ketika $K_p=1$; $K_i=2$
 Hasil yang dihasilkan sangat tidak stabil (Instability). Error yang terjadi sangat besar.. Pada level juga terjadi overflow disebabkan respon pompa dan katup yang begitu lambat. Hal ini disebabkan kurangnya nilai P.



- Ketika $K_p=100$; $K_i = 2$

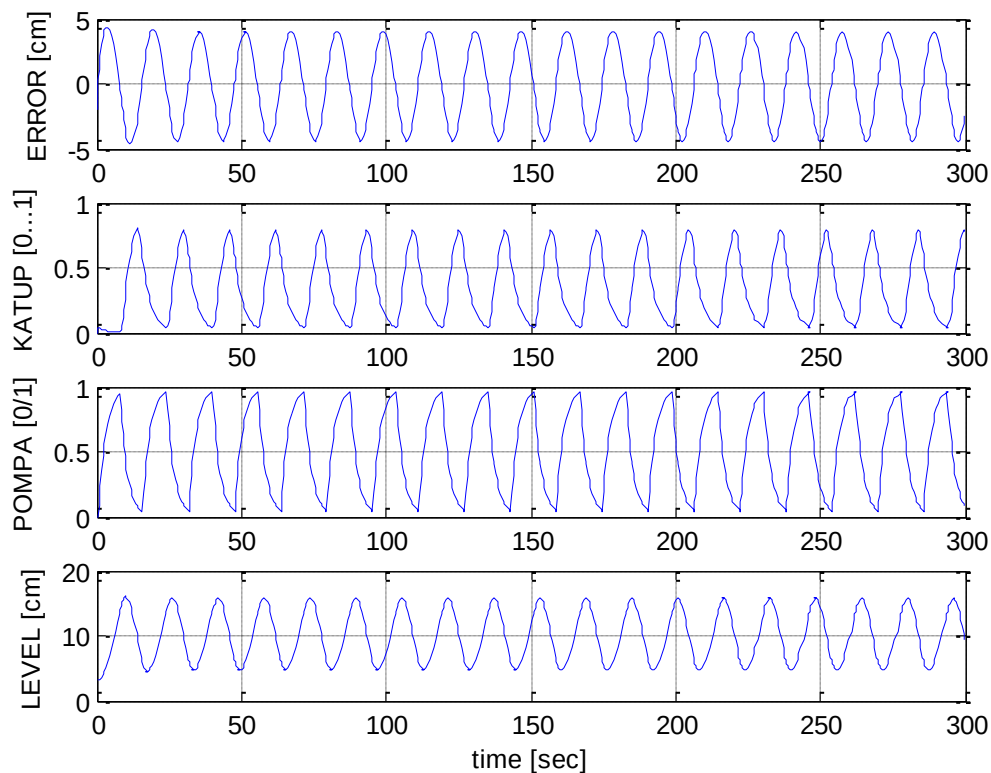
Dengan memperbesar pengendali P. Keluaran yang dihasilkan masih tidak stabil tetapi dalam kondisi ini overflow sudah tidak terjadi. Dan error mengecil. Respon katup dan pompa sudah baik.



- Ketika $K_p = 1$; $K_i = 180$

Dalam kondisi ini kondisi yang dihasilkan cenderung sama dengan kondisi sebelumnya walaupun tetap memiliki perbedaan diantaranya adalah kondisi buka

tutup katup sudah tidak begitu tajam. Hal ini karena pengaruh pengendali P sudah berkurang.

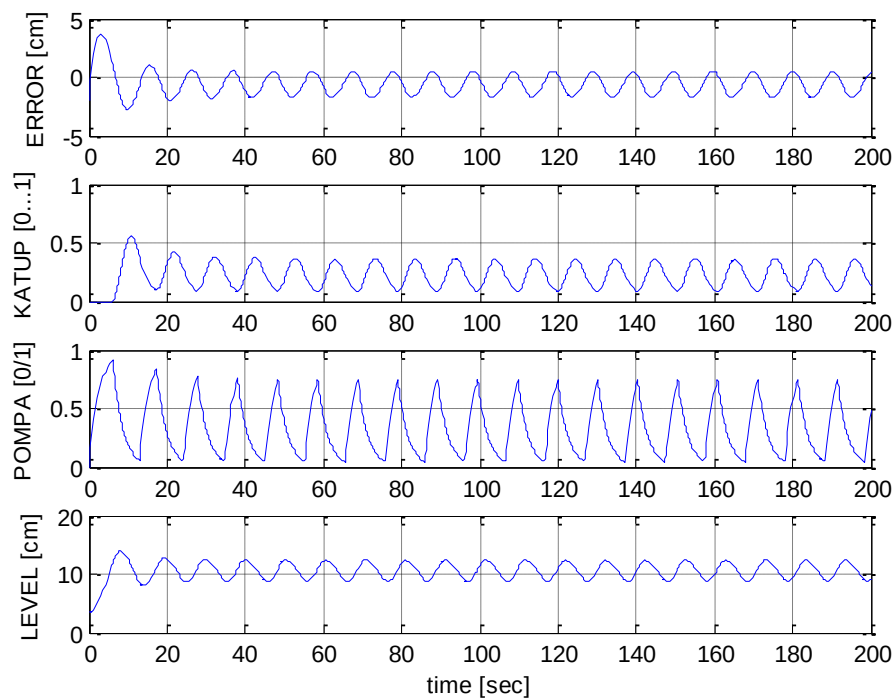


Pengendali PD

Pada simulasi ini pengendali yang digunakan adalah pengendali PD. Dari hasil simulasi ini dapat dilihat bahwa pengendali ini memiliki respon terhadap laju perubahan kesalahan yang menghasilkan koreksi yang berarti sebelum kesalahan semakin besar. Jadi efeknya adalah menghasilkan tindakan pengendalian yang cepat.

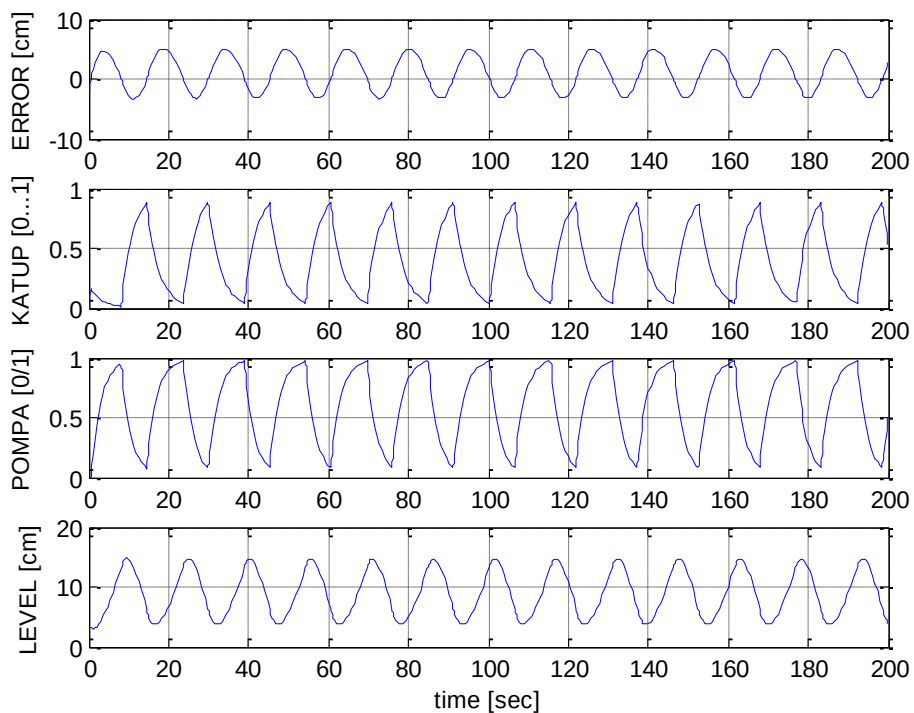
- Ketika $K_p = 1$; $K_d = 1$

Tiap-tiap variabel keluaran awalnya memiliki gerakan atau respon yang besar. Tapi kemudian mengalami peredaman dan cenderung berosilasi pada garis stabil. Hal ini karena adanya pengaruh dari pengendali D yang memiliki tindakan pengendalian yang cepat. Dari grafik terlihat bahwa errornya awalnya besar, kemudian mengalami penurunan. Errornya negatif ini berarti ketinggian yang dihasilkan lebih tinggi dari ketinggian yang seharusnya sesuai settingan. Terlihat bahwa katupnya hanya terbuka sedikit.



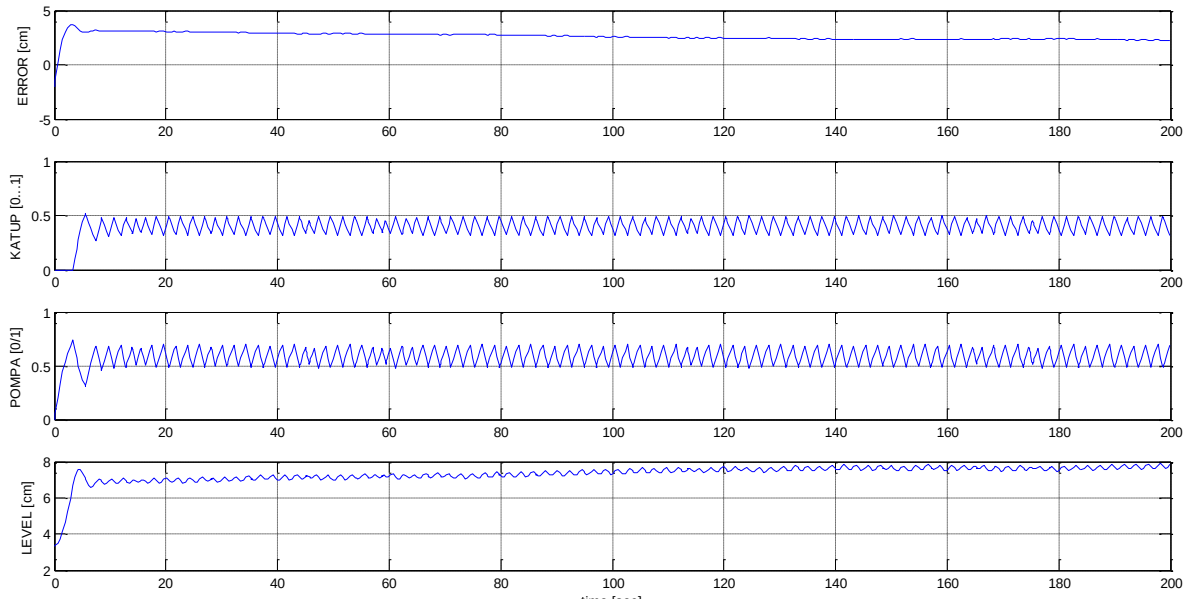
- Ketika $K_p = 270$; $K_d = 1$

Error tetap mengalami osilasi namun berada pada garis nol dengan osilasi yang cenderung tetap. Namun kerja pompa dan katup begitu tajam (cepat) sehingga level ketinggian air yang dihasilkan juga bertambah dan berkurang dengan cepat. Hal ini karena pengaruh pengendali P.



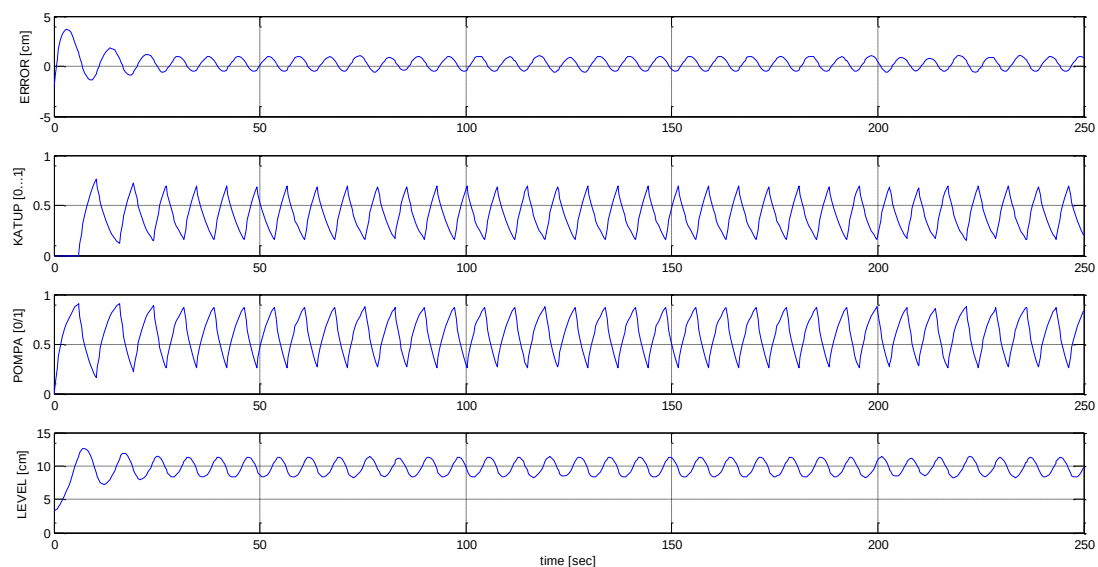
- Ketika $K_p = 1$; $K_d = 300$

Errornya positif. Berarti bahwa ketinggian yang dihasilkan lebih rendah dari ketinggian seharusnya. Terlihat bahwa untuk mengatasi hal tersebut katupnya mmbuka sedikit dan pompanya on cukupk besaar. Error cenderung menuju titik nol. Ini karena adanya pengaruh pengendali D yang memperbaiki kesalahan dengan cepat



- Ketika $K_p = 300$; $K_d = 300$

Nilai K_p kembali mempengaruhi kinerja sistem sehingga diperoleh hasil yang mirip ketika $K_p = 300$ dan $K_d = 1$.



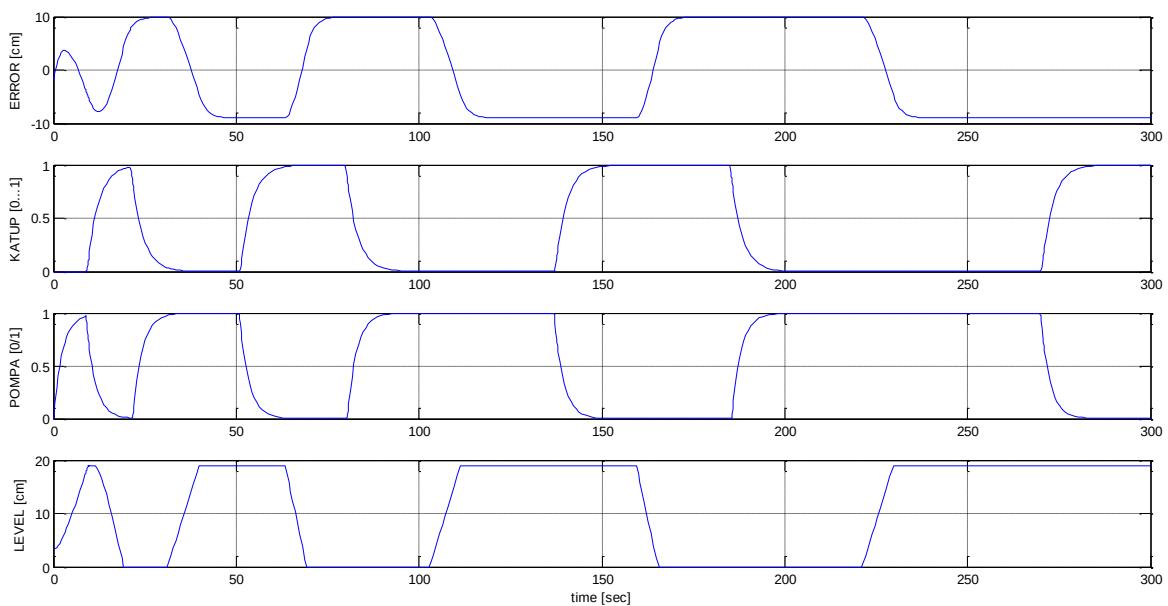
Pengendali PID

Pada simulasi ini Dari hasil simulasi ini digunakan pengendali P, I, dan D. Dari hasil simulasi diperoleh bahwa pengendali PID menggabungkan keunggulan-keunggulan yang dimiliki oleh

tiap-tiap pengendali. Pengendali P yang bersifat sebagai penguat yang mempercepat respon atau kinerja, pengendali I yang memperbaiki kesalahan akibat buruknya respon transien dan pengendali D yang memiliki respon yang cepat terhadap kesalahan sehingga kesalahan tidak akan membesar. Seperti yang dilihat pada pengambilan hasil data berikut ini.

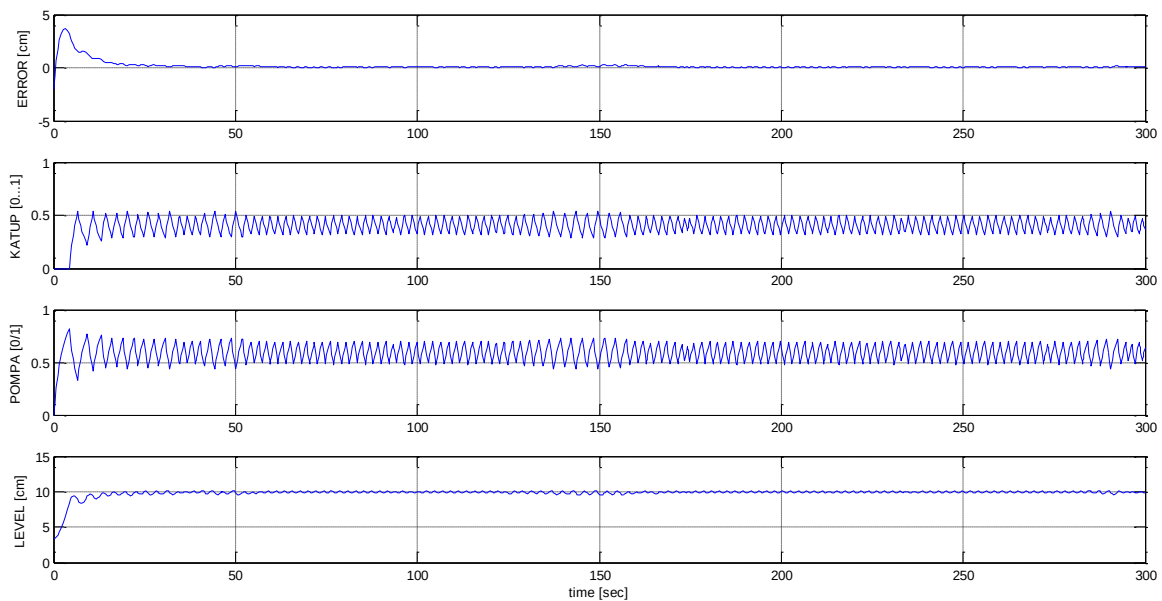
- Ketika $K_p = 1$; $K_i = 2$; $K_d = 1$

Pengaruh I sangat berpengaruh dalam sistem ini dapat dilihat respon semakin yang cenderung semakin lama. Sehingga level ketinggian air cenderung mengalami overflow dan kekosongan dalam waktu yang sangat lama. Errornya juga begitu besar



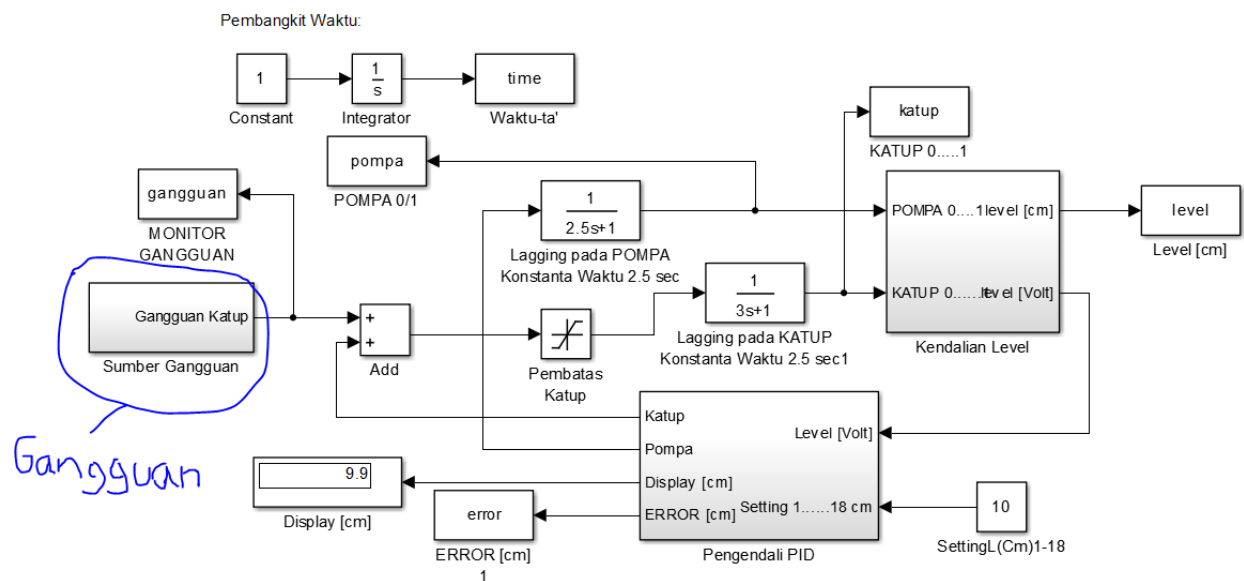
- Ketika $K_p = 200$; $K_i = 100$; $K_d = 1000$

Dengan pengaturan ini, nilai level ketinggian air pada tangki diperoleh level yang stabil sesuai dengan settingan. nilai K_p dibuat besar agar penguatan dari kecepatan respon dari kerja sistem tidak begitu besar namun tidak pula labat. Nilai K_i dibuat paling rendah yaitu sebesar 100 yang mempengaruhi perbaikan kesalahan akibat buruknya respon transien. Nilai K_d dibuat besar agar respon terhadap kesalahan semakin cepat. Terlihat bawah awalnya error besar namun diatasi oleh PID sehingga errornya semakin kecil. Dan diperoleh ketinggian yang stabil.

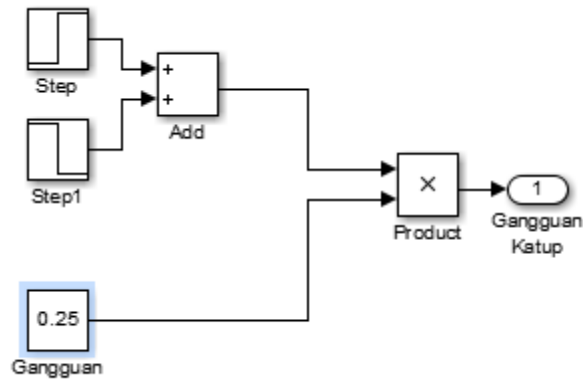


- SIMULASI WATER LEVEL CONTROL DENGAN PEMBERIAN GANGGUAN**

Pada percobaan ini, dilakukan pengendalian dengan PID tapi dengan menggunakan gangguan. gangguan diberikan pada katup dan gangguan dapat dimonitor. Berikut gambar simulasinya dengan matlab:

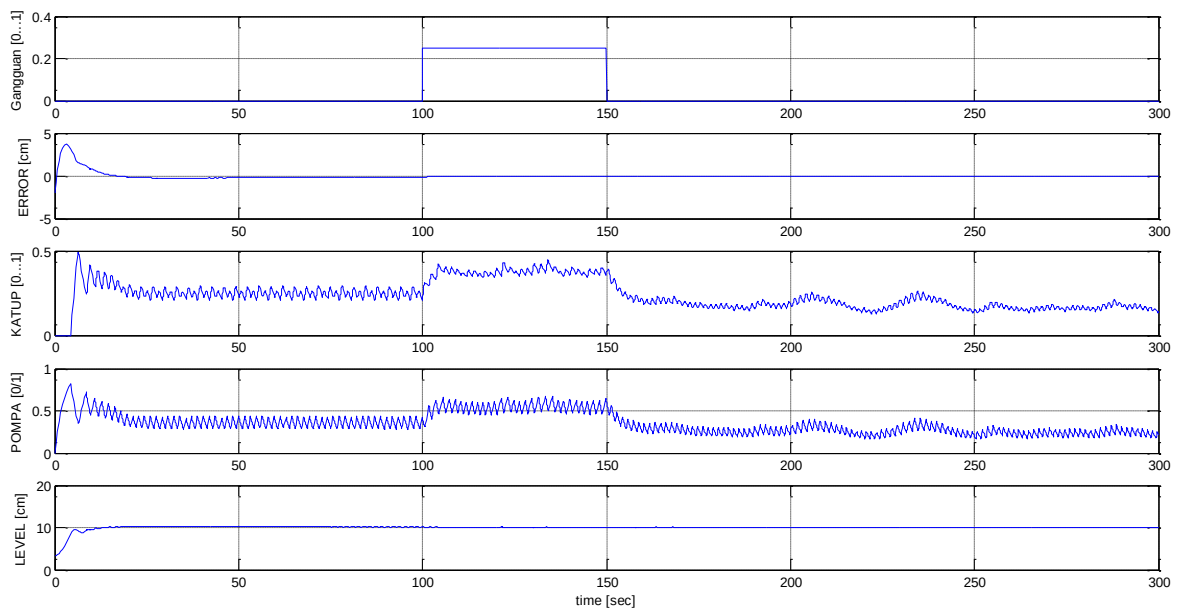


Gambar Simulasi dengan Gangguan



Gambar Subsistem Gangguan

Subsistem gangguan yang diberikan pada katup seperti pada gambar diatas. Dimana gangguannya berasal dari sinyal step detik n ke n . Sinyal step ini dikalikan dengan 0.25. hasil kali ini adalah gangguannya. Gangguan ini ditambahkan pada nilai output katup dari pengendali PID sehingga dengan demikian akan mempengaruhi kinerja katup dan pompa dan tentu juga error. Dengan menggunakan PID, maka gangguan ini diminimalisir. Berikut hasil simulasinya:



Pada pengendali PID. Dilakukan pengaturan nilai $K_p = 10$; $K_i = 10$; $K_d = 50$. Dengan pengaturan ini, diperoleh nilai level yang seimbang. Nilai K_d dibuat sangat besar. Hal ini agar kesalahan yang terjadi segera ditangani.

Dari gambar diatas terlihat bahwa gangguan terjadi dari detik ke 100 sampai 150 dengan nilai gangguan sebesar 0.25. sesuai dengan yang diset pada subsistem Gangguan. Dengan pemberian gangguan terlihat bahwa pada katup dan pompa terjadi perubahan nilai saat gangguan. Namun dengan pengendali PID. Error yang terjadi mendekati nol sehingga

level hampir seimbang. Nilai error ini ditangani dengan pemberian K_d sangat besar yaitu 50. Sehingga pengendali derivatif akan mengatasi error ini.

KESIMPULAN

Dari simulasi-simulasi yang dilakukan, pada saat simulasi tanpa pengendali. Ketinggian air diatur oleh pompa dan katup dimana settingannya harus dilakukan secara manual. Sedangkan simulasi dengan pengendali dan dengan gangguan, secara otomatis katup dan pompa akan terset sesuai settingan ketinggian level. Dan dihasilkan level yang stabil dengan mengatur nilai pengendali Proportional K_p , Integral K_i dan nilai derivatif K_d .

BAB II

SILO 1 TO SILO 2 MENGGUNAKAN CONVEYOR

PENDAHULUAN

Sistem silo ke silo menggunakan conveyor merupakan sistem yang umum dalam sistem produksi industri seperti industri batubara, emas, semen dan sebagainya. Silo adalah tempat penampungan bahan baku dan produk yang dihasilkan oleh proses industri. Dan conveyor berfungsi sebagai peralatan yang digunakan untuk memindahkan material secara kontinu dengan jalur yang tetap.

Pada proyek ini, dilakukan simulasi dari model Silo 1 ke Silo 2 menggunakan conveyor. Simulasi dilakukan menggunakan bantuan matlab dan simulink. Pada proyek ini dilakukan simulasi agar perpindahan dari silo ke silo, tidak ada bahan baku material hilang atau rusak

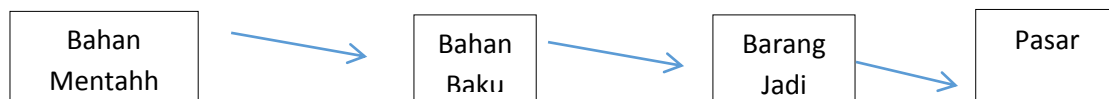
TUJUAN

1. Mengetahui Prinsip Kerja Sistem Silo ke Silo dengan menggunakan konveyor.
2. Mengetahui parameter yang berpengaruh pada pengiriman material.
3. Menentukan parameter material dari silo ke silo sehingga tidak terjadi penurunan produksi.

DASAR TEORI

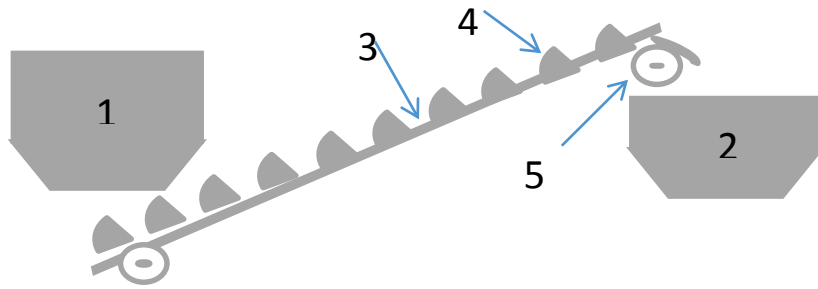
1. Model Silo 1 ke Silo 2

Proses teknologi Industri di industri manufactur dapat kita melihat bagan dibawah ini:



Bahan mentah diperoleh dari Sumber Daya Alam. Dari bahan mentah tersebut diekstraksi menjadi bahan baku kemudian di proses. Lalu mengalami proses yang panjang sehingga menjadi barang jadi, produk yang di hasilkan kembali di reycle jika sudah tidak bisa terpakai. Proses industry ada dua yakni secara fisika dan kimia, suatu materi mengalami perubahan fisika, adalah perubahan zat yang bersifat sementara, seperti perubahan wujud, bentuk atau ukuran. Perubahan ini tidak menghasilkan zat baru. Perubahan kimia merupakan yang bersifat kekal dengan menghasilkan zat baru. Perubahan kimia disebut juga reaksi kimia

Gambar dibawah adalah model silo ke silo yang akan disimulasikan. Model ini merupakan model dari alat yang ada di laboratorium Energi Listrik jurusan Elektro Fakultas Teknik Uniersitas Hasanuddin:



Dengan keterangan :

1. Silo 1
2. Silo2
3. Konveyor
4. Belt / Cawang
5. Motor

Dengan spesifikasi sebagai berikut :

a. SILO 1

- Diameter bagian Tabung = 26 cm
- Tinggi bagian tabung = 20,5 Cm
- Tinggi Bagian kerucut = 12 cm
- Diameter kerucut yang kecil = 4,5 cm
- Diameter kerucut besar = diameter tabung

b. Silo 2

- Diameter bagian Tabung = 26 cm
- Tinggi bagian tabung = 10 cm
- Tinggi Bagian kerucut = 12 cm
- Diameter kerucut yang kecil = 4,5 cm
- Diameter kerucut besar = diameter tabung

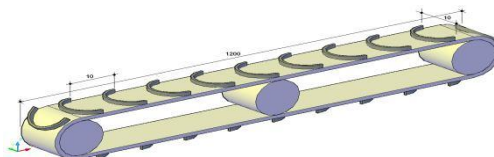
c. Konveyor

Konveyor terdiri dari 15 cawang antara Silo 1 dan Silo 2

2. Silo dan Conveyor

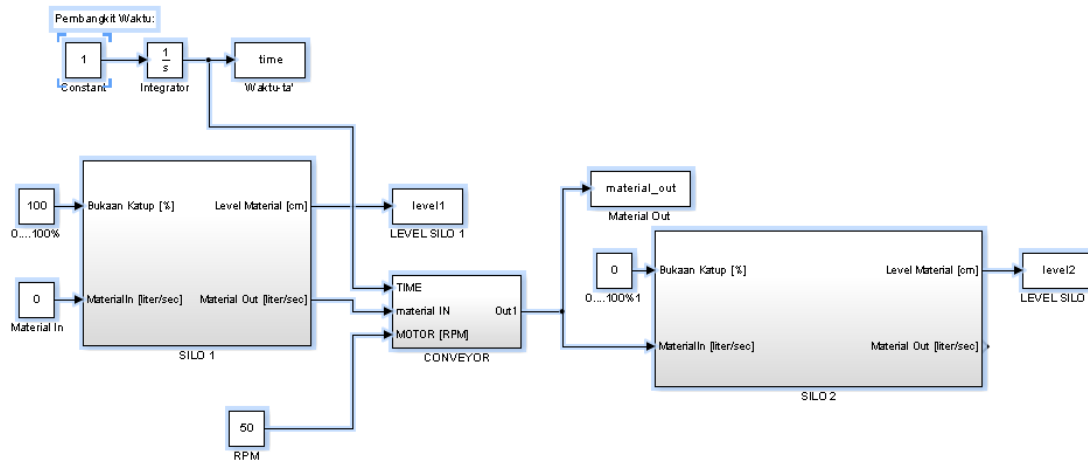
Silo adalah tangki penampungan bahan baku dan produk yang dihasilkan oleh proses industri. Pada proyek ini menggunakan 2 silo, yaitu silo 1 dan silo 2 dengan ukuran yang telah dipaparkan sebelumnya.

Conveyor adalah alat untuk mengangkut material. Pada proyek ini, conveyor mengangkut material dari silo 1 ke silo 2. Conveyor terdiri dari 15 cawang.



ANALISA DAN HASIL PENGAMATAN

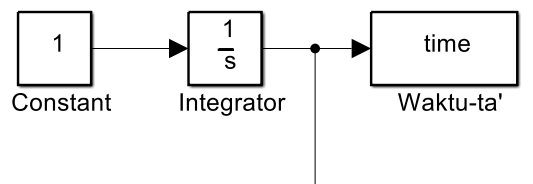
Sistem ini dimodelkan dengan menggunakan program Matlab 2013 simulink, dengan tiga bagian utama yaitu pembangkit waktu, Subsystem Silo1, Subsystem Silo2 dan Subsystem Conveyor seperti pada gambar dibawah:



Gambar 2. Model Sistem Silo to Silo

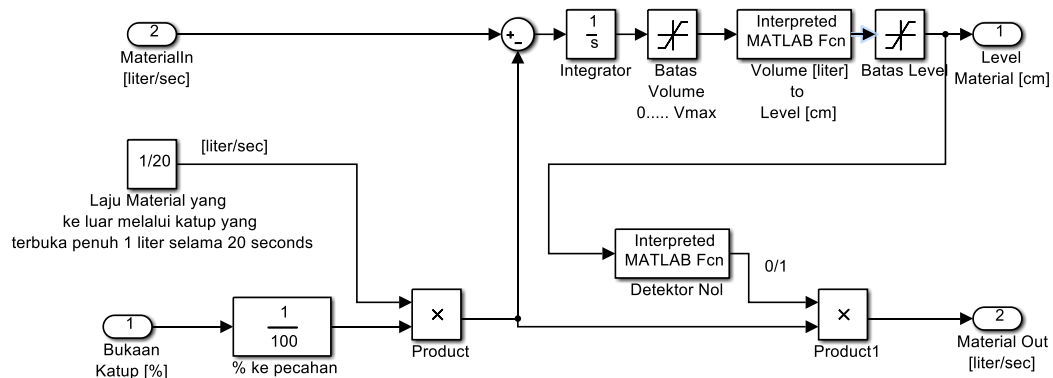
Dari gambar terlihat bahwa Silo 1 dan silo 2 variabel inputnya adalah bukaan katup dan material in. Dan outputnya adalah level material dan material out. Pada conveyor inputnya berupa waktu, material in dan motor.

Pembangkit Waktu:



Gambar 3 : Pembangkit Waktu

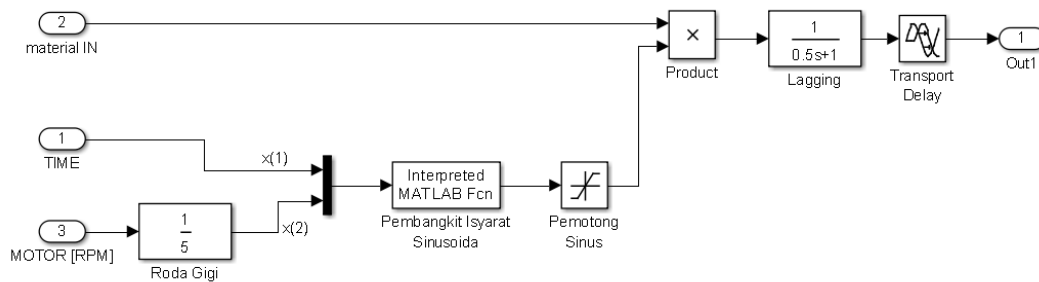
Pada bagian subsistem diatas terlihat bahwa pembangkit waktu dijadikan sebagai parameter waktu untuk melakukan perhitungan. Nilai constant nya adalah 1. Parameter ini digunakan oleh sistem pada besaran-besaran yang membutuhkan besaran waktu. Misalnya dalam perhitngan frekuensi, perioda, atau dalam pembangkit sinyal sinus.



Gambar 4. Subsistem Silo 1

Pada gambar subsistem silo 1 diatas terlihat bahwa ada 2 input yaitu material dengan satuan liter/sec dan bukaan katup dalam %. Untuk output ada 2, yaitu level material dalam

(cm) dan material out dalam liter/second. Jumlah material pada silo 1 diatur pada material in. laju serta bukaan katup mempengaruhi material out.

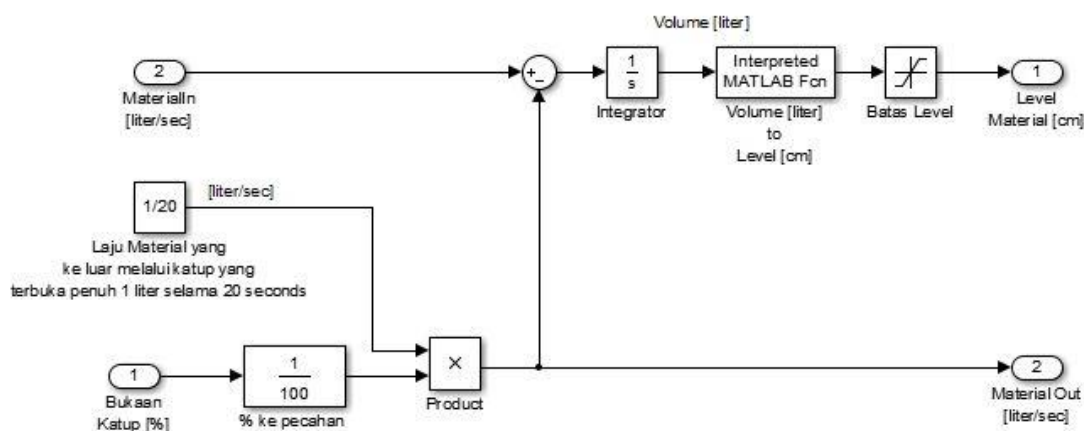


Gambar 4. Subsistem conveyor

Conveyor merupakan pengangkut material dari silo 1 ke silo 2. Konveyer akan membawa material dengan menggunakan cawang yang terpasang pada conveyor. Cawang ini sebagai penampung material yang akan ditumpahkan pada silo 2..

Untuk membentuk model yang dapat mewakili cara kerja dari conveyor, maka dibutuhkan 3 buah input, dan 1 output. Inputan terdiri dari input material, pewaktuan (time), dan kecepatan motor. Ketiga inputan inilah yang membuat sistem dapat bekerja. Sedangkan output hanya berupa material. Walaupun jika lebih diperinci, output dari sistem ini bisa lebih dari 1.

Untuk menjalankan conveyor dibutuhkan sebuah penggerak, maka dimasukkanlah inputan berupa kecepatan motor. Dalam pemodelan conveyor, waktu dibutuhkan sebagai besaran untuk membangkitkan gelombang sinus. Fungsi gelombang sinus untuk memodelkan bentuk cawang yang ada pada conveyor. Selanjutnya, material yang dibawah oleh conveyor. Untuk melakukan pemodelan sinyal sinus yang telah dipotong, kemudian dikalikan dengan material input. Sehingga dari sinilah akan didapatkan jumlah material yang terbawa pada setiap cawang yang ada pada conveyor.



Pada gambar subsistem silo 2 diatas terlihat bahwa ada 2 input yaitu material dengan satuan liter/sec dan bukaan katup dalam %. Untuk output ada 2, yaitu level material dalam (cm) dan material out dalam liter/second. Jumlah material pada silo 2 berasal dari conveyor dan bukaan katupnya diset 0

ANALISA DAN HASIL PENGAMATAN

Pengamatan dilakukan dengan merubah variabel yang ada pada koneyor. Perubahan dilakukan untuk mengamati sejauh mana pegaruh variabel tersebut terhadap efisiensi conveyor. Ada dua variabel yang diubah pada konveyer yaitu kecepatan motor dan tinggi cawang. Untuk mengubah tinggi cawang, maka amplitudo pada pembangkit sinyal sinus yang akan diubah.

Hasil yang diperoleh sebagai berikut:

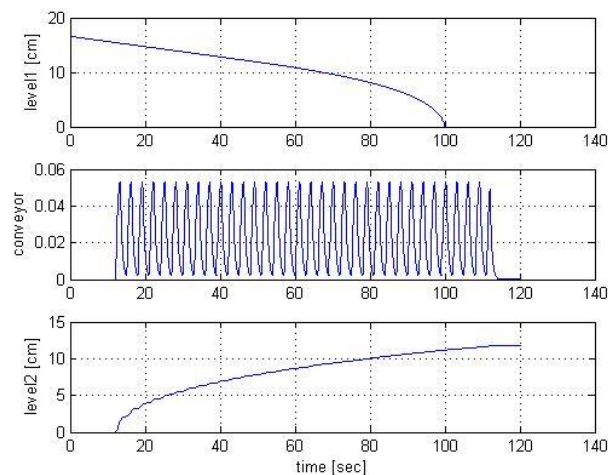
- **Mengubah-ubah Kecepatan Conveyor :**

Volumer mterial silo 1 = 5 liter

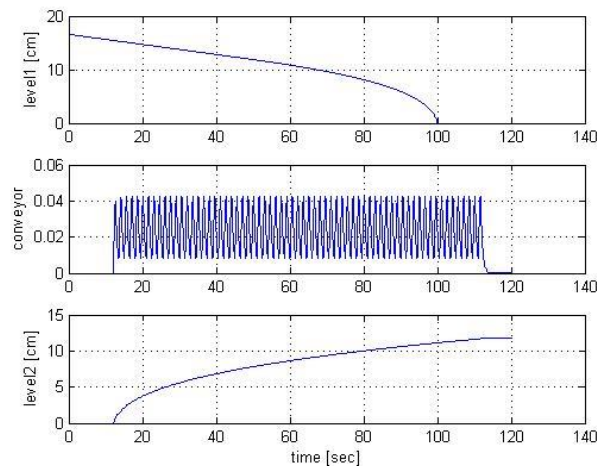
Level Silo 1 : 16.6053

Tinggi Cawang (amp. Pembentuk sinus) : 2

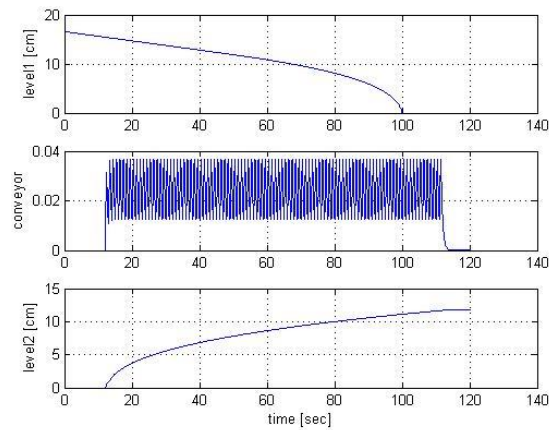
- 100 RPM



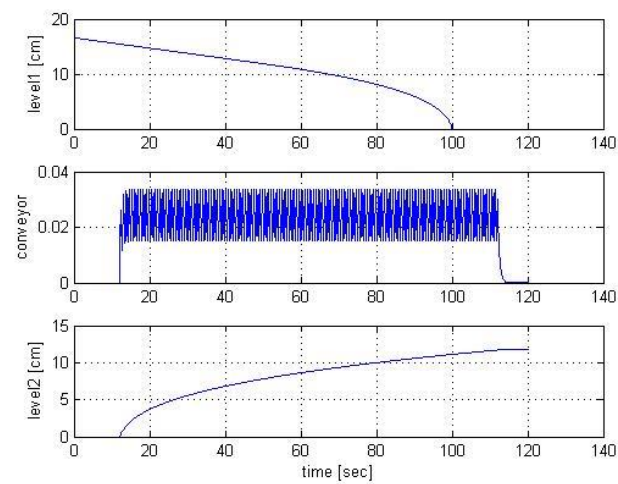
- 200 RPM



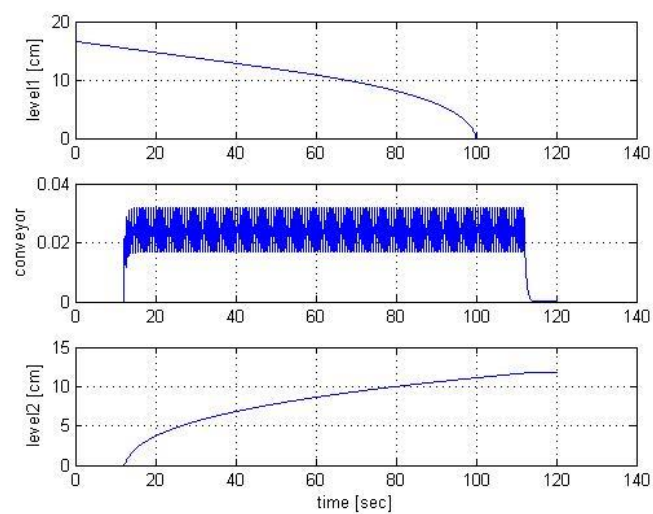
- 300 RPM



- 400 RPM



- 500 RPM



Dari grafik-grafik diatas terlihat bahwa untuk level pada silo 1 menunjukkan penurunan linear dai detik 0 sampai detik 80. Ini terjadi disaat material pada bagian tabung silo yang berkurang. Pada detik 81 sampai 120 terjadi penurunan yang non linear.

Ini terjadi saat material pada bagian kerucut silo yang berkurang. Begitupun pada level silo 2 terjadi delay disebabkan awal pengangkutan material dari silo 1 ke silo 2. Pada detik 15 samai detik 40 terjadi kenaikan non linear akibat yang disii kerucutnya silo2. Detik 41 sampai 120 yang diisi adalah tabung silo2.

Untuk perubahan pada variabel kecepatan, perubahan yang didapat tidak begitu signifikan. Jika kita amati data pada variabel tersebut, terdapat kecendrungan bahwa semakin kecepatan ditambahkan level material yang pada silo 2 semakin menurun. Sehingga yang terlihat bahwa semakin cepat putaran motor, maka akan menyebabkan material yang diangkut semakin banyak yang hilang. Walaupun perubahan berada pada rentang nilai yang kecil, namun jika pada material dengan jumlah yang besar nilai ini tentu sangat berpengaruh pada hasil produksi dan efisiensi Alat.

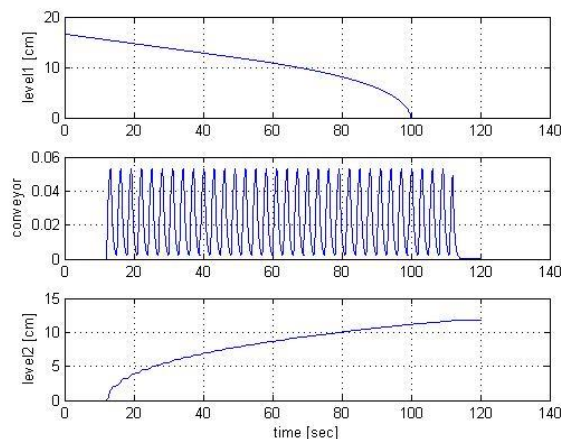
- **Mengubah-ubah Tinggi Cawang**

Volumer mterial silo 1 = 5 liter

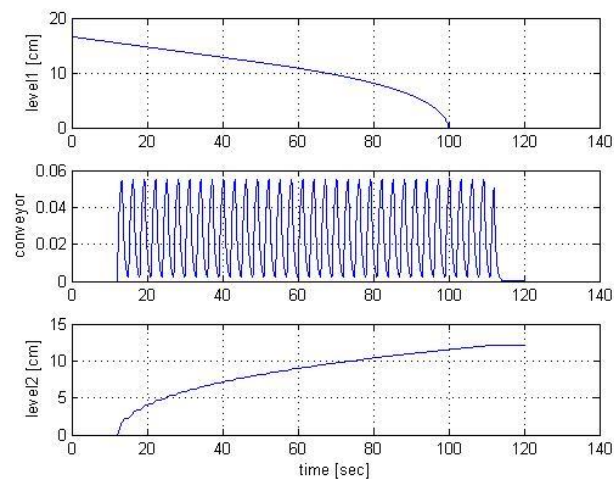
Level Silo 1 : 16.6053

Kecepatan motor : 100 RPM

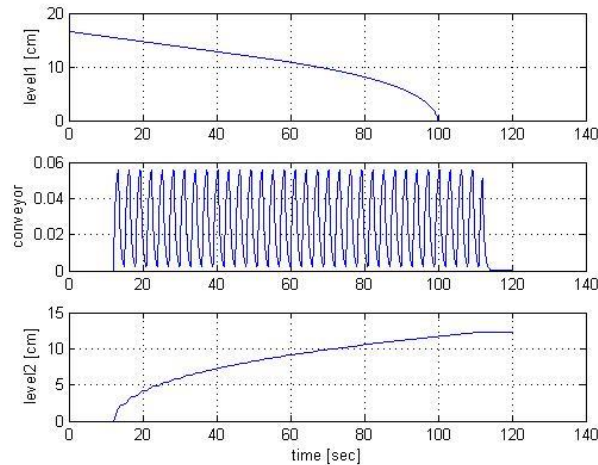
- 2



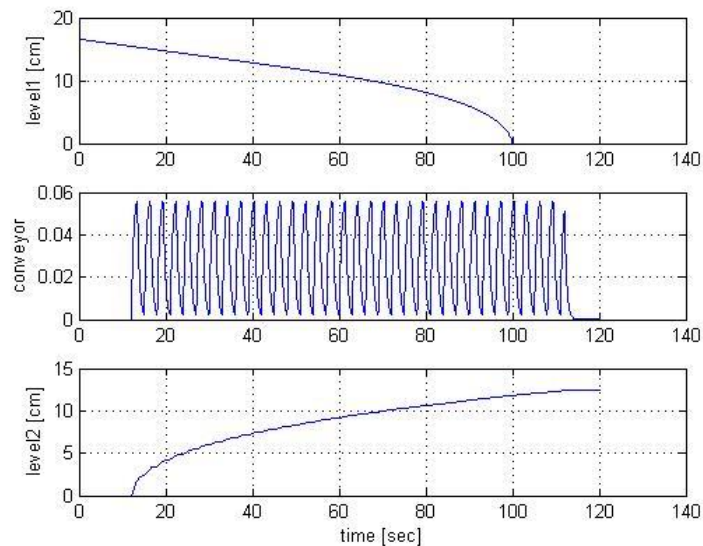
- 3



- 4



- 5



Sama halnya saat tinggi cawang yang diubah. Dari grafik-grafik diatas terlihat bahwa untuk level pada silo 1 menunjukkan penurunan linear dari detik 0 sampai detik 80. Ini terjadi disaat material pada bagian tabung silo yang berkurang. Pada detik 81 sampai 120 terjadi penurunan yang non linear. Ini terjadi saat material pada bagian kerucut silo yang berkurang. Begitupun pada level silo 2 terjadi delay disebabkan awal pengangkutan material dari silo 1 ke silo 2. Pada detik 15 samai detik 40 terjadi kenaikan non linear akibat yang disii kerucutnya silo2. Detik 41 sampai 120 yang diisi adalah tabung silo2.

Untuk perubahan pada amplitudo sinyal sinus, atau dalam hal ini, nilai kemiringan cawang pada conveyor, data yang diperoleh sebagaimana yang diperlihatkan oleh tabel diatas. Dari data tersebut, terlihat ada perbedaan antara kecepatan dan amplitudo. Jika pada data kecepatan kecendrungan dari data tersebut berbanding terbalik, sedangkan pada amplitudo data cenderung berbanding lurus. Dengan kecepatan motor yang tetap , semakin tinggi cawang semakin sedikit material yang diangkut hilang.

KESIMPULAN

Prinsip kerja sistem ini sangat ditentukan oleh kualitas conveyor. Kecepatan motor conveyor mengurangi efisiensi dari conveyor, sedangkan tinggi cawang dapat meningkatkan efisiensi conveyor.

TEKNOLOGI KENDALI PROSES



DISUSUN OLEH:

MUH. IHSAN SHALIHIN (D41112256)

UMAR HASAN (D41112303)

TEKNIK KOMPUTER, KENDALI, DAN ELEKTRONIKA

JURUSAN ELEKTRO FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

BAB I

PENDAHULUAN

1.1. LATAR BELAKANG

Laporan ini dibuat untuk memenuhi syarat melulusi matakuliah Teknologi Kendali Proses. Sebagaimana kita ketahui, teknologi berkembang dengan pesatnya pada tiap tahunnya. Kita tidak hanya dituntut untuk mengetahui bidang ilmu kita, khususnya sebagai mahasiswa Elektro. Kita juga dituntut untuk mempelajari ilmu-ilmu lain khususnya yang berkaitan dengan ilmu Teknik Elektro. Dalam mata kuliah ini kita mempelajari bagaimana itu proses fisika dan proses kimia terjadi dalam suatu industri. Dalam makalah ini kita mempelajari atau membahas tentang bagaimana itu proses Teknolgi Kendali Proses dan bagaimana keluarannya juga bagaimana membuatnya stabil.

1.2. TUJUAN

Praktikum dari matakuliah ini bermaksud agar kita mampu membuat suatu model sistem fisik dan mengubahnya kedalam model yang dapat dianalisis, sehingga dapat dianalisis kestabilan dan respon keluarannya. Dalam hal ini akan dilakukan dengan bantuan software Simulink MATLAB.

BAB II

CONTROLLED LEVEL TANK

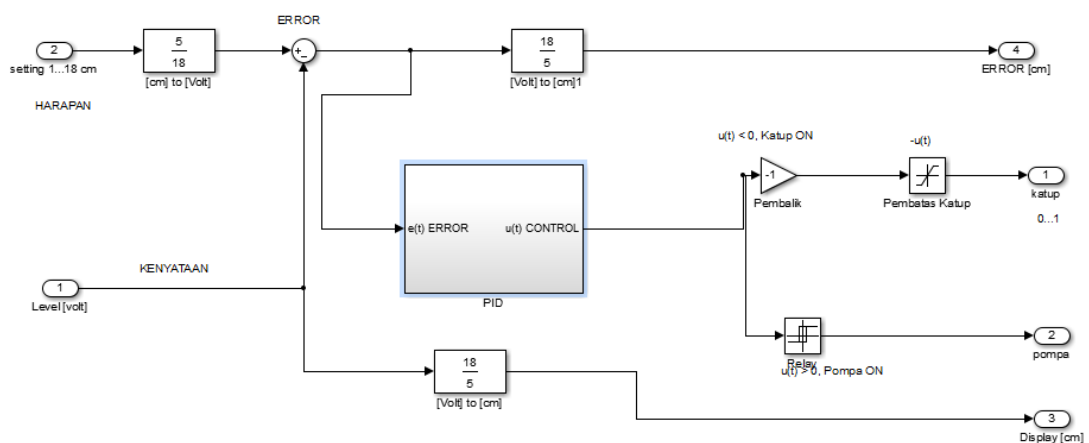
Level control adalah sebuah proses yang menjaga kestabilan level air dalam tangki penampungannya. Sebuah pompa untuk memompa air dari bawah ke dalam tangki penampungan, juga katup untuk mengalirkan keluar air berlebih dari dalam tangki, dan sensor tekanan yang berfungsi mengukur ketinggian air didalam tangki.

Berikut ini adalah data-data teknis yang dibutuhkan dalam perancangan model proses:

- Tangki Kendalian Level
- Tangki
- Pompa
- Katup Proporsional
- Sensor Tekanan

Diasumsikan bahwa tinggi tangki adalah 20 cm dengan tinggi pipa pembuangan overflow adalah 18 cm.

Dapat diamati bahwa sistem alat ini mempunyai dua blok, blok pengendali dan blok kendalian. Pengendali akan mendapat isyarat setpoint, yang berupa nilai pengaturan ketinggian air yang diinginkan. Pada pengendali, dibutuhkan dua buah masukan sebagai parameter acuannya, yaitu setpoint dan level ketinggian air yang telah dicapai. Pada kendalian didalam sistem disamping, kendalian akan menerima masukan berupa isyarat kendalian dari pengendali, yaitu isyarat kendalian pompa katup. Satu-satunya keluaran dari kendalian adalah hasil dari pembacaan sensor ketinggian dengan satuan volt.



1. KENDALIAN LEVEL

Kendalian atau Plant memiliki dua masukan (isyarat kendalian dan dua buah keluaran: feedback dari sensor dan monitor).

Pompa

Pompa memiliki isyarat kendalian dari 0 sampai 1. Pompa yang digunakan memiliki kecepatan aliran maks. 8 liter/menit.

Pompa diputar oleh motor yang dikontrol oleh sebuah relay, maka pada pompa akan berlaku suatu model hysteresis (kondisi saat pompa tidak memiliki titik yang sama untuk on atau off dari keadaan sebelumnya).

Digunakan sebuah blok transfer function untuk membuat lagging dengan time constant τ dengan persamaan $\frac{1}{\tau s + 1}$.

Keluaran dari blok tranfer function ini dapat dipasang variabel monitoring “pompa” untuk mengetahui respon dari kecepatan motor pompa.

Kita dapat menggunakan sebuah blok product dengan keluaran berupa kecepatan aliran yang dipompa kedalam tangki dalam satuan liter/detik. Diasumsikan $\tau = 2.5$, blok diagramnya sebagai berikut:

Katup

Katup akan beroperasi dengan rentang isyarat kendalian dari 0 sampai 1.

Seperti pada motor, katup juga tidak dapat langsung menutup dan membuka sekat berdasarkan isyarat kendaliannya.

Katup dapat membuang fluida dengan debit maksimum 0.7 m³/hours, dan isyarat kendalian yang telah keluar dari blok lagging akan menghasilkan isyarat dari 0 sampai 1.

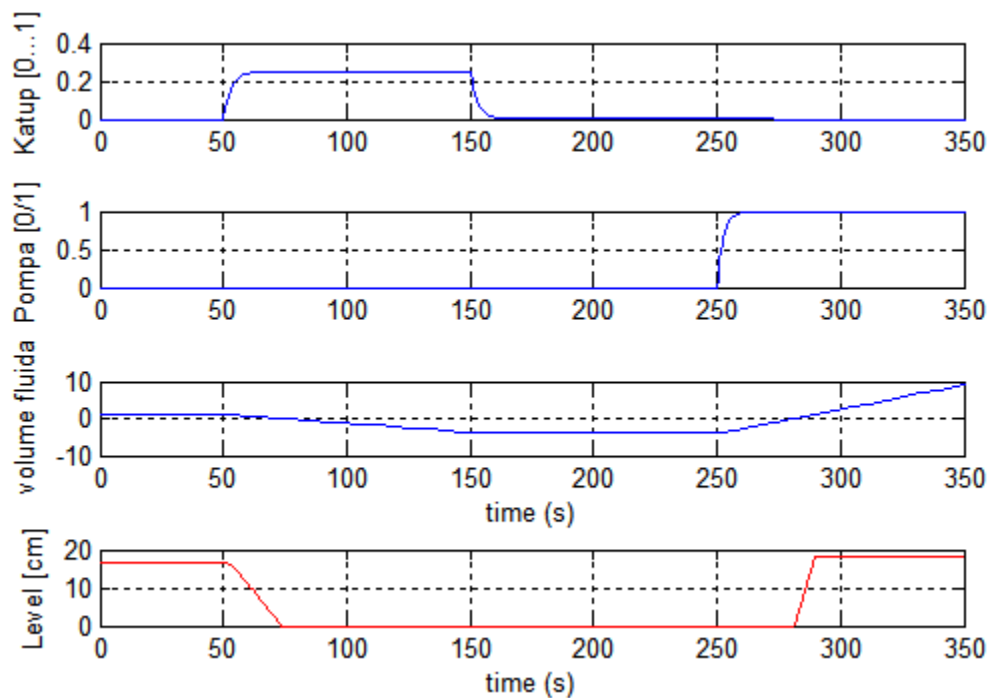
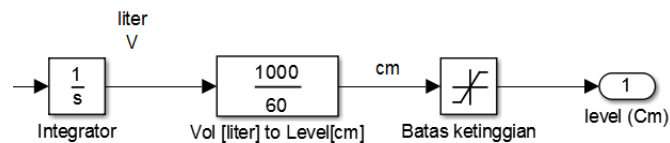
Untuk noise generator, tujuannya adalah menghasilkan gangguan berbentuk pulsa yang dapat diaktifkan selama waktu tertentu, misalnya dari 100 sampai 150 detik dengan amplitudo 0.25.

Level controlled tank

Level controlled tank merupakan tangki penampungan yang akan diatur ketinggiannya.

Hasil dari penjumlahan memiliki satuan liter/detik –turunan volume terhadap waktu–. Sehingga untuk mendapatkan volume sesaat pada tangki maka dilakukan pengintegralan sehingga menghasilkan satuan volume.

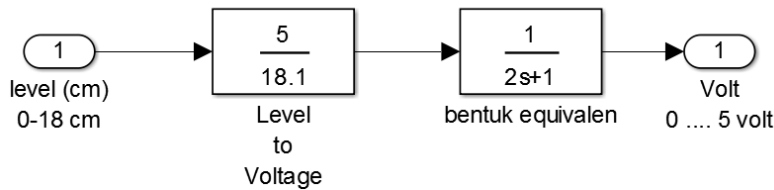
Pada blok diagram yang menggunakan saturation setelah blok integrator, volume akan tetap bernilai negatif, dan level air akan mulai bertambah saat volume ini telah melewati titik nol. Blok integrator with saturation akan diberikan batas bawah 0 dan batas atas 1.086. Pada blok ini juga dapat diisi initial condition, yakni kondisi awal dari tangki dalam satuan liter (misalnya 1).



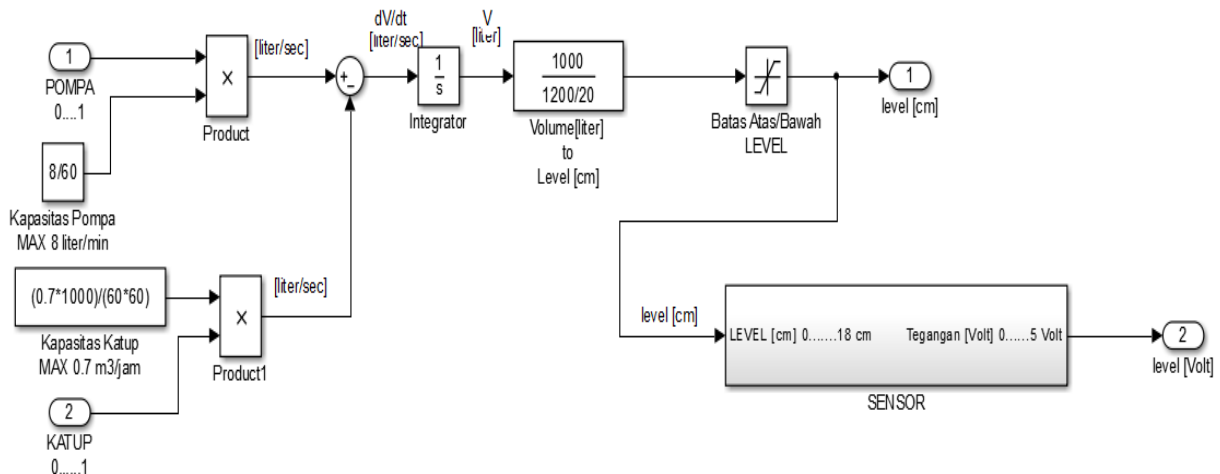
Sensor

Pada sensor ketinggian maksimum adalah 18.1 cm, dan akan dikonversi dalam satuan volt dari 0 sampai 5, maka dari satuan cm dapat diubah ke volt dengan mengalikannya dengan sebuah blok transfer function $\frac{5}{18.1}$.

Blok diagram sensor ditunjukkan pada gambar berikut:



Blok kendalian level:



PENGENDALIAN PID

Pengendalian PID memberikan isyarat kendalian pada kendalian agar level ketinggian fluida pada tangki tetap stabil sesuai dengan nilai setpoint. Masukan dari pengendali ada dua, yaitu setpoint dan hasil pembacaan sensor dalam volt. Sedangkan galat (error) adalah selisih antara harapan (setpoint) dan kenyataan(hasil pembacaan).

Display menunjukkan hasil pembacaan sensor dalam satuan centimeter. Rentang keluaran sensor adalah 0 sampai 5 volt untuk pembacaan 0 sampai 18.1 cm, maka perlu diberikan blok pembalik dari volt ke cm, yaitu dengan memberi nilai $\frac{18.1}{5}$.

Hasil pembacaan sensor adalah dalam satuan volt, maka untuk mencari error atau galat satuan perlu dikonversi menjadi satuan yang sama. Yang perlu diperhatikan juga adalah pembatasan pada setpoint, yang diatur berlebih, yang mengakibatkan pompa terus on karena tinggi fluida pada tangki tidak akan mungkin melebihi tinggi pipa pembuangan overflow tersebut. Dari masukan setpoint dipasang sebuah blok saturation yang nilainya adalah 0 sampai 18.1 cm.

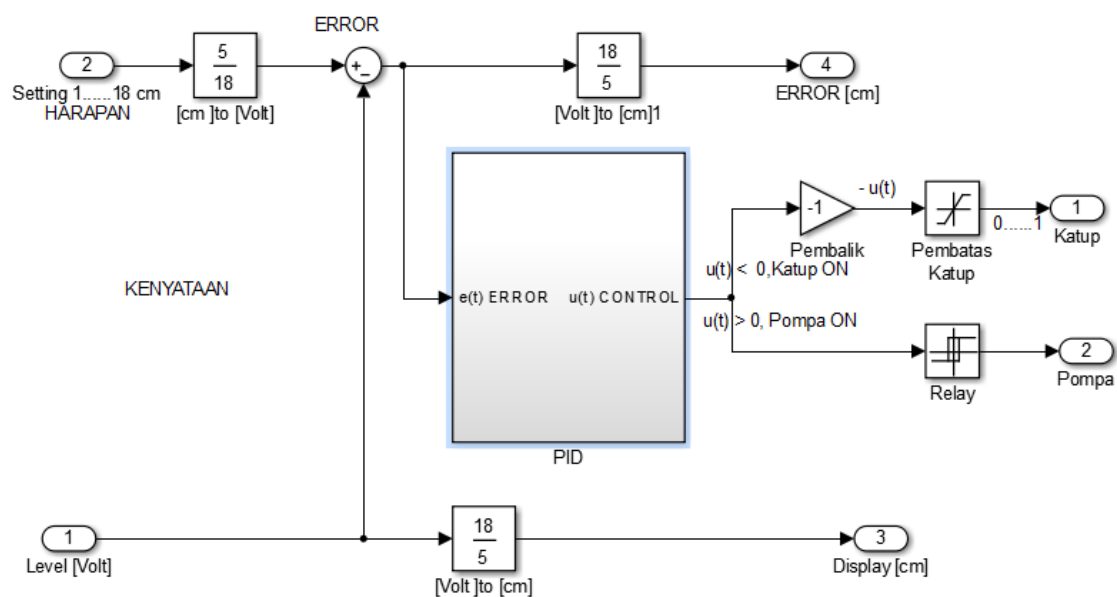
Harapan dan kenyataan haruslah dalam satuan yang sama agar error (volt) bisa didapatkan dengan menghitung selisih antara keduanya. Apabila bernilai positif, artinya ketinggian fluida

kurang dari harapan, dan jika hasilnya negatif, artinya ketinggian fluida melebihi harapan (katup perlu dibuka).

PID ditambahkan untuk pengendalian pompa dan katup. Keluaran dari pengendali apabila bernilai positif akan menggerakkan pompa dan apabila bernilai negatif akan menggerakkan katup.

Untuk membuat katup tidak terlalu rentan terhadap gangguan, batasan atas yang dimasukkan adalah 5-10. Batasan atas juga tidak boleh terlalu tinggi, sebab efek lagging pada katup dapat menyimpan isyarat yang terlalu besar sehingga dapat menyebabkan sistem malah menjadi lebih tidak stabil.

Blok diagram PID secara lengkap:



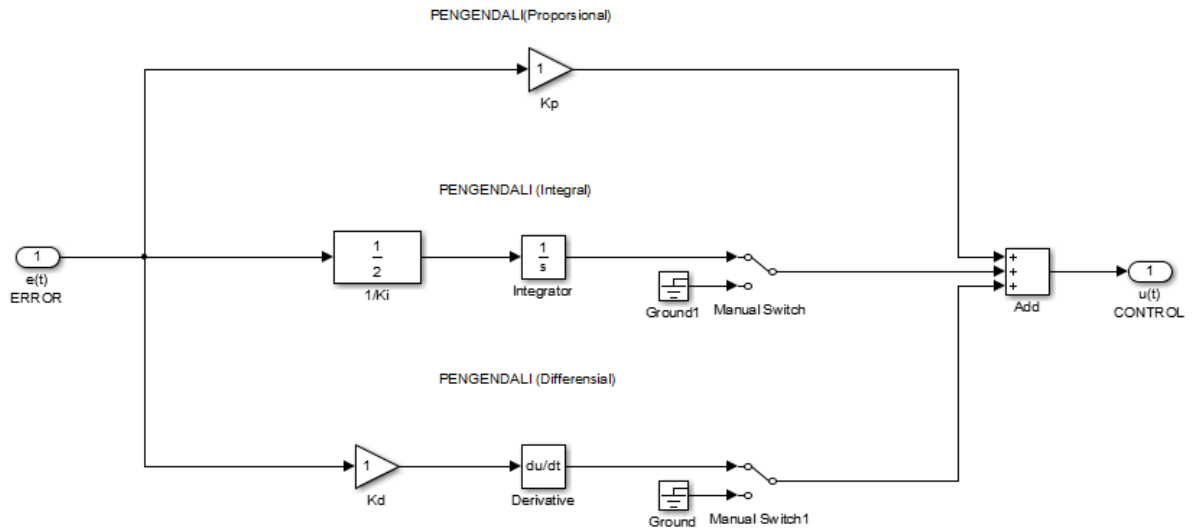
Selanjutnya adalah PID. Pendengali PID terdiri atas tiga buah pengendali; pengendali proportional (P), pengendali Differensial (D) dan pengendali Integral (I).

Pengendali proportional : memperbesar error sehingga akan lebih cepat dilakukan koreksi.

Pengendali differensial : meredam osilasi pada respon sistem.

Pengendali integral : mempercepat respon sistem mencapai keadaan stabil.

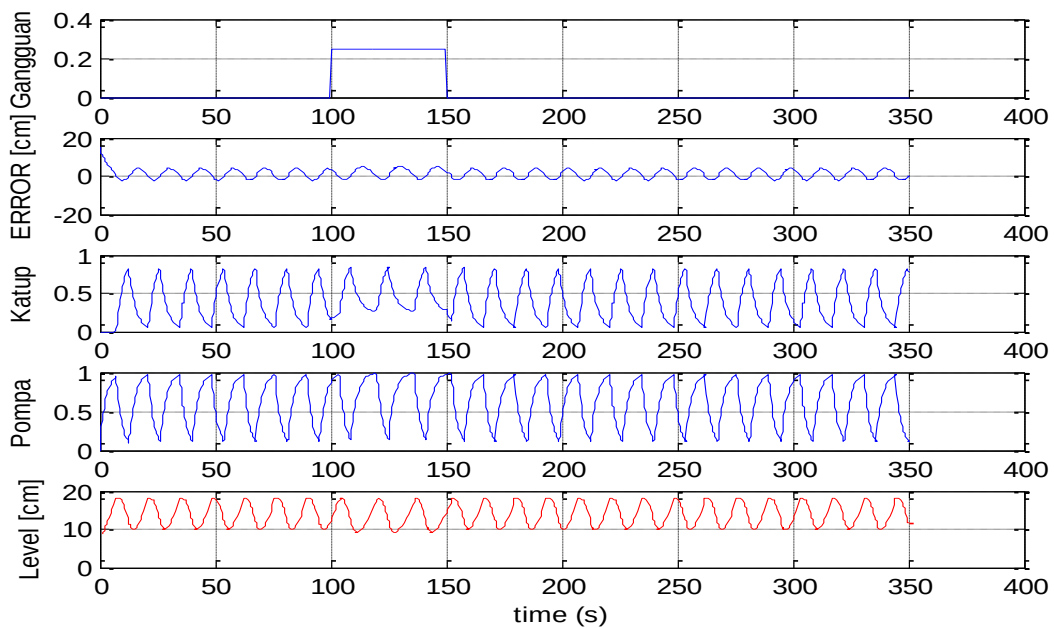
Blok diagram pengendali PID:



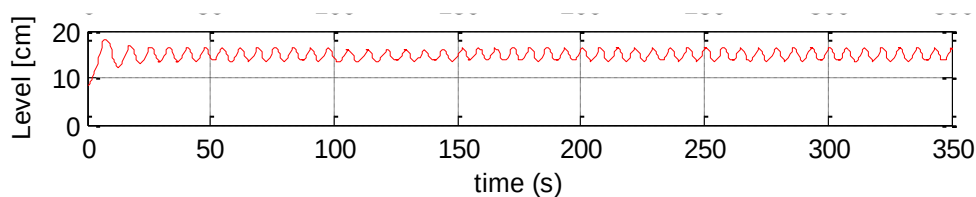
- 4 (empat) macam pengendali
- (1) Pengendali P, $K_P > 0$, $1/K_I = 0$ dan $K_D = 0$
 - (2) Pengendali PI, $K_D = 0$
 - (3) Pengendali PD, $1/K_I = 0$
 - (4) Pengendali PID, $K_P > 0$, $1/K_I > 0$, $K_D > 0$

Pelajari dan literatur, karakteristik masing-masing 4 pengendali kemudian terapkanlah

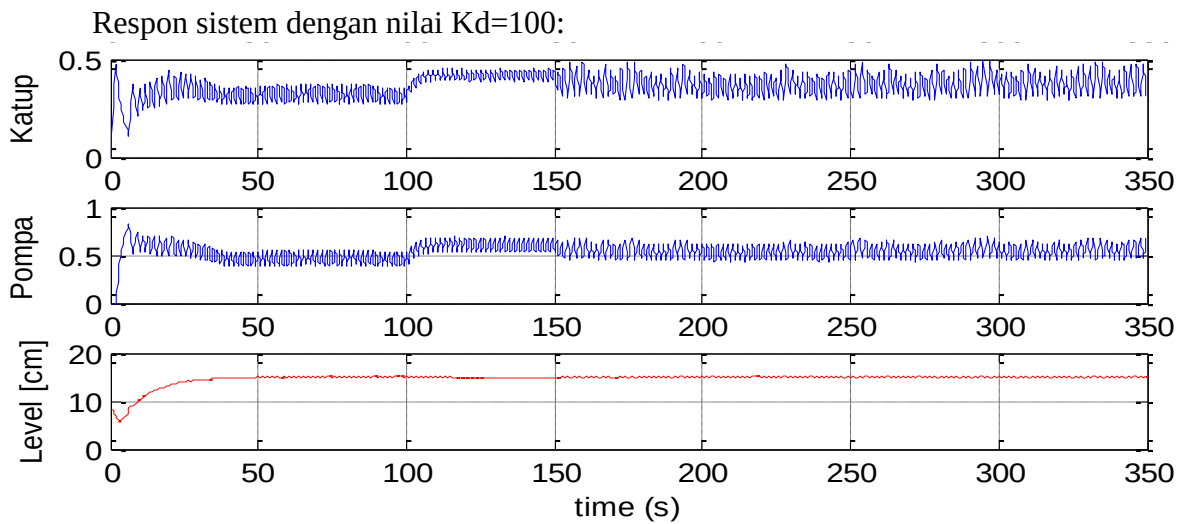
Keluaran pengendali PID $u(t)$ merupakan akumulasi dari pengendali P, I dan D.



Respon sistem menunjukkan bahwa level ketinggian fluida hanya berosilasi di sekitar setpoint. Untuk meredam osilasi maka dipasanglah kombinasi pengendali PD ($K_D=10$), sehingga respon sistem menjadi seperti grafik di bawah ini:

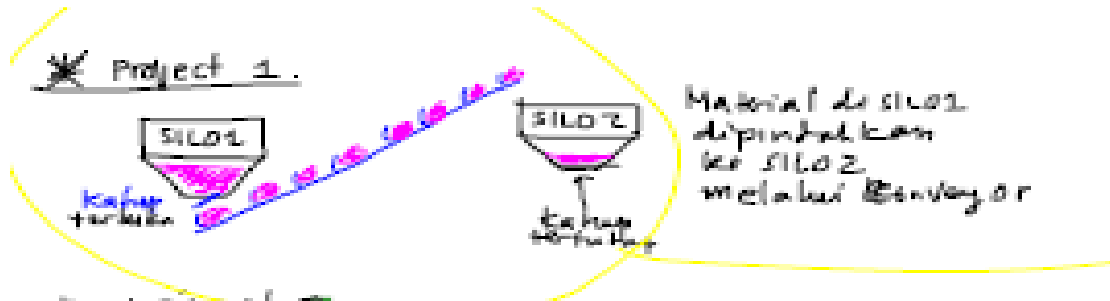


Terlihat bahwa osilasi yang dihasilkan cukup teredam. Untuk lebih meredam respon level ketinggian fluida, perlu dinaikkan nilai koefisien differensial (K_d) menjadi cukup besar untuk meredam osilasi yang terjadi (misal $K_d=50-100$).



BAB III

PROYEK SILO-TO-SILO



Sistem ini adalah suatu model proses untuk memindahkan benda berupa pasir antara dua tempat penampung atau silo. Sistem ini dilengkapi wadah-wadah seperti gelas atau cawan sebagai penahan tiap gundukan pasir.

Tempat menampung pasir pertama adalah Silo1. Pada bagian bawah Silo1 terdapat sebuah katup proporsional. Ujung katup akan langsung mengarah ke sebuah conveyor. Conveyor ini mempunyai wadah-wadah atau cawan-cawan yang menahan pasir yang akan jatuh dari katup sehingga tidak jatuh selama perjalanan. Conveyor akan bergerak membawa gundukan pasir menuju Silo2 untuk ditumpahkan pada Silo 2 tersebut.

Data-data perancangan model proses:

a. Silo1

-Katup (debit maks 1liter/20detik)

-Penampung

Diameter tabung 26 cm

Tinggi tabung 20.5 cm

Diameter atas kerucut terpancung terbalik 26 cm

Diameter bawah kerucut terpancung terbalik 4.5 cm

Tinggi kerucut terpancung terbalik 12 cm

b. Silo2

-Katup (debit maks 1liter/20detik)

-Penampung

Diameter tabung 26 cm

Tinggi tabung 10.5 cm

Diameter atas kerucut terpancung terbalik 26 cm

Diameter bawah kerucut terpancung terbalik 4.5 cm

Tinggi kerucut terpancung terbalik 12 cm

c. Conveyor

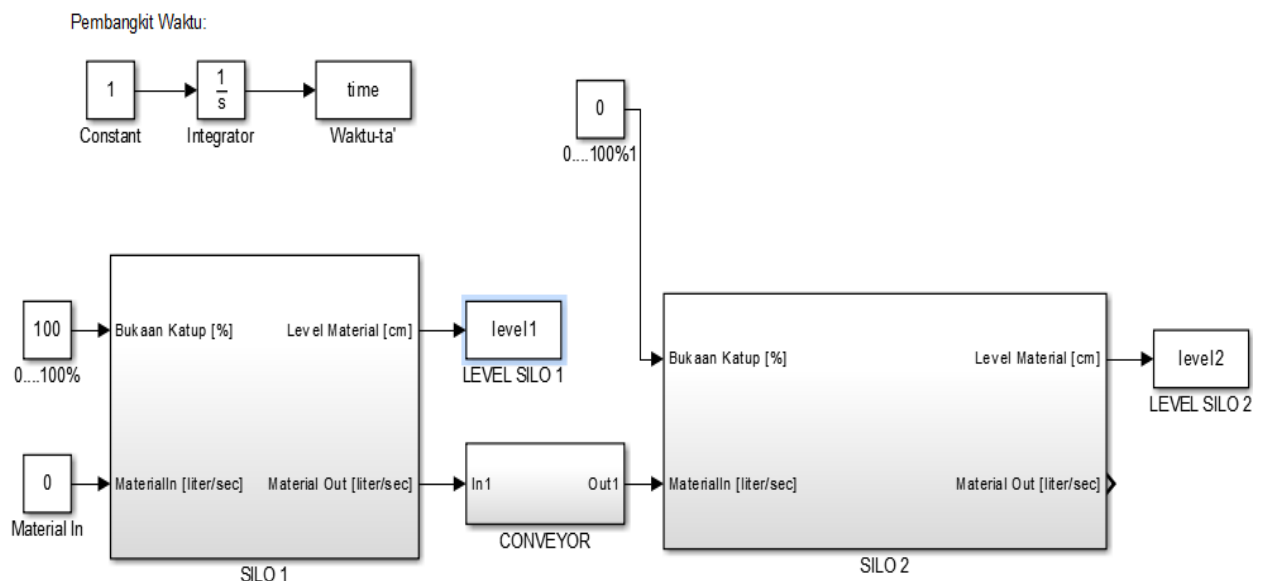
Kecepatan Motor 250rpm, (adjustable)

Roda gigi (kecepatan conveyor) 1:5 motor

Banyaknya cawan di atas conveyor: 10 cawan

Silo1 akan diisi material dari luar. Sebagai keluaran dari silo1, adalah material yang keluar dari katup. Untuk mengamati level material (dalam cm) pada Silo1, dapat pula ditambahkan sebuah variabel monitoring “Level1” pada keluaran lain dari Silo1.

Silo1 akan menjatuhkan material pada conveyor dan membentuk gundukan-gundukan pasir yang berjalan di atasnya. Pada conveyor dibuat dua parameter masukan, yaitu material_in dan kecepatan motor. Hanya akan dibuat keluaran tunggal untuk conveyor, yaitu material_out, yakni berupa gundukan-gundukan pasir yang dibawa dan dijatuhkan pada Silo2. Dipasang variabel monitoring dengan nama “_out” untuk memonitoring bentuk keluaran dari conveyor.



Pada Silo2 juga akan ditambahkan sebuah variabel monitoring dengan nama “Level2” untuk memantau ketinggian sesaat pasir pada Silo2. Katup pada Silo2 dibuat selalu tertutup, agar tidak ada material yang terbuang.

Silo1

Bukan cuma bukaan katup yang menjadi parameter masukan dari Silo1. Material_In merupakan parameter masukannya. Kita harus membagi nilai bukaan pada katup tersebut dengan 100. Untuk membagi, kita dapat menggunakan sebuah transfer function. Data menunjukkan bahwa katup memiliki debit 1 liter / 20 detik untuk bukaan penuh (setara dengan 0.05liter/detik).

Debit penambahan material pada Silo adalah material_in dikurangi debit bukaan katup. Dari parameter masukan material_in pada Silo langsung dapat dikurangkan dengan debit bukaan katup dalam satuan yang sama.

Untuk kerucut terpancung berlaku rumus $V = \frac{1}{3}\pi h(R^2 + Rr + r^2)$

Didapatkan bahwa volume maksimum dari Silo1 adalah 13.4389 liter. Mak batasan blok integrator yang digunakan diberi batasan saturasi dari 0 sampai 13.4389.

Variabel monitoring “Level1” dipasang bertujuan untuk memantau level ketinggian material pada Silo1. Perlu dilakukan konversi dari volume material dalam liter menjadi ketinggian material dalam cm.

Menggunakan “Interpreted MATLAB Fcn”, blok yang memiliki respon keluaran berdasarkan listing program vol to vol.

$$\text{Kerucut yang belum penuh, } t = \sqrt[3]{\frac{\pi h_0^3 r^2 + 3 V h_0^2}{\pi r^2}} - h_0.$$

Perlu ada sebuah blok pembatas yang me-0-kan material keluaran saat material di dalam Silo telah habis. Kita gunakan blok Interpreted MATLAB Fcn, yang keluarannya dikalikan dengan debit keluaran Silo1. Jika material belum habis, blok Interpreted MATLAB Fcn akan memberikan isyarat 1, sedangkan apabila material sudah habis, akan diberikan isyarat 0. Hasil perkalian inilah yang menjadi keluaran dari subsystem Silo1 untuk ditujukan ke conveyor.

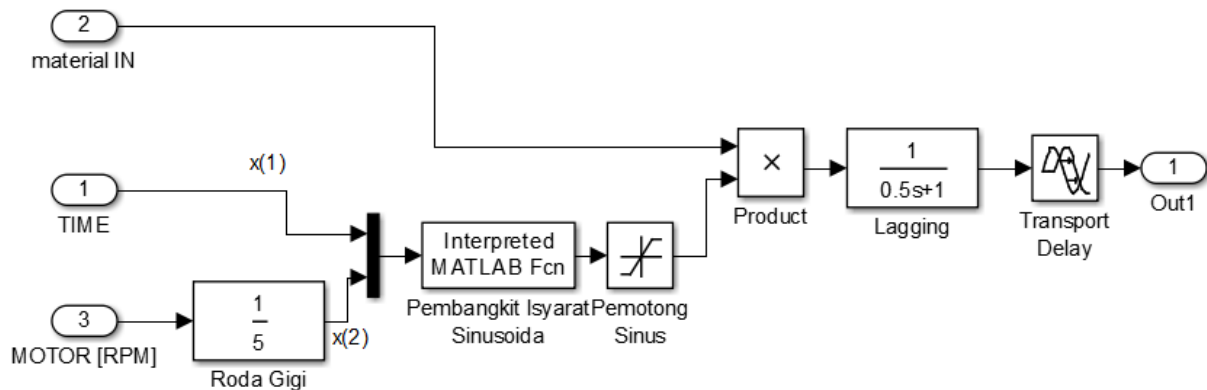
SILO2

Blok diagram Silo2 dapat menampung material yang lebih sedikit dari Silo1, dari segi bentuk blok diagram Silo2 sama dengan blok diagram pada Silo1. Perbedaan dari subsystem pada Silo2 hanya terletak pada batasan saturasi dari blok integrator-nya.

CONVEYOR

Tujuan sistem ini adalah untuk mengubah volume yang masuk yang mengalami perubahan bentuk menjadi gundukan-gundukan materia, proses memindahkan material mengakibatkan delay.

2 masukan Conveyor yaitu material_in dan kecepatan motor conveyor (rpm).



Bagian inilah menerangkan data-data teknis dari conveyor sehingga menjadi hal berguna untuk bagian lainnya. Di atas conveyor ada 10 buah cawan, maka dibutuhkan waktu selama 10 kali dari waktu menjatuhkan 1 cawan.

Sisa hasil bagi waktu terhadap waktu setiap cawan (n_time) akan menghasilkan nilai dari 0 sampai n_time (lamanya waktu setiap cawan dari mulai diisi). Didapatkan dari menggunakan fungsi “mod” dari blok Math Function. n_time juga menunjukkan lamanya waktu untuk 1 periode clock.

Kita dapat membuat keluaran akan bernilai 0 selama sisa hasil bagi lebih kecil dari setengah siklus clock, dan akan bernilai 1 pada setengah siklus berikutnya.

Dari persamaan ini, kita dapat membuat sebuah fungsi yang mendefinisikan volume yang tepat dari bentuk-bentuk gundukan yang dihasilkan, walau dalam satuan liter/detik. Namun, variabel n_time pada listing program berikut akan digantikan dengan variabel “waktu” yang ukurannya sengaja dibuat hanya 80% dari n_time , agar jarak antar satu gundukan dan gundukan berikutnya tetap terlihat.

List Program

#Conveyer Nol

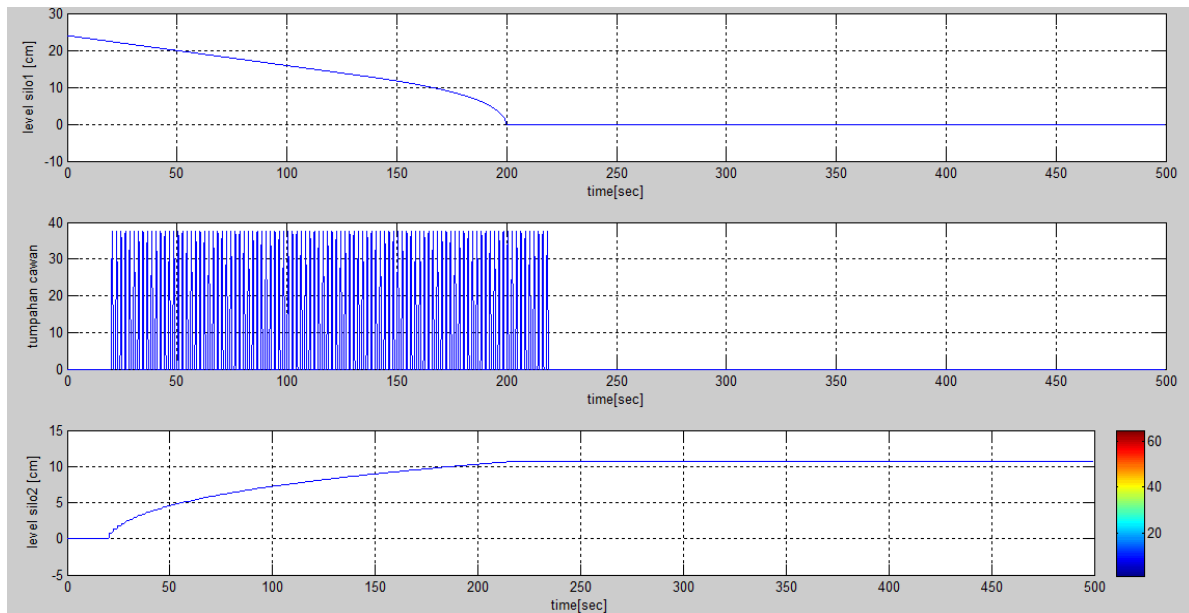
```
function y = conveyor(x)
t = x(1); % Dari pembangkit waktu
A = x(2); % Banyaknya material
yang keluar SILO1 [cm3/second]
% Menghitung T:
% Misalnya delay pada conveyor 20
seconds (tergantung kecepatan
motor)
% Misalnya sepanjang conveyor ada
10 cawan
T = 20/10; % [seconds] T =
2*pi/omega
omega = 2*pi/T; % [rad/sec]
Xsin = A*sin(omega*t); % Gelombang
sinusoide dengan amplitude A dan
frekuensi sudut omega
Xsin_potong = Xsin - (0.25*A); %
Memotong gelombang sinusoide pada
0.25*A
if Xsin_potong > 0
    y = Xsin_potong;
else
    y = 0;
end
```

#Detector _nol

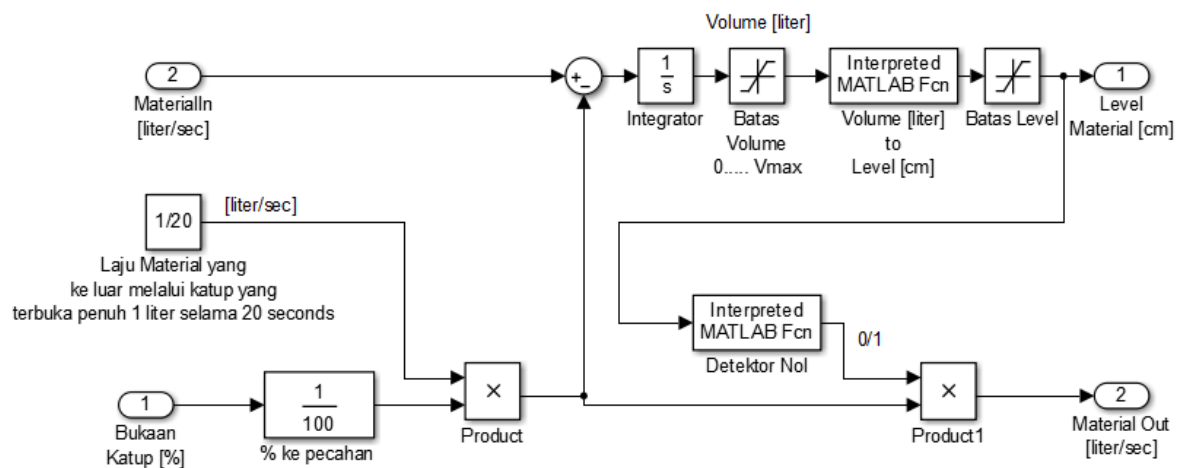
```
function y = detektor_nol(x)
level = x;
if level < 1
    y = 0;
else
    y = 1;
end
```

#Vol_to Vol

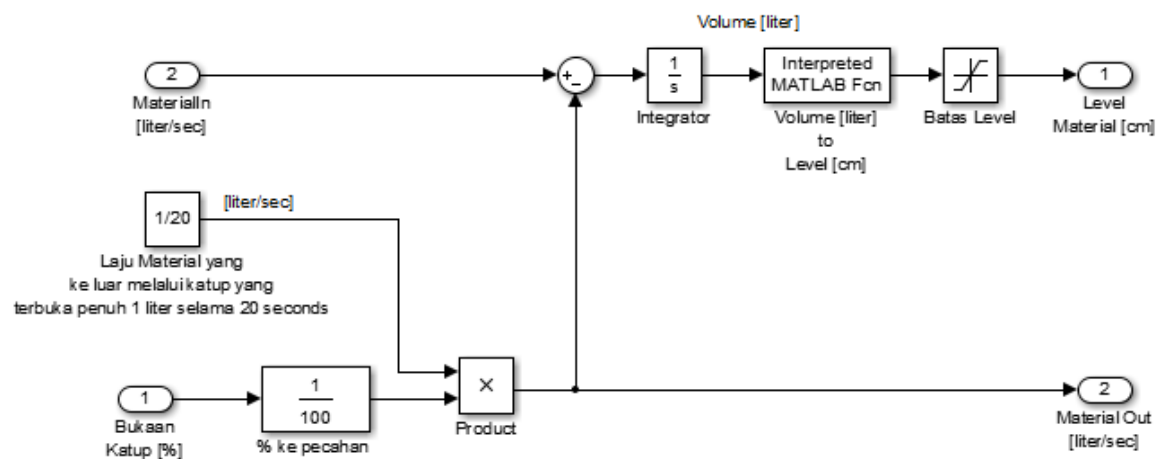
```
function y = vol_to_level(x)
Vin = x; % Volume input [cm3]
R = 14; % Jari-jari penampang silinder pada
SILO [cm]
r = 2.5; % Jari-jari katup [cm]
hk = 13; % Tinggi kerucut terpotong [cm]
hs1 = 10; % Tinggi silinder SILO1 [cm]
hs2 = 20; % Tinggi silinder SILO2 [cm]
x = (hk*r)/(R-r); % Tinggi kerucut kecil
% Volume kerucut Vk adalah Volume Kerucut
Besar
% ..... dikurangi Volume Kerucut Kecil:
Vk = (1/3)*(hk+x)*pi*R^2 - (1/3)*x*pi*r^2;
if Vin > Vk
    h = hk + ((Vin-Vk)/(pi*R^2));
else
    h =
(((hk+x)^2)*((3*Vin)+(x*pi*r^2))/(pi*R^2))^
(1/3)) - x;
end
y = h; % Level material dalam silo [cm]
```



SUBSYTEM SILO1



SUBSYTEM SILO2



BAB IV

PENUTUP

KESIMPULAN

Untuk memperkecil penumpukan yang terjadi pada saat pemrosesan kita menerapkan pengendali PID dengan mengubah-ubah nilai (value) yang sesuai sehingga kita mendapatkan penumpukan yang sedikit dan produk yang besar.

Mempelajari pemodelan sistem PID dalam Teknologi Kendali Proses sangat penting. Pemodelan ini dapat membantu kita melakukan analisis terhadap kesalahan-kesalahan sistem yang mungkin terjadi untuk meminimalisir kerugian dalam suatu proses.

LAPORAN PERKULIAHAN TEKNOLOGI KENDALI PROSES



CHAIDIR ANWAR_D411 12 012

TEKNIK KOMPUTER KENDALI ELEKTRONIKA (TKKE)

FAKULTAS TEKNIK JURUSAN ELEKTRO

UNHAS

BAB I

PENDAHULUAN

1.1. LATAR BELAKANG

Dalam rangka memenuhi tuntutan zaman sebagai mahasiswa teknik elektro, mahasiswa kini perlu memiliki kemampuan untuk dapat menerapkan ilmu yang dipelajarinya kedalam bidang kerjanya masing-masing. Dengan berbagai perkembangan teknologi, tentu perlu pembelajaran yang lebih banyak. Bagi seorang mahasiswa teknik elektro dengan konsentrasi Teknik Kendali, perlulah mendapat pembelajaran lebih mengenai cara mengendalikan dan merancang sebuah sistem fisik kedalam bentuk matematis yang dapat dianalisis, dalam hal kaitannya dengan suatu sistem fisik yang melakukan suatu proses, baik berupa proses fisika maupun kimia. Untuk itu penting bagi kita dalam memahami cara-cara yang diperlukan untuk membuat perancangan suatu sistem fisik kedalam model yang mudah untuk dianalisis –dalam hal ini adalah kestabilannya dan respon keluarannya–.

1.2. TUJUAN

Mampu mengubah suatu model sistem fisik kedalam model yang dapat dianalisis, baik matematis maupun dalam bagan kotak, sehingga dapat dianalisis kestabilan dan respon keluarannya. Dalam hal ini akan dilakukan dengan bantuan software *Simulink MATLAB R2013a (8.1.0.604)*.

BAB II

PROYEK GUNT

Gunt merupakan sebuah perusahaan di Jerman yang membuat miniatur dari berbagai proses. Salah satunya adalah *level control*, *hsi*. Berikut adalah gambaran dari sistem fisik tersebut:



Level control merupakan sebuah proses yang berusaha menjaga kestabilan level air pada tangki penampungannya. Miniatur ini memiliki sebuah pompa untuk memompa air dari bawah ke dalam tangki penampungannya, sebuah katup proporsional untuk membuang kelebihan air dari yang diinginkan, serta sebuah sensor tekanan yang dapat mengukur ketinggian air di dalam tangki penampungan. Sistem ini juga dilengkapi dengan sebuah pipa pembuangan ketika terjadi kelebihan tampungan pada tangki. Berikut ini adalah data-data teknis yang dibutuhkan dalam perancangan model proses:

1. Level-controlled tank

- capacity: 1,2L

2. Storage tank

- capacity: 3,7L

3. Pump

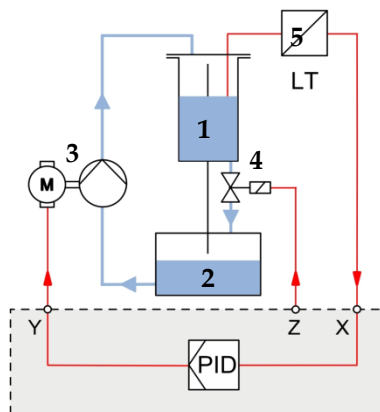
- power consumption: 18W

- max. flow rate: 8L/min

- max. head: 6m

4. Proportional valve: K_{vs} : 0,7m³/h

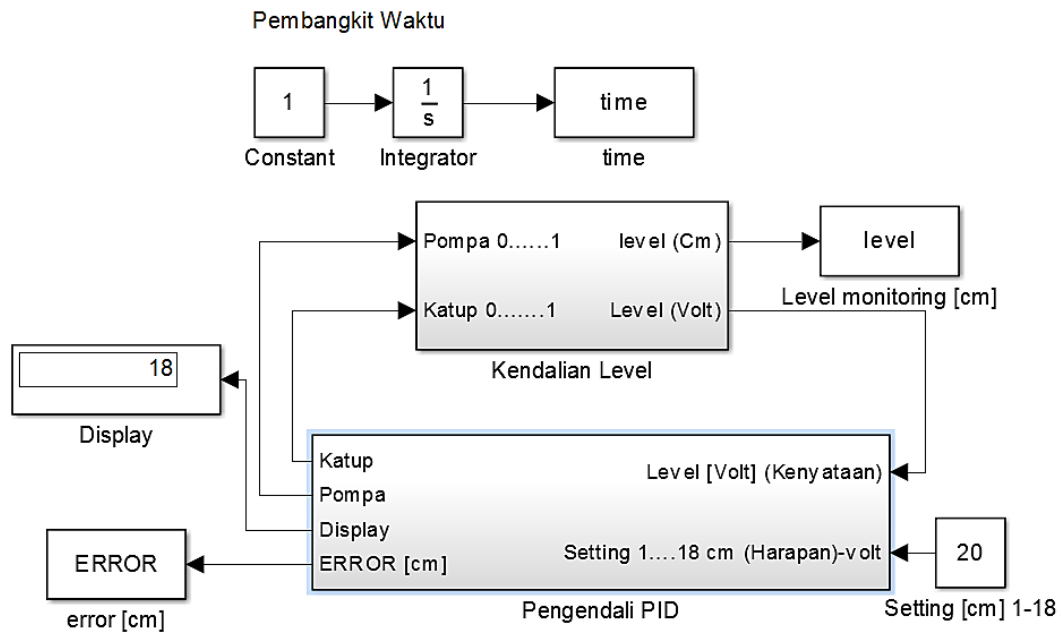
5. Pressure sensor: 0...30mbar (0...300mm)



Bagan skematik proses dari miniatur diperlihatkan pada gambar disamping. Sebagai informasi tambahan yang tidak diketahui, diasumsikan bahwa tinggi tangki adalah 20 cm dengan tinggi pipa pembuangan *overflow* adalah 18 cm.

Dari skematik proses diatas, dapat dilihat bahwa ada dua blok utama, yakni pengendali dan kendalian. Pengendali akan mendapat isyarat *setpoint*, yang berupa nilai pengaturan ketinggian air yang diinginkan. Pada pengendali, dibutuhkan dua buah masukan sebagai parameter acuannya, yaitu *setpoint* dan level ketinggian air yang telah dicapai, yang diambil dari pembacaan sensor yang sudah dalam satuan volt. Sebagai keluaran dari pengendali adalah isyarat kendalian untuk *subsystem* katup dan pompa. Keluaran lain dari pengendali disini sebagai tambahan adalah *display* yang akan menunjukkan hasil pembacaan dari *sensor* (lebih tepatnya dikatakan sebagai *transducer*) dalam satuan *centimeter*. Keluaran lainnya adalah galat (*error*) yang disimpan pada sebuah variabel "*ERROR*" (*disebut to workspace*) yang akan menyimpan kondisi galat setiap waktu selama selang waktu simulasi sehingga respon dari sistem dapat teramati. Untuk kemudahan, selanjutnya untuk dilakukan *monitoring* akan digunakan bentuk variabel seperti ini.

Sedangkan untuk kendalian, akan menerima masukan berupa isyarat kendalian dari pengendali, yaitu isyarat kendalian pompa katup. Satu-satunya keluaran dari *subsystem* kendalian adalah hasil dari pembacaan sensor ketinggian dalam satuan *volt*. Namun sebagai tambahan, dipasang sebuah variabel “*level*” –yang dimaksud adalah *to workspace*, bertipe *array*– untuk level ketinggian air pada tangki. Untuk kemudahan, dibuatlah pembangkit waktu dengan persamaan $\int 1 dt = t$ untuk membantu *memonitoring* respon sistem terhadap waktu (dengan nama variabel “*time*”). Berikut ini adalah model umum yang telah kita capai hingga saat ini, dengan Kendalian Level dan Pengendali PID berupa *subsystem*.



1. KENDALIAN LEVEL

Seperti yang telah dijelaskan sebelumnya, Kendalian atau *Plant* memiliki dua masukan, yaitu isyarat kendalian untuk pompa serta katup, serta dua buah keluaran, yakni umpan balik atau *feedback* dari sensor serta untuk *monitoring* ketinggian air pada tangki.

1.1.POMPA

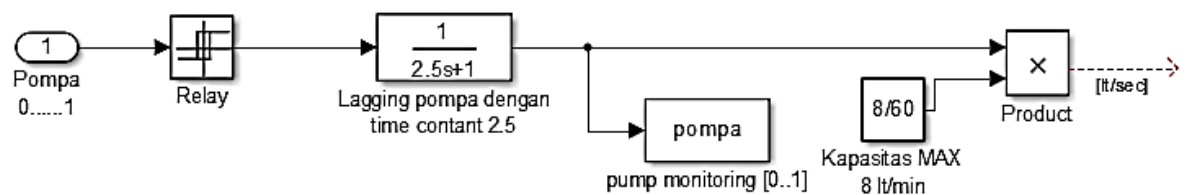
Pompa memiliki isyarat kendalian yang diasumsikan dari nol sampai satu [0..1], dimana nol adalah kondisi saat pompa dimatikan, sedangkan satu adalah saat pompa dinyalakan dengan kecepatan maksimum. Diketahui bahwa pompa yang digunakan memiliki kecepatan aliran maksimum 8 liter/menit. Untuk itu dibuat sebuah blok *constant* yang digunakan sebagai pembanding terhadap sinyal kendalian. Sebagai catatan, pompa juga dapat diberi isyarat kendalian kurang dari 0 karna sifat *hysteresis* dari pompa yang harus diberi isyarat kendalian kurang dari 0 untuk memmatikannya.

Karena pompa tentunya diputar oleh motor yang dikontrol oleh sebuah *relay*, maka pada pompa akan berlaku suatu model *hysteresis*, yakni kondisi dimana pompa tidak memiliki titik yang sama untuk on atau off dari keadaan sebelumnya. Untuk itu, dari input pompa dipasang sebuah blok *relay* yang memberikan efek *hysteresis* tersebut. Sebagai parameter pada blok relay, diberikan titik pergantian *on* pada 0.2, dan titik pergantian *off* pada -0.2, dengan keluaran adalah 1 saat *on* dan 0 saat *off*.

Setelah *on*, motor pompa tidak akan serta merta berputar dengan kecepatan maksimum sehingga aliran pompa maksimum tidak mungkin akan langsung tercapai. Untuk motor orde 1, motor pompa akan *lagging* terhadap respon masukannya dengan

time constant (τ) tertentu. Untuk itu, digunakan sebuah blok *transfer function* untuk membuat *lagging* dengan *time constant* τ dengan persamaan $\frac{1}{\tau s + 1}$.

Keluaran dari blok *transfer function* ini dapat dipasang variabel *monitoring* “pompa” untuk mengetahui respon dari kecepatan motor pompa. Hasil dari *lagging* berupa kecepatan motor terhadap putaran maksimumnya [0..1], sehingga berdasarkan putaran motor maksimum, kita dapat membandingkan dengan kecepatan aliran maksimum – blok *constant* yang dibahas sebelumnya – untuk mendapatkan kecepatan aliran air sesaat yang dipompa ke dalam tangki. Untuk membandingkan, kita dapat menggunakan sebuah blok *product* dengan keluaran berupa kecepatan aliran yang dipompa ke dalam tangki dalam satuan *liter/detik*. Jika diasumsikan $\tau=2.5$, maka secara keseluruhan blok diagram pompa dapat digambarkan sebagai berikut:



1.2.KATUP

Katup yang digunakan adalah katup proporsional, artinya tidak hanya tertutup atau terbuka secara penuh, tapi memiliki *range*, seperti setengah tertutup, atau seperempat terbuka. Untuk itu, maka isyarat kendalian dapat berupa nilai-nilai dari nol sampai satu, dengan asumsi bahwa 0 adalah saat pompa tertutup penuh dan 1 saat pompa terbuka penuh. Sebagai catatan, katup tidak akan diberikan keadaan isyarat kendalian kurang dari 0 atau lebih dari 1.

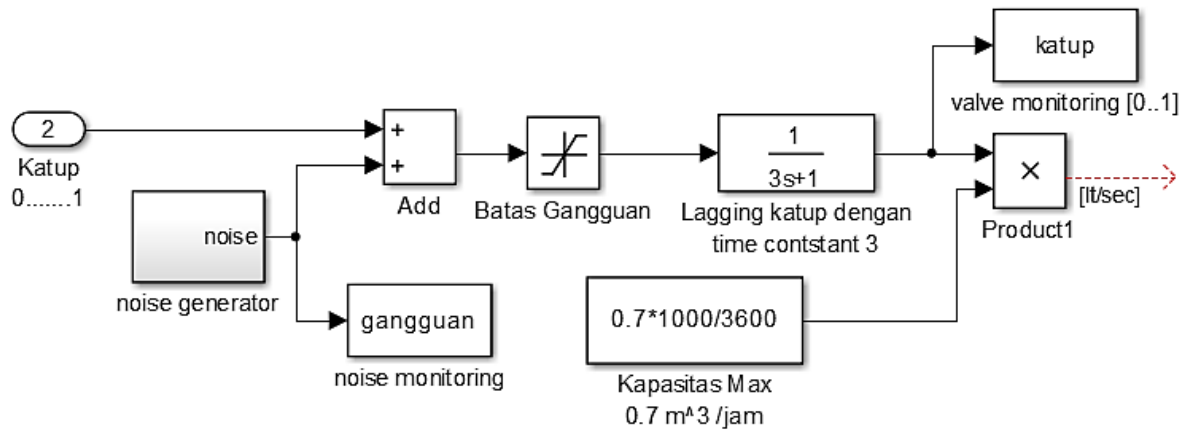
Katup akan beroperasi dengan rentang isyarat kendalian dari 0 sampai 1f. Oleh sebab itu, isyarat kendalian diberi blok *saturation*, sehingga memberi batasan apabila ada gangguan yang muncul dan menyebabkan isyarat kendalian yang diterimanya bernilai tidak pada rentang tersebut.

Seperti pada motor, katup juga tidak dapat langsung menutup dan membuka sekat berdasarkan isyarat kendaliannya, melainkan akan mengalami *lagging*. Oleh sebab itu, setelah blok *saturation* kembali dipasangkan sebuah blok *transfer function* yang memberikan sifat *lagging* tersebut pada katup. Setelah blok ini akan diberikan sebuah variabel *monitoring* dengan nama “katup” untuk memantau status dari katup.

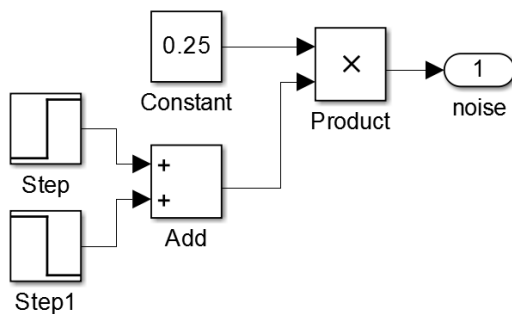
Karena katup dapat membuang fluida dengan debit maksimum 0.7 m³/hours, dan isyarat kendalian yang telah keluar dari blok *lagging* akan menghasilkan isyarat dari 0 sampai 1, maka kita dapat kembali membandingkan besarnya aliran yang keluar dari katup sesaat terhadap debit maksimumnya dengan kembali menggunakan blok *product*. Tentu saja dengan mengkonversi satuan dari debit maksimum dalam *liter/sec* terlebih dahulu. Keluarannya berupa debit aliran air yang dibuang dari tangki melalui katup dalam satuan *liter/second*.

Sebagai tambahan, sebelum melewati blok *saturation*, isyarat kendalian dapat diberikan gangguan untuk melihat respon sistem apabila terdapat gangguan. Akan ada

sebuah *subsystem* yang dibuat dapat menghasilkan gangguan pada waktu tertentu sesuai keinginan, yang kemudian akan ditambahkan dengan blok *add* sehingga menghasilkan suatu isyarat kendalian yang terganggu. *Subsystem* penghasil gangguan dapat dipantau dengan memasang variabel *monitoring* “*gangguan*”. Sehingga apabila keseluruhan sistem digambarkan, dengan *time constant* diasumsikan bernilai 3 detik, akan membentuk susunan blok diagram sebagai berikut:



Untuk *noise generator*, yakni *subsystem* penghasil gangguan, misalnya dapat dibuat dengan blok diagram di bawah ini. Tujuannya adalah menghasilkan gangguan berbentuk pulsa yang dapat diaktifkan selama waktu tertentu, misalnya dari 100 sampai 150 detik dengan amplitudo 0.25.

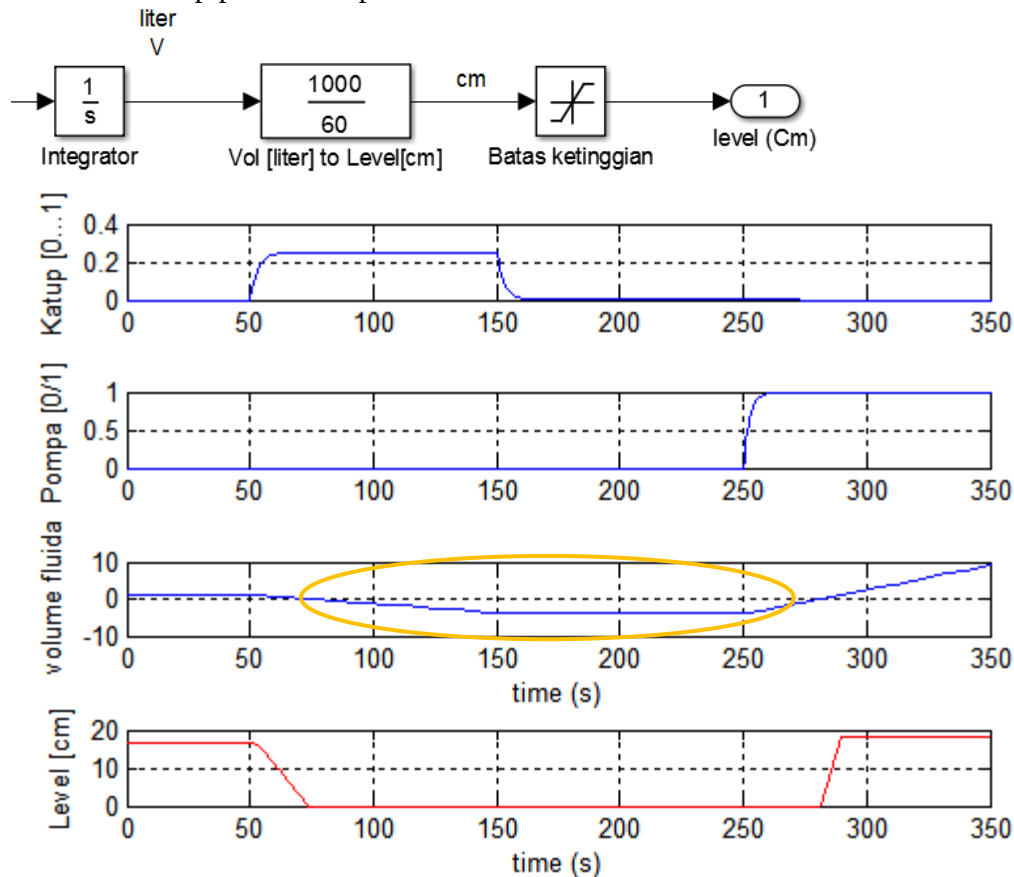


1.3.LEVEL CONTROLLED TANK

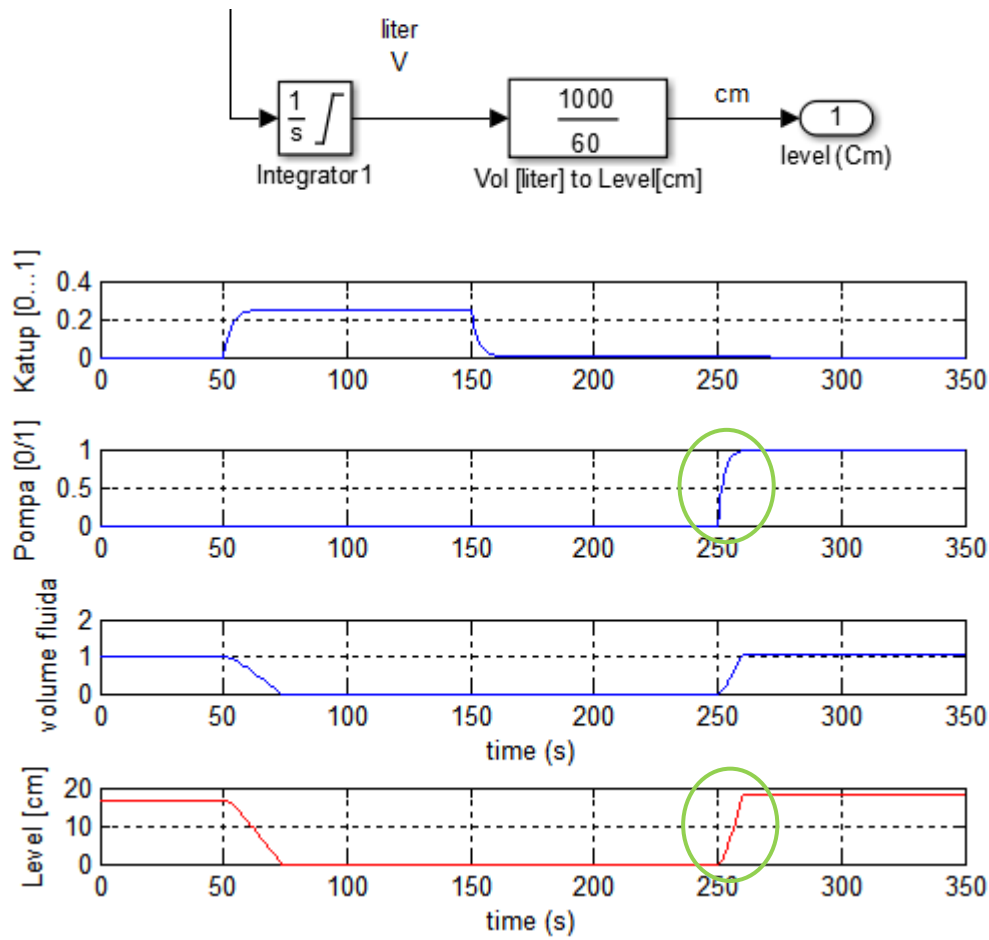
Level controlled tank merupakan tangki penampungan yang akan diatur ketinggiannya. Tangki ini akan mendapat tambahan fluida dari pompa, dan kelebihan fluida dapat dibuang melalui katup. Karena pompa bersifat menaikkan level fluida pada tangki, dan katup bersifat mengurangi aliran level fluida, maka aliran sesaat pada tangki adalah kecepatan aliran pada pompa dikurangi kecepatan aliran pada katup. Untuk itu dipasanglah suatu blok yang mengurangi aliran pompa terhadap aliran katup tersebut, yakni blok *sum*.

Hasil dari penjumlahan ini memiliki satuan *liter/detik* –turunan volume terhadap waktu–. Sehingga untuk mendapatkan volume sesaat pada tangki maka dilakukan pengintegralan sehingga menghasilkan satuan volume. Dengan asumsi luas penampang tangki adalah 60 cm^2 , dengan tinggi pipa pembuangan *overflow* 18 cm – 18.1 cm untuk toleransi 0.1cm– maka batas dari volume ini adalah dari 0 sampai 1.086 liter. Volume merupakan integrasi dari kecepatan aliran fluida setiap waktu. Untuk itu perlulah menggunakan suatu blok *integrator* yang dilengkapi *saturation* agar volume yang ada di dalam tangki tidak bernilai di luar rentang ini. Mengapa

tidak menggunakan blok *saturation* saja setelah diubah menjadi satuan cm agar lebih mudah perhitungannya? Sebab, apabila dilakukan setelah blok *integrator*, maka pada kondisi katup dibuka maksimum tanpa tambahan fluida dari pompa, hasil *integrator* tetap akan bernilai negatif walaupun pada *monitoring* nilai ini tetap terbaca sebagai level 0cm akibat pemasangan blok *saturation* setelah *integrator*. Namun, begitu pompa dinyalakan, level air tidak akan langsung naik, karena pompa perlu kembali menolak volume fluida yang bernilai negatif tersebut. Itulah mengapa pembatasan volume fluida dilakukan langsung pada blok *integrator*, sehingga apapun yang terjadi, volume fluida akan bernilai kurang dari nol. Mari membandingkan grafik respon sistem terhadap perbedaan peletakan *saturation*.



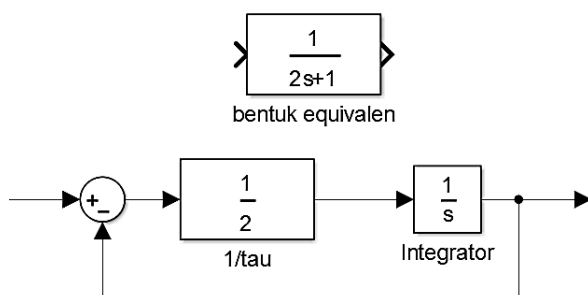
Pada blok diagram yang menggunakan *saturation* setelah blok *integrator*, volume akan tetap bernilai negatif, dan level air akan mulai bertambah saat volume ini telah melewati titik nol. Berbeda dengan blok *integrator* yang telah dilengkapi dengan batas (saturation) untuk volumenya. Saat pompa dinyalakan, maka saat itu juga level ketinggian fluida di dalam tangki akan naik. Secara fisik, model yang kedua inilah yang masuk akal. Blok *integrator with saturation* akan diberikan batas bawah 0 dan batas atas 1.086. Pada blok ini juga dapat diisi *initial condition*, yakni kondisi awal dari tangki dalam satuan liter (misalnya 1).



Karena keluaran yang akan dimonitoring dari *subsystem* ini adalah level ketinggian tangki dalam satuan cm, perlu dilakukan perubahan dari volume menjadi ketinggian. Ketinggian dalam cm adalah volume dalam cm^3 dibagi luas penampangnya dalam cm^2 . Untuk itu digunakan sebuah blok *transfer function* yang mengkonversi nilai ini sebagai keluaran. Sebuah *subsystem sensor* akan dipasang pula pada keluaran ini untuk mengawasi level ketinggian fluida dalam cm sehingga menjadi level ketinggian dalam *volt*.

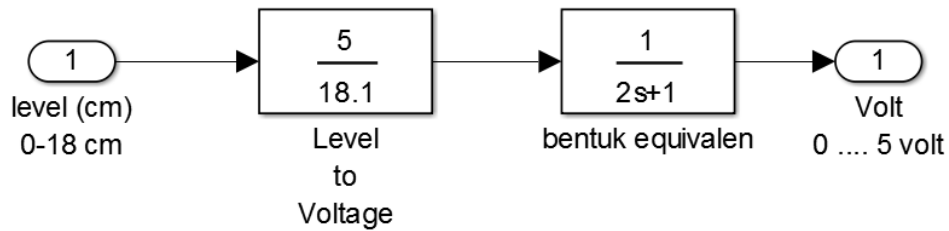
1.4.SENSOR

sensor bertujuan untuk membaca ketinggian air dalam cm, dan memberikan keluaran dalam satuan *volt*. Untuk ketinggian maksimum adalah 18.1 cm, dan akan dikonversi dalam satuan *volt* dari 0 sampai 5, maka dari satuan cm dapat diubah ke *volt* dengan mengalikannya dengan sebuah blok *transfer function* $\frac{5}{18.1}$.

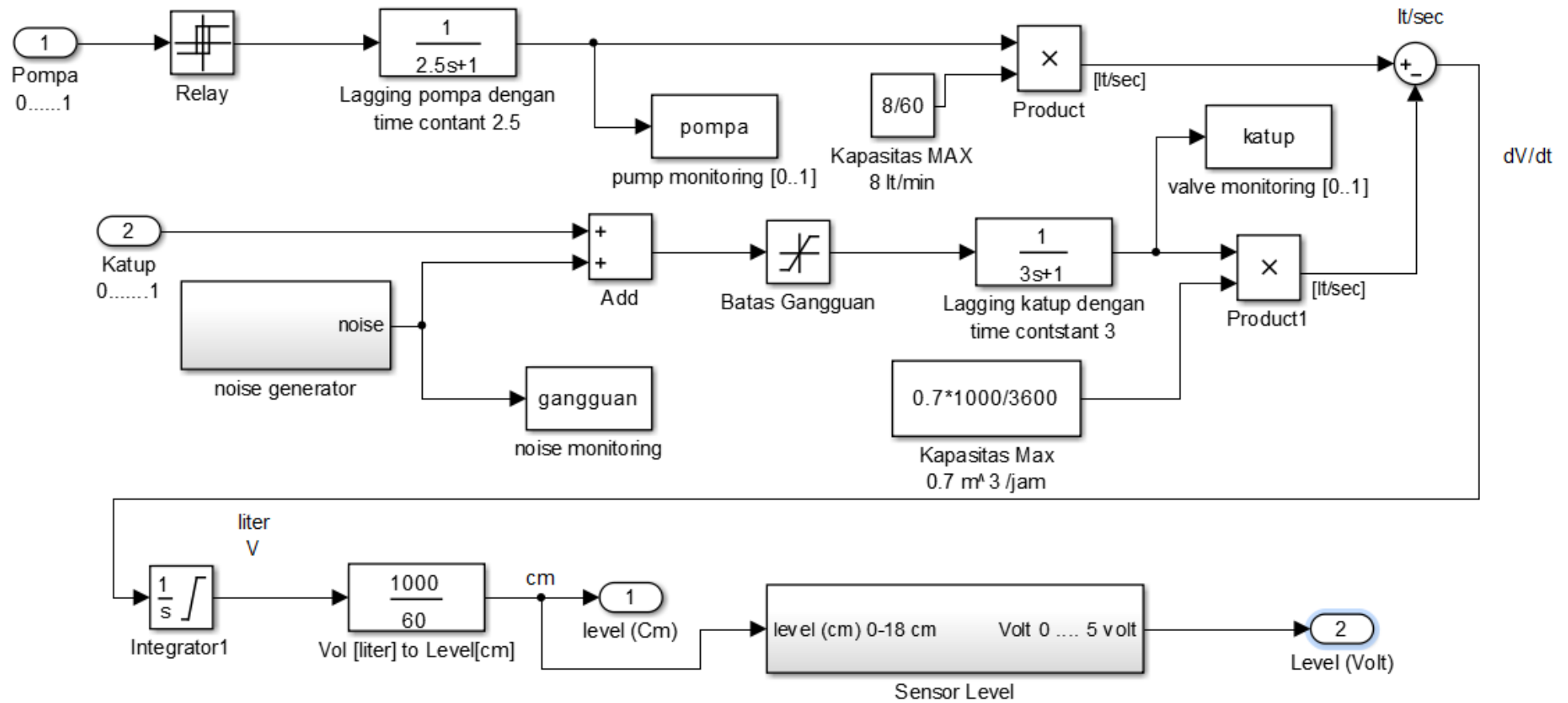


Namun, seperti sensor pada umumnya, tentu pembacaan akan mengalami *lagging*. Misalnya saja dengan *time constant* 2, maka blok *lagging* dapat berbentuk seperti sebelumnya (gambar atas), atau dengan penjabaran (gambar bawah). Kelebihan dari gambar di bawah, kita dapat menentukan nilai awal pada blok *integrator*. Yang perlu diketahui adalah penambahan nilai awal pada

sensor akan merusak hasil pembacaan sensor, dan dapat menyebabkan masalah kemudian. Misalns saja tidak akan pernah di dapatkan ketinggian fluida yang tepat jika dilihat dari pembacaan sensornya, sehingga output akan terus berosilasi atau ketinggian air tidak akan sesuai dengan *setpoint* yang diberikan. Untuk itu, pemberian nilai awal pada sensor tidak perlulah diberikan, karena sensor akan membaca nilai awal dari ketinggian fluida dengan sendirinya, walaupun tanpa memberikan nilai awal yang sama pada sensor. Secara lengkap, blok diagram *subsystem* sensor ditunjukkan pada gambar berikut:



Secara lengkap, blok diagram *subsystem* kendalian level diberikan pada gambar di bawah ini:



2. PENGENDALI PID

Pengendali PID adalah *subsystem* yang memberikan isyarat kendalian pada kendalian agar level ketinggian fluida pada tangki tetap stabil sesuai dengan nilai *setpoint* yang diberikan. Masukan dari pengendali ada dua, yakni *setpoint* atau harapan dan hasil pembacaan sensor dalam *volt*, atau kenyataan. Sedangkan galat atau *error* adalah selisih antara harapan dan kenyataan.

Yang pertama adalah *display* yang menunjukkan hasil pembacaan sensor dalam satuan *centimeter*. Karena rentang keluaran sensor adalah 0 sampai 5 volt untuk pembacaan 0 sampai 18.1 cm, perlu diberikan blok pembalik dari volt ke cm, yakni *transfer function* dengan nilai $\frac{18.1}{5}$, sehingga dari blok ini akan langsung didapatkan level ketinggian fluida dalam satuan cm.

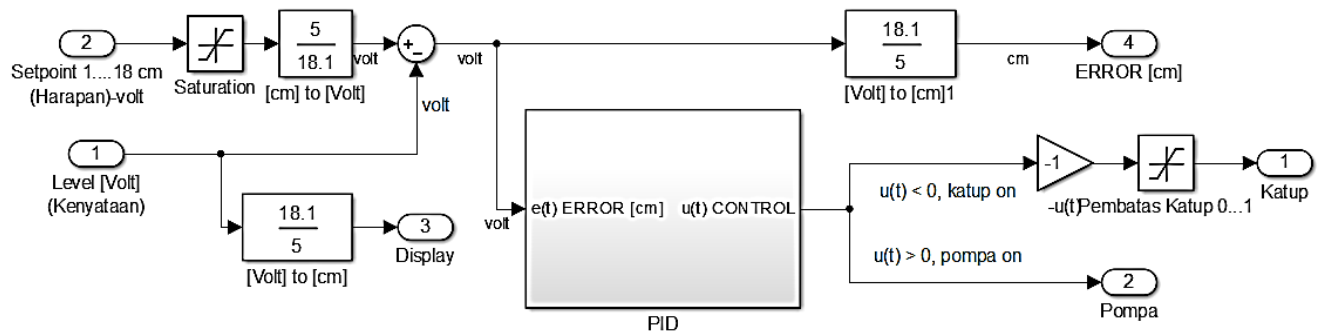
Karena hasil pembacaan sensor adalah dalam *volt*, maka untuk mencari galat, perlu dilakukan konversi *setpoint* terlebih menjadi satuan yang sama, *volt*. Namun, sebelum dilakukan konversi, perlu dilakukan pembatasan pada *setpoint*, agar nantinya *setpoint* berlebih, misalnya lebih besar dari tinggi pipa pembuangan *overflow*, maka akan mengakibatkan pompa terus *on* karena tinggi fluida pada tangki tidak akan mungkin melebihi tinggi pipa pembuangan *overflow* tersebut. Untuk itu, dari masukan *setpoint* dipasang sebuah blok *saturation* yang nilainya dibuat dari 0 sampai 18.1 cm. Barulah setelah itu sebuah blok *transfer function cm-to-volt* ($\frac{5}{18.1}$) diletakkan.

Sekarang, keduanya –harapan dan kenyataan– telah dalam satuan yang sama. Barulah *error(volt)* bisa didapatkan dengan mengurangkan antara harapan dan kenyataan. Jika hasil bernilai negatif, artinya ketinggian fluida melebihi harapan sehingga katup perlu dibuka, dan apabila bernilai positif, artinya ketinggian fluida kurang dari harapan, sehingga pompa yang perlu di jalankan. Untuk memantau *error* dalam *centimeter* sebagai salah satu keluaran dari pengendali, kembali setelah *error* dalam *volt* didapatkan, diletakkan sebuah blok *transfer function volt-to-cm* ($\frac{18.1}{5}$) yang keluarannya sudah merupakan *error* dalam satuan *centimeter*.

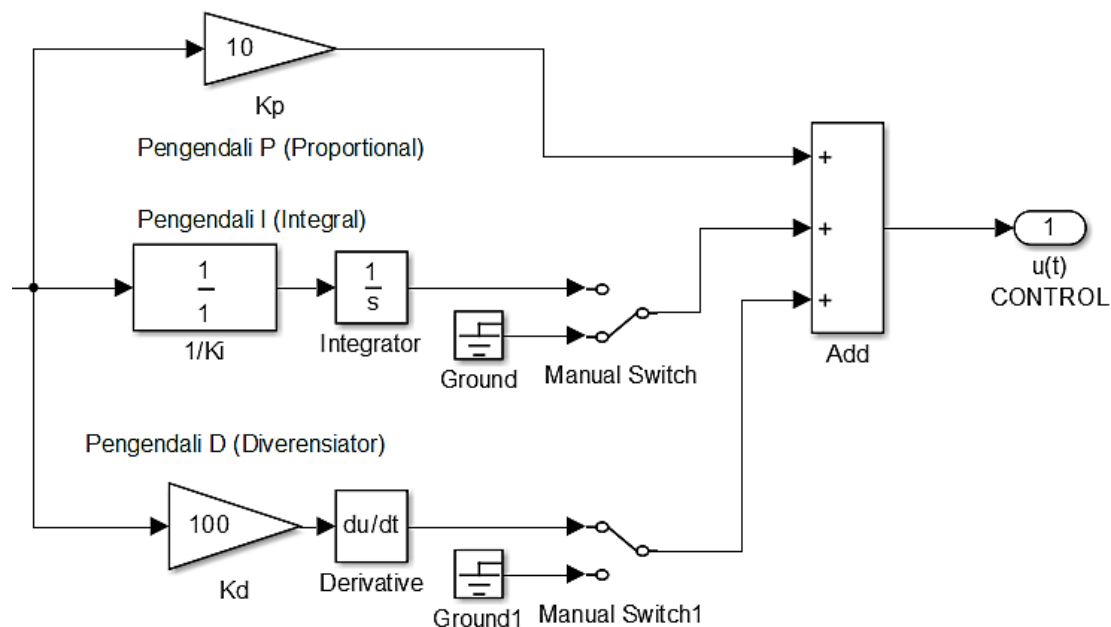
Pengendalian pompa dan katup dilakukan dengan menambahkan sebuah *subsystem PID* yang berisi pengendali PID berdasarkan *error* yang diterimanya. Oleh sebab itu, keluaran dari pengendali apabila bernilai positif akan dibuat untuk menggerakkan pompa dan apabila bernilai negatif akan dibuat untuk menggerakkan katup. Untuk itu dari keluaran *subsystem PID* langsung disambungkan ke keluaran yang menuju pompa, sehingga keluaran PID tersebut akan langsung mengendalikan pompa. Sedangkan untuk katup saat keluaran PID bernilai negatif, perlu dilakukan pembalikan dan pembatasan terlebih dahulu. Dilakukan pembalikan karena yang mengaktifkan katup adalah keluaran negatif dari PID, sedangkan katup akan menerima isyarat kendalian dari 0 sampai 1 (positif). Untuk itulah keluaran PID dilakukan pembalikan terlebih dahulu. Sedangkan dilakukan pembatasan dari 0 sampai 1 dilakukan karena agar isyarat kendalian tidak bernilai negatif ataupun bernilai diatas 1. Namun, untuk membuat katup tidak terlalu rentan terhadap gangguan, batasan atas yang dimasukkan bukanlah 1, melainkan 5-10. Batasan atas juga tidak boleh terlalu tinggi, sebab efek *lagging* pada katup dapat menyimpan isyarat yang terlalu besar sehingga dapat menyebabkan sistem malah menjadi lebih tidak stabil. Pembatasan diatas 1 ini tidaklah menjadi masalah, sebab pada kendalian sendiri sudah diberikan batasan sehingga isyarat yang diterimanya tidak melewati rentang 0..1.

Lalu, mengapa pompa tidak diberikan pembatasan yang sama? Pompa memiliki sifat *hysteresis*, sehingga butuh nilai isyarat kendalian negatif yang cukup untuk mematikan pompa. Jika diberikan batasan yang sama pada pompa, isyarat kendalian pompa tidak akan pernah bernilai negatif, sehingga pompa akan terus-menerus *on*. Inilah sebab mengapa pompa tidak diberikan batasan isyarat kendalian.

Berikut ini adalah blok diagram dari *subsystem* Pengendali *PID* secara lengkap:

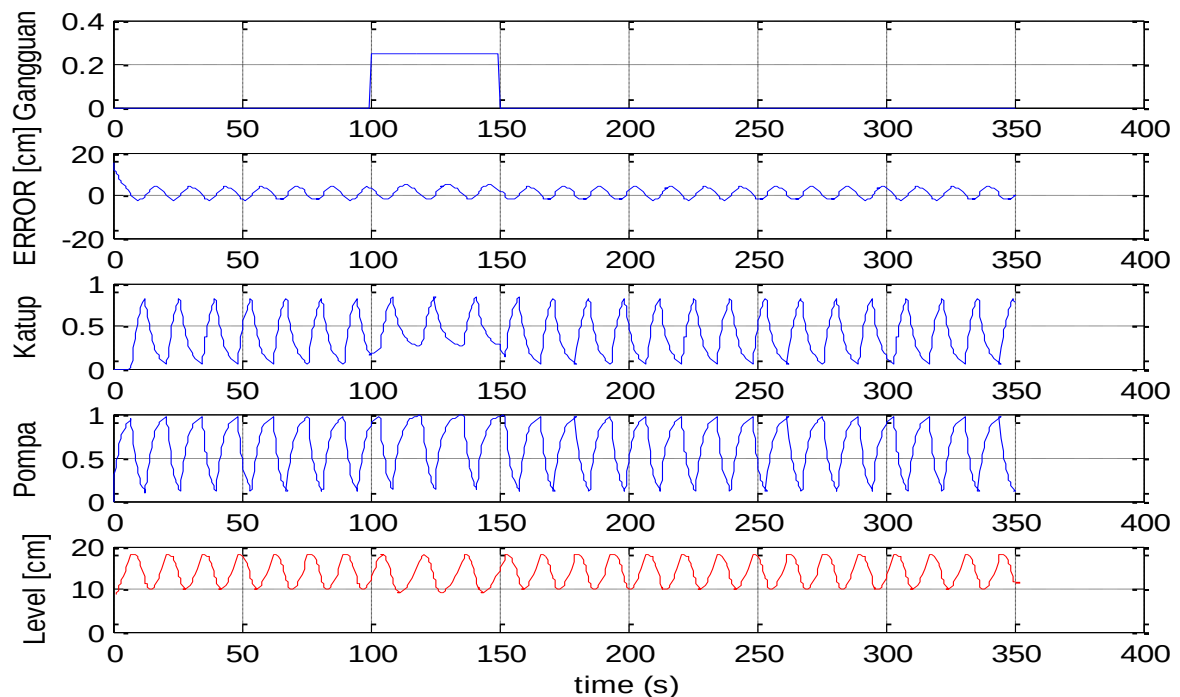


Selanjutnya adalah *subsystem* *PID*. Pendengali *PID* terdiri atas tiga buah pengendali; pengendali *proportional* (*P*), pengendali *Differensial* (*D*) dan pengendali *Integral* (*I*). Pengendali *proportional* bersifat memperbesar error sehingga akan lebih cepat dilakukan koreksi. Pengendali *differensial* bersifat meredam osilasi pada respon sistem. Sedangkan pengendali *integral* bersifat mempercepat respon sistem mencapai keadaan stabil, namun bersifat memberikan efek osilasi pada respon sistem tersebut. Berikut ini adalah blok diagram pengendali *PID* secara umum:

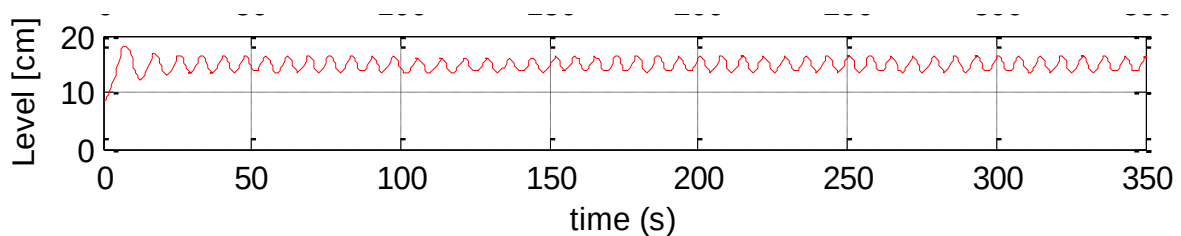


Terlihat bahwa keluaran pengendali *PID* $u(t)$ merupakan akumulasi dari pengendali *P*, *I* dan *D*. Grakfik berikut akan menunjukkan bagaimana respon sistem (ketinggian fluida) dengan

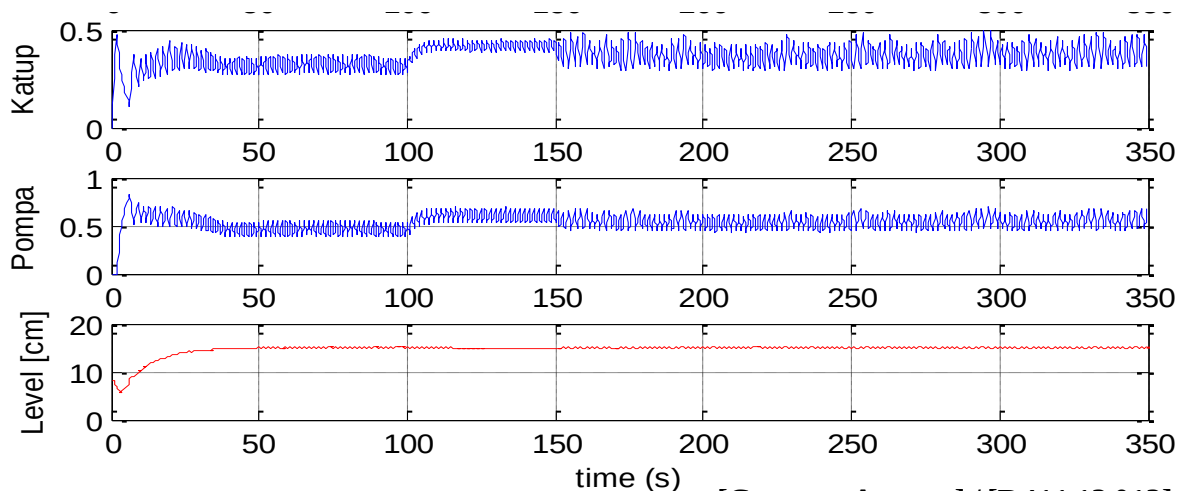
hanya menggunakan pengendali proporsional dengan $K_p=10$ terhadap *setpoint* 15 cm (dengan volume awal 0.5):



Dari respon sistem terlihat bahwa level ketinggian fluida hanya berosilasi di sekitar *setpoint*. Untuk meredam osilasi maka dipasanglah kombinasi pengendali PD ($K_d=10$), sehingga respon sistem menjadi seperti grafik di bawah ini:

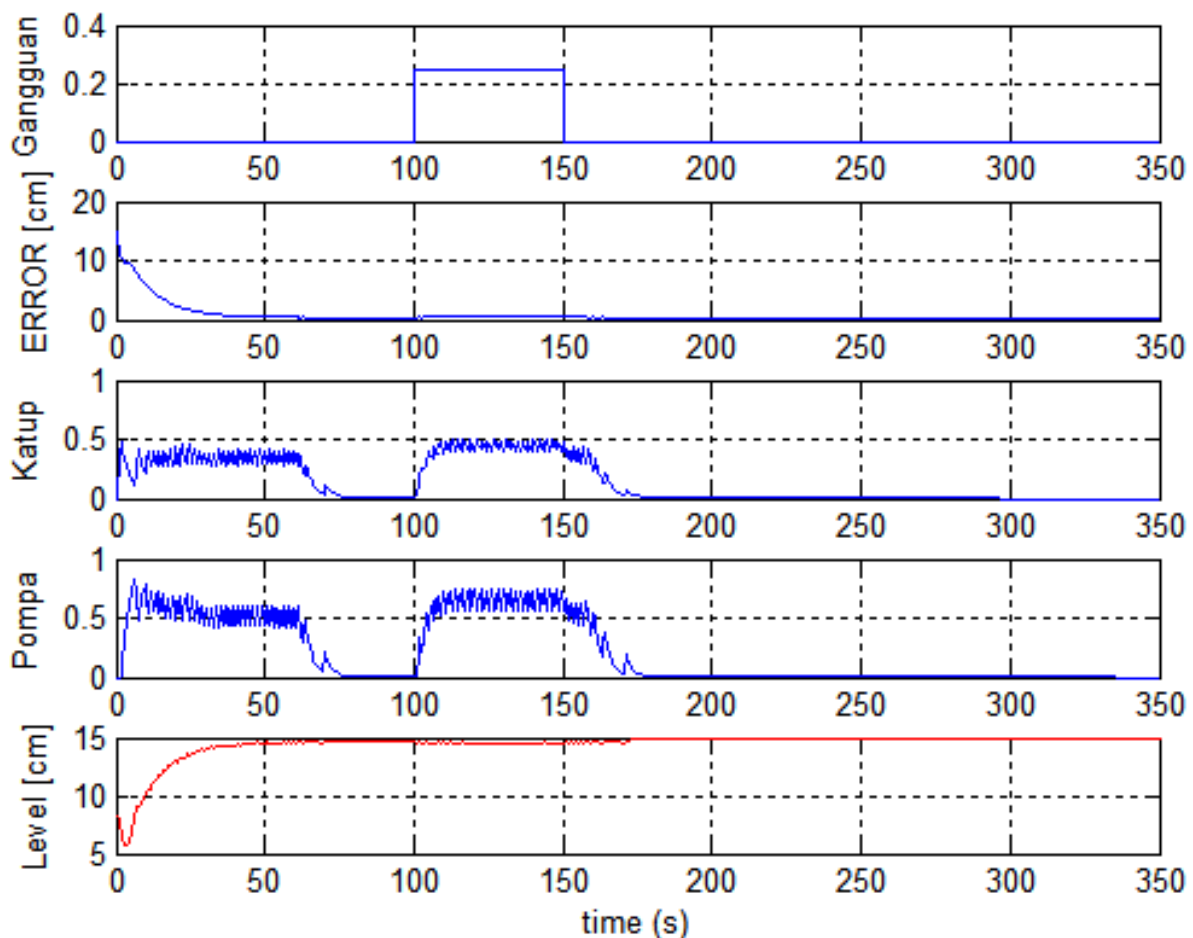
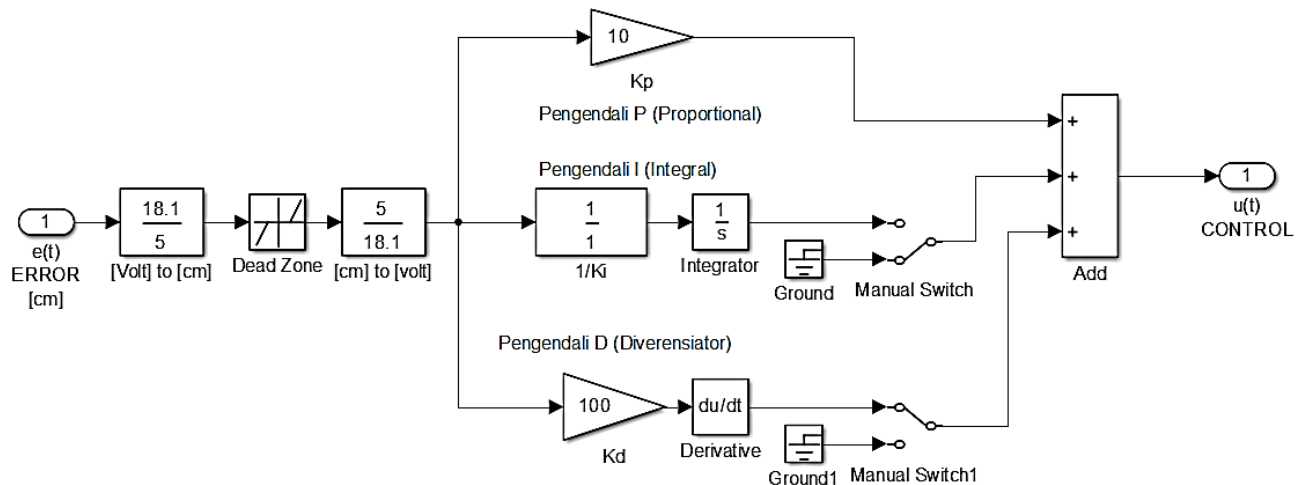


Jelas terlihat bahwa osilasi yang dihasilkan sudah cukup teredam. Untuk lebih meredam respon level ketinggian fluida pada tangki, kita hanya perlu menaikkan nilai koefisien differensial (K_d) menjadi cukup besar untuk meredam osilasi yang terjadi, misalnya $K_d=50-100$. Berikut adalah contoh respon sistem dengan nilai $K_d=100$:



[CHAIDIR ANWAR] / [D411 12 012]

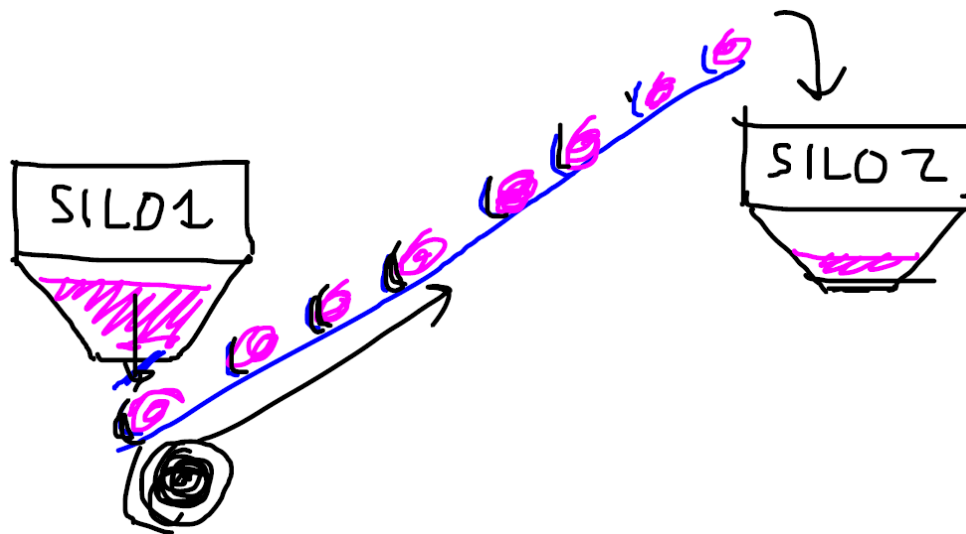
Sekarang, respon output sudah sangat stabil. Masalah berikutnya yang muncul adalah walaupun telah mencapai ketinggian fluida yang sangat mendekati *setpoint*, katup dan pompa terus bekerja selama ada *error* yang tercipta, walaupun nilainya sangat kecil. Ini bisa diatasi dengan memberikan koreksi *error*, berupa toleransi *error* yang masih dapat diterima. Misalnya untuk toleransi *error* yang baik untuk kondisi *hysteresis* relay dari -0.2 s/d 0.2 adalah ± 0.4 cm. Sedangkan untuk kondisi *hysteresis* relay dari -0.1 s/d 0.1 toleransi ± 0.1 cm sudah cukup baik. Berikut ini blok diagram lengkap dari *subsystem* PID beserta grafik respon sistem terhadap kondisi *hysteresis* relay dari -0.2 s/d 0.2 dengan toleransi ± 0.4 cm:



Respon sistem seperti inilah yang dicari. Level ketinggian fluida dalam tangki akan tetap stabil (dengan toleransi *error* tertentu) walaupun ada gangguan yang muncul pada katup. Sedangkan apabila telah berada pada level ketinggian fluida yang ditentukan pada *setpoint*, maka katup dan pompa akan berhenti aktif, dan mencegah pompa serta katup beroperasi secara berlebihan.

BAB III

PROYEK SILO-TO-SILO



Proyek ini adalah membuat suatu model proses untuk memindahkan benda berupa pasir antara dua tempat penampung (disebut silo) menggunakan *conveyor* –lebih tepat disebut *elevator*–. *Conveyor* ini dilengkapi cawan-cawan sebagai penahan tiap gundukan pasir. Silo yang digunakan berbentuk tabung pada bagian atas, dan kerucut terpancung terbalik pada bagian bawah. Ada dua silo yang digunakan, disebut Silo1 dan Silo2. Konfigurasi instalasi dapat dilihat pada gambar di atas.

Silo1 adalah tempat tampungan awal pasir. Pada bagian bawah Silo1 terdapat sebuah katup proporsional. Mulut katup akan langsung mengarah ke sebuah *conveyor*. *Conveyor* ini dilengkapi dengan cawan-cawan yang akan menahan gundukan pasir yang jatuh dari katup sehingga tidak jatuh selama perjalanan. *Conveyor* akan digerakkan oleh motor –dipisahkan oleh roda gigi–, sehingga cawan tersebut akan membawa setiap gundukan ke bagian ujung lainnya dari *conveyor* untuk ditumpahkan pada Silo2. Silo2 inilah yang akan kembali menampung setiap gundukan pasir yang telah dikeluarkan oleh Silo1. Pada Silo2, mulut katup akan tetap dibuat tertutup.

Berikut ini adalah data-data teknis yang dibutuhkan dalam perancangan model proses:

1. Silo1

-**Katup** (proporsional; debit maks 1liter/20detik)

-**Penampung**

Diameter tabung 26 cm

Tinggi tabung 20.5 cm

Diameter atas kerucut terpancung terbalik 26 cm

Diameter bawah kerucut terpancung terbalik 4.5 cm

Tinggi kerucut terpancung terbalik 12 cm

2. Silo2

-**Katup** (proporsional; debit maks 1liter/20detik)

-**Penampung**

Diameter tabung 26 cm

Tinggi tabung 10.5 cm

Diameter atas kerucut terpancung terbalik 26 cm

Diameter bawah kerucut terpancung terbalik 4.5 cm

Tinggi kerucut terpancung terbalik 12 cm

3. Conveyor

Kecepatan Motor 250rpm, (*adjustable*)

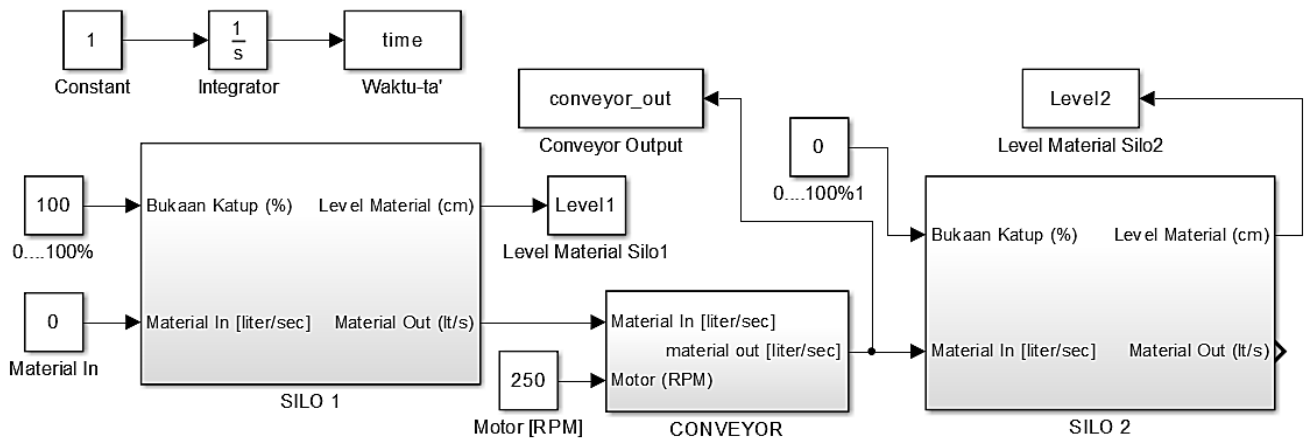
Roda gigi (kecepatan *conveyor*) 1:5 motor (artinya sebanyak 250/5 cawan ditumpahkan setiap menit)

Jumlah cawan di atas *conveyor* 10 cawan

Dari model di atas jelas terlihat bahwa ada tiga *subsystem* yang harus dibangun, Silo1, *conveyor* dan Silo2. Silo1 akan dapat diisi dengan material dari luar. Selain itu, perlu pula diatur seberapa besar katup pada Silo1 akan dibuka. Sebagai keluaran dari silo1, adalah material yang keluar dari katup. Untuk mengamati level material (dalam cm) pada Silo1, dapat pula ditambahkan sebuah variabel *monitoring* “*Level1*” pada keluaran lain dari Silo1.

Material keluaran dari Silo1 akan jatuh pada *conveyor* dan membentuk gundukan-gundukan pasir yang berjalan diatasnya. Selain banyaknya pasir dari Silo1 yang menjadi masukan dari *conveyor*, kecepatan motor perlu pula diatur untuk menentukan seberapa cepat gundukan material akan dibawa dan ditumpahkan ke Silo2. Untuk itu pada *conveyor* dibuat dua parameter masukan, yaitu *material_in* dan kecepatan motor untuk *conveyor*. Sedangkan hanya akan dibuat keluaran tunggal untuk *conveyor*, yaitu *material_out*, yakni berupa gundukan-gundukan pasir yang dibawa dan dijatuhkan pada Silo2. Akan dipasang sebuah variabel *monitoring* dengan nama “*_out*” untuk *memonitoring* bentuk keluaran dari *conveyor*.

Sedangkan untuk Silo2, secara blok diagram akan sama saja dengan Silo1, hanya saja akan ada sedikit perbedaan terkait dengan parameter-parameter saturasinya. Seperti halnya Silo1, pada Silo2 juga akan ditambahkan sebuah variabel *monitoring* dengan nama “*Level2*” untuk memantau ketinggian sesaat pasir pada Silo2. Sebagai catatan, katup pada Silo2 ini akan selalu dibuat tertutup, agar tidak ada material yang terbang dengan percuma. Secara lengkap, model umum yang telah kita capai saat ini ditunjukkan pada gambar berikut:



1. SILO1

Telah dibahas sebelumnya bahwa bukaan katup merupakan salah satu parameter masukan dari Silo1. Selain bukaan katup, *Material_In* merupakan parameter masukan lainnya. Jika bukaan katup dinyatakan dalam persen[0..100], maka nilai bukaan ini perlu terlebih dahulu dikonversi ke nilai yang sebenarnya, yaitu dengan membagi nilai bukaan tersebut dengan 100. Untuk membagi, kita dapat menggunakan sebuah *transfer function*. Dari data teknis diketahui bahwa katup memiliki debit 1 liter / 20 detik untuk bukaan penuh, atau setara dengan 0.05liter/detik. Seperti biasanya, kita akan membandingkan nilai bukaan katup terhadap nilai maksimum untuk mendapatkan debit yang sebanding dengan bukaan katup tersebut. Untuk membandingkan, kita dapat kembali menggunakan sebuah blok *product*. Hasil dari membandingkan dengan blok ini akan memiliki keluaran berupa debit keluarnya material dari Silo1 dalam liter-per-detik, sebanding dengan nilai bukaan katup yang diberikan.

Jika *material_in* dalam liter/detik adalah debit masuknya material baru pada Silo, sedangkan bukaan katup menunjukkan debit keluarnya material dari Silo, maka debit pertambahan material pada Silo adalah *material_in* dikurangi debit bukaan katup. Untuk itu, dari parameter masukan *material_in* pada Silo langsung dapat dikurangkan dengan debit bukaan katup dalam satuan yang sama. Digunakan sebuah blok *sum*. Sekarang kita telah mendapatkan debit pertambahan material rata-rata sesaat dalam liter/detik pada Silo.

Seperti teknik pada proyek sebelumnya, jika debit adalah turunan volume terhadap waktu, maka volume merupakan integral dari debit terhadap waktu. Untuk itu, keluaran dari blok *sum* akan dimasukkan pada sebuah blok *integrator* untuk mendapatkan volume dari Silo dalam liter. Blok *integrator* yang digunakan adalah blok yang dilengkapi *saturation*. Alasannya masih sama, agar volume pada Silo tidak akan pernah bernilai negatif, sekalipun dilakukan pembatasan tinggi material pada bagian akhir blok diagram. Penjelasan lebih rinci tentang blok ini dapat dilihat pada pembahasan sebelumnya (halaman 6-7 sub-bab 1.3. *level controlled tank*).

Sebagai referensi, untuk kerucut terpancung berlaku rumus $V = \frac{1}{3}\pi h(R^2 + Rr + r^2)$, dengan h adalah tinggi kerucut terpancung, R jari-jari tutup besar, dan r jari-jari tutup kecil. Dengan rumus ini ditambah dengan rumus volume tabung, didapatkan bahwa volume maksimum dari Silo1 adalah 13.4389 liter. Untuk itu pada batasan blok *integrator* yang digunakan diberi batasan saturasi dari 0 sampai 13.4389. Sedangkan *initial condition* merupakan volume awal dari Silo1 (misalnya 5 liter).

Variabel *monitoring* “*Levell*” yang dipasang bertujuan untuk memantau level ketinggian material pada Silo1 dalam satuan cm. Untuk itu perlu dilakukan konversi dari volume material dalam liter menjadi ketinggian material dalam cm. Dalam melakukan tugas ini, pilihan paling mudah adalah dengan menggunakan sebuah blok “*Interpreted MATLAB Fcn*”, yakni sebuah blok yang memiliki respon keluaran berdasarkan *listing program* yang diketikkan oleh pengguna. Untuk melakukan tugas ini, berikut *listing program* yang digunakan:

```
function y = volume_to_level(x)
V = 1000*x; % mendefinisikan volume sebagai masukan dalam liter
R = 26/2; % Jari-jari silinder/tutup besar kerucut
r = 4.5/2; % Jari-jari katup /tutup kecil kerucut
h = 12; % Tinggi kerucut terpancung tersisa
Vk = (1/3)*(pi)*(R^2 + (R*r) + r^2)*h; % Volume kerucut terpancung
ho = (r/(R - r))*h; % Tinggi potongan kerucut (yang terpotong)
if V >= Vk %Jika kerucut telah penuh terisi material
    y = h + ((V-Vk)/(pi*(R^2))); % level material adalah tinggi kerucut ditambah tinggi volume pada tabung
else % Jika kerucut belum penuh
    y = (((3*V*(ho^2)) + (pi*(r^2)*(ho^3)))/(pi*(r^2)))^(1/3) - ho;
end
```

Untuk kerucut yang belum penuh, digunakan dari persamaan $t = \sqrt[3]{\frac{\pi h^3 r^2 + 3 V h^2}{\pi r^2}} - h_0$.

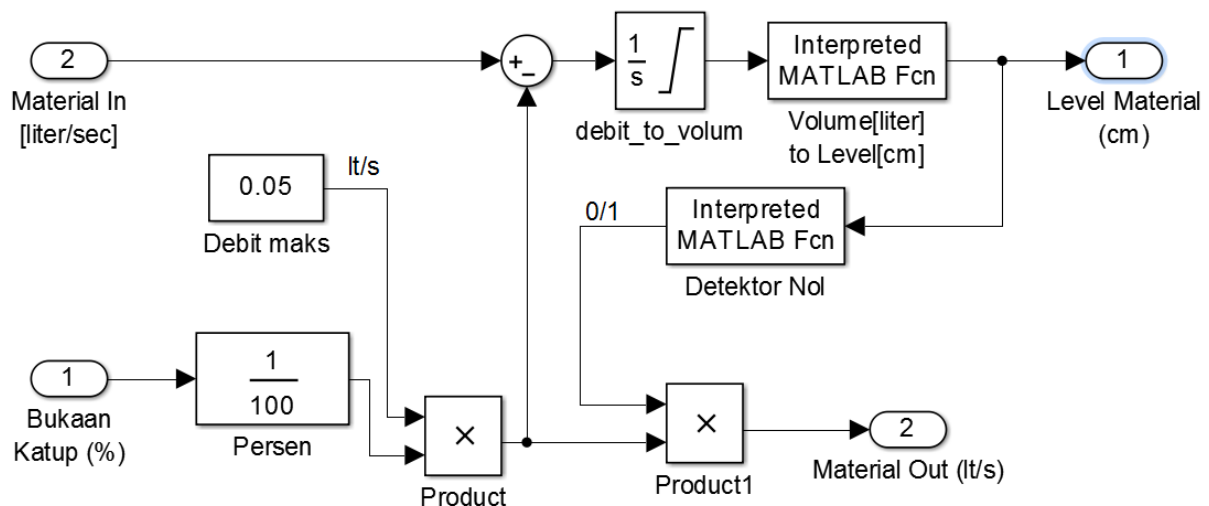
Keluaran dari blok fungsi terprogram ini dapat langsung menjadi keluaran dari *subsystem* Silo1 untuk dipantau melalui variabel *monitoring* “*Levell*”. Namun, masalah yang muncul adalah bagaimana seandainya bila katup terus terbuka sedangkan material pada Silo1 telah habis. Pada *integrator* mungkin tidak akan menjadi masalah, karena telah diberi batasan sebelumnya. Masalah yang sebenarnya adalah pada material keluaran (*material_out*) dari Silo1 yang akan diterima oleh *conveyor*. Perlu ada sebuah blok pembatas yang me-nolkan material keluaran saat material di dalam Silo telah habis. Untuk itu, digunakan sebuah blok *Interpreted MATLAB Fcn*(disebut *detektor_nol*) dari level ketinggian material (dalam cm), yang keluarannya dikalikan dengan debit keluaran

Silo1. Jika material belum habis, blok *Interpreted MATLAB Fcn* akan memberikan isyarat 1, sehingga tidak akan mengganggu nilai debit keluaran dari Silo1. Sedangkan apabila material sudah habis, akan diberikan isyarat 0, sehingga debit hasil perkalian akan ikut menjadi nol. Dari hasil perkalian inilah yang benar-benar sudah dapat menjadi keluaran dari *subsystem* Silo1 untuk ditujukan ke *conveyor*.

Berikut adalah *listing program* dari blok *detektor_nol*:

```
function y=detektor_nol(x)
if x==0
    y=0;
else
    y=1;
end
```

Berikut adalah blok diagram dari *subsystem* Silo1 secara utuh:



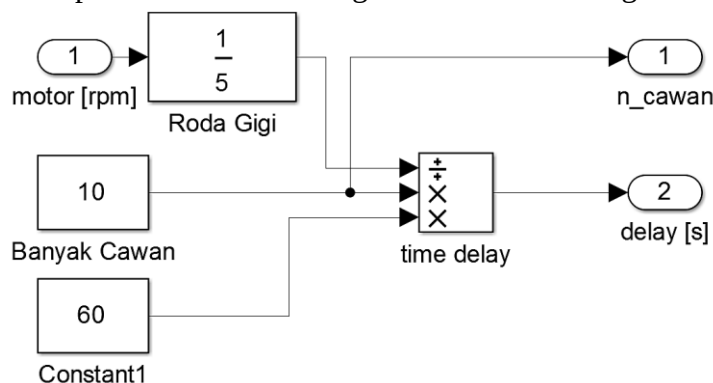
2. CONVEYOR

Subsystem ini bertujuan untuk mengubah volume yang masuk akan mengalami perubahan bentuk menjadi gundukan-gundukan material pada setiap cawannya, disertai dengan *delay* akibat dari proses memindahkan material. Kedengarannya sederhana, tapi *subsystem* inilah yang paling rumit.

Akan ada dua masukan dari *subsystem* ini, yakni *material_in* dan kecepatan motor *conveyor* dalam rpm. Akan kembali dibuat dua buah *subsystem*, yakni *subsystem* yang menerangkan data teknis dari *conveyor*, serta *subsystem* yang membentuk material masukan menjadi gundukan-gundukan material.

2.1.KETERANGAN SISTEM

Bagian inilah yang akan menerangkan data-data teknis dari *conveyor* sehingga menjadi hal berguna untuk bagian lainnya. Jika diketahui kecepatan motor sebagai parameter masukan adalah x rpm, maka *conveyor* akan menjatuhkan sebanyak $x/5$ cawan dalam setiap menit, atau setiap cawan membutuhkan $60 : \frac{x}{5}$ detik untuk dijatuhkan. Karena ada 10 buah cawan di atas *conveyor*, maka dari mulai mengisi cawan sampai mulai menjatuhkannya pada ujung satunya, dibutuhkan waktu selama 10 kali dari waktu menjatuhkan 1 cawan, sebagai waktu *delay* dalam detik. Jika keluaran dari *subsystem* ini adalah banyak cawan (n_cawan) dan *delay*, maka sistem ini dapat direalisasikan dengan susunan blok diagram sebagai berikut:



2.2.PEMBENTUK GUNDUKAN

Subsystem inilah yang paling rumit. *Subsystem* ini dibuat dengan tiga buah parameter masukan; *material_in*, n_cawan (jumlah cawan), dan *delay*.

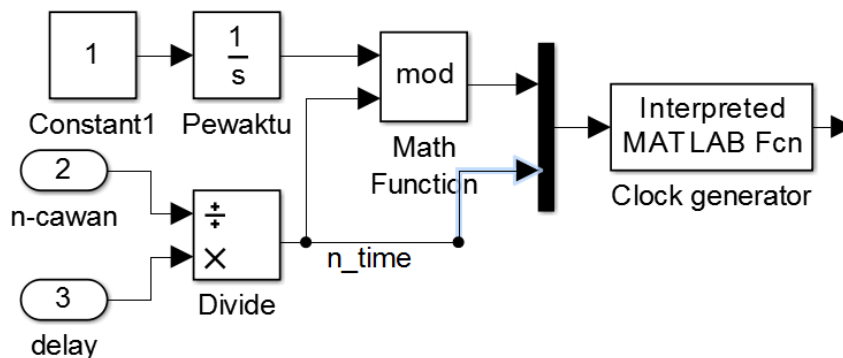
Pertama-tama, kita akan membutuhkan sebuah nilai yang menunjukkan waktu untuk mengisi setiap cawan dari tumpahan katup Silo1 (dinamakan n_time). n_time dapat dengan mudah didapatkan dengan membagi waktu *delay* total terhadap jumlah cawan yang ada (n_cawan). Waktu ini akan menjadi salah satu keluaran *subsystem*, karena setara dengan waktu yang dibutuhkan untuk memproses 1 buah gundukan pada setiap cawan, sehingga disebut juga *process time*.

Rangkaian pembentuk gundukan akan dibuat bekerja berdasarkan jumlah volume dari Silo1 yang diterima setiap waktu pengisian 1 cawan. Artinya yang kita butuhkan sekarang adalah sebuah pewaktu yang menentukan kapan volume dari setiap gundukan material akan mulai dihitung, menahan volume tersebut sampai gundukan berikutnya, dan mengulang parameter waktu sebagai parameter fungsi untuk pembentuk gundukan –akan dibahas kemudian.

Sebagai pembuka, kita akan menghitung volume material yang diterima *conveyor* untuk mengisi satu buah cawan dan menjadi sebuah gundukan. Volume ini dapat kembali kita dapatkan dengan blok *integrator*. Namun, kita harus terus mengulang menghitung untuk gundukan-gundukan berikutnya. Untuk itu kita membutuhkan sebuah blok *integrator* yang dilengkapi dengan kemampuan *external reset*. Jika kita

akan melakukan *reset* setiap selesai menghitung volume untuk satu gundukan, dan yang kita gunakan adalah mode *reset* tepi turun (*falling*), maka *clock* yang kita butuhkan adalah memiliki tepi turun setiap pergantian satu periode *clock*. Artinya *clock* yang akan dibuat adalah bernilai nol selama setengah periode awal *clock*, dan bernilai satu pada setengah periode berikutnya. Untuk melakukannya, kita dapat menggunakan sebuah blok *Interpreted MATLAB Fcn*, namun dengan beberapa parameter masukan.

Kita dapat menggunakan blok diagram beserta *listing program* berikut untuk membangkitkan *clock* yang tepat:



```
function y=fclock(x)
waktu=x(1); %waktu dari sisa bagi waktu terhadap waktu
n_time
delay=x(2); %waktu untuk memproses setiap cawan (n_time)
banding=delay/2; %setengah periode n_time
if waktu<banding
    y=0; %output 0 selama setengah siklus pertama clock
else y=1; %output 1 selama setengah siklus berikutnya
end
end
```

Pertama kita akan membutuhkan sebuah pewaktu yang terus membangkitkan waktu selama simulasi dijalankan. Sisa hasil bagi waktu terhadap waktu setiap cawan (*n_time*) akan menghasilkan nilai dari 0 sampai *n_time*, yakni lamanya waktu setiap cawan dari mulai diisi. Bisa didapatkan dengan menggunakan fungsi “mod” dari blok *Math Function*. *n_time* juga menunjukkan lamanya waktu untuk 1 periode *clock*. Dari range nilai ini, kita dapat membuat keluaran akan bernilai 0 selama sisa hasil bagi lebih kecil dari setengah siklus *clock*, dan akan bernilai 1 selama setengah siklus berikutnya, yakni jika sisa hasil bagi telah lebih besar dari setengah siklus *clock*. Akhirnya kita telah mendapatkan sebuah sistem *clock* yang memberikan tepi turun setiap volume untuk 1 cawan telah selesai dihitung.

Setelah selesai dihitung volume untuk setiap cawan, perlulah membentuk gundukan material dengan volume yang tepat sama dengan nilai tersebut. Kita dapat membentuk gundukan material namun dalam satuan liter/detik, karena sebagai masukan untuk silo

berikutnya, dapat digunakan sebuah blok *Interpreted MATLAB Fcn*. Jika keluaran berbentuk debit yang dibentuk seperti gundukan-gundukan (seperti grafik positif fungsi sinus), namun dengan tetap memperhatikan volume yang diterima, dengan ilmu kalkulus dapat dibentuk persamaan berikut:

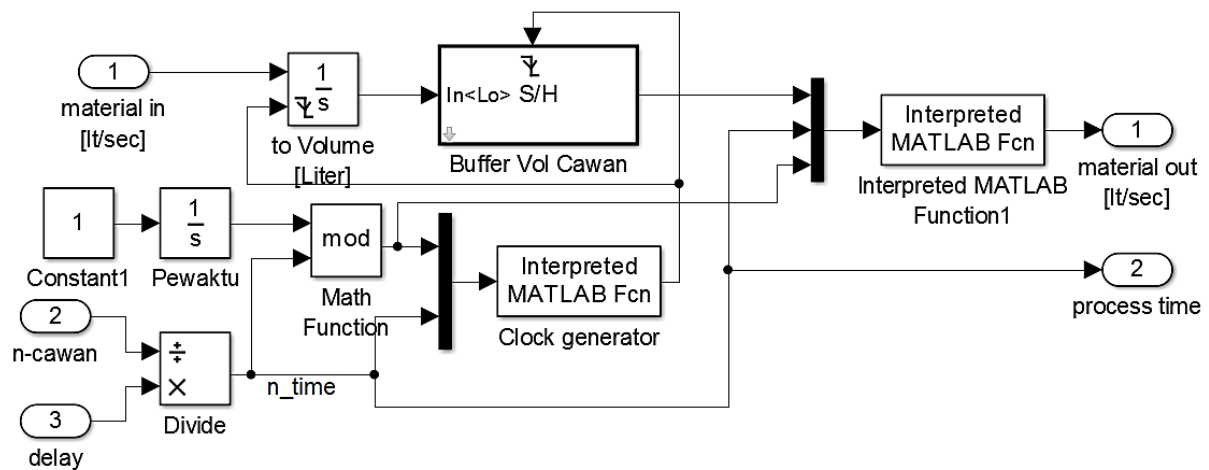
$$V_{in} = \int_0^{n_{time}} A \cdot \sin(\omega \cdot t) dt ; \text{ dengan } \omega = \frac{\pi}{n_{time}}$$

$$A = \frac{\pi \cdot V_{in}}{2 \cdot n_{time}}$$

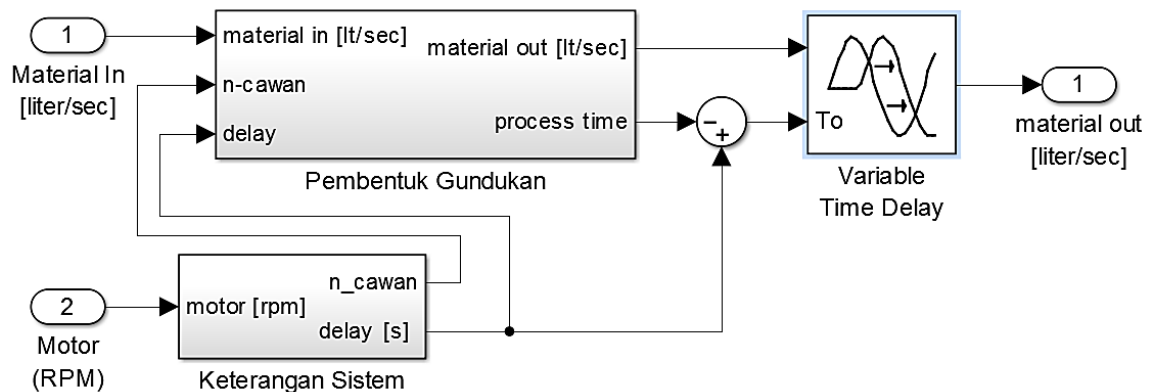
Dari persamaan ini, kita dapat membuat sebuah fungsi yang mendefinisikan volume yang tepat dari bentuk-bentuk gundukan yang dihasilkan, walau dalam satuan liter/detik. Namun, variabel *n_time* pada *listring program* berikut akan digantikan dengan variabel “waktu” yang ukurannya sengaja dibuat hanya 80% dari *n_time*, agar jarak antar satu gundukan dan gundukan berikutnya tetap terlihat. Berikut ini adalah *listring program*-nya:

```
function y=pembentuk_cawan(x)
vol=x(1); %volume untuk setiap cawan
n_time=x(2); %waktu pengisian tiap cawan
ftime=x(3); %parameter waktu
waktu=n_time-(n_time/5); %panjang gundukan dalam detik
omega=pi/waktu; %kecepatan sudut fungsi
A=0.5*pi*vol/waktu; %menentukan amplitudo yang tepat agar
volume sesuai
if ftime<=waktu
    y=A*sin(omega*ftime);
else y=0;
end
end
```

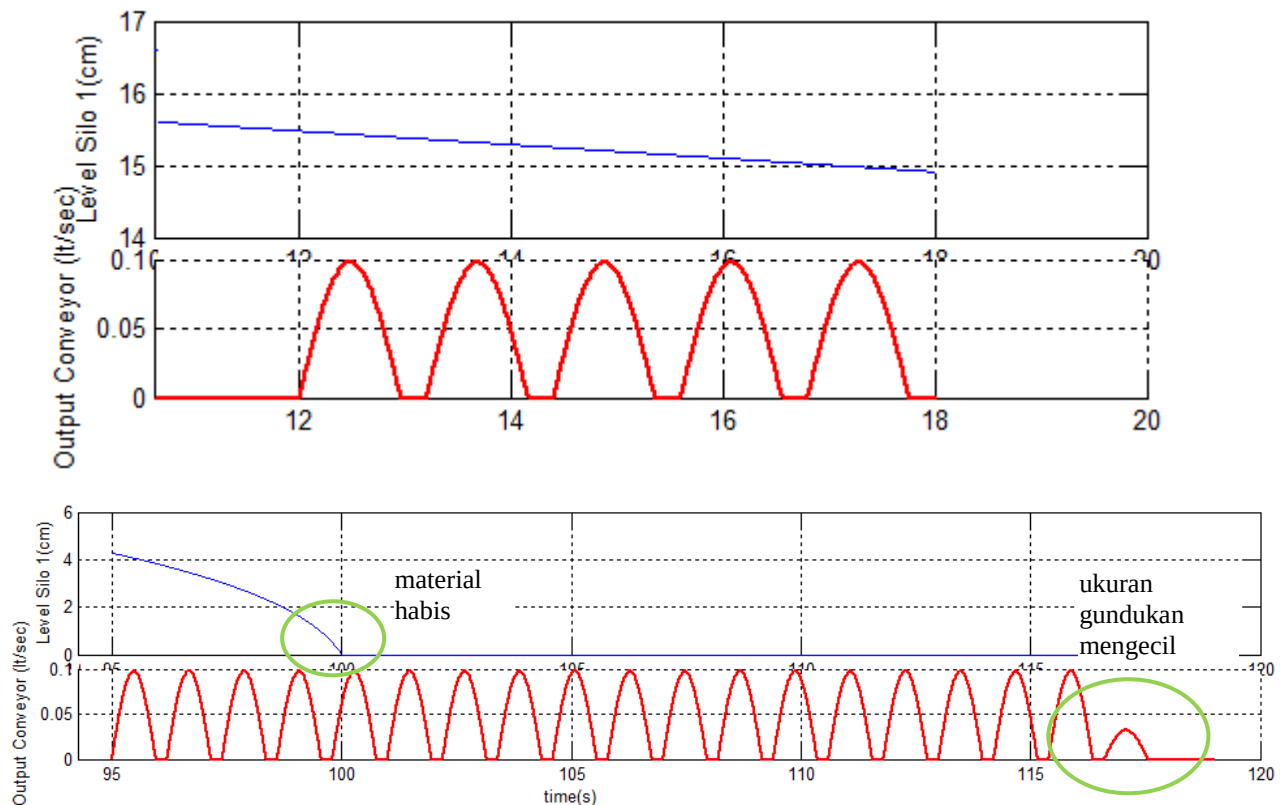
Namun, masalahnya adalah kita membutuhkan sebuah nilai masukan dari fungsi berupa volume yang akan dibuat gundukan berdasarkan hitungan dari blok *integrator*. Artinya, kita harus menahan hasil hitungan blok *integrator* tepat sebelum nilainya di-selama satu periode *clock*. Untuk melakukannya, kita membutuhkan sebuah blok yang tepat, *Sample and Hold*, dengan *clock* yang dibuat sama dengan *clock* reset dari blok *integrator* tersebut. Secara lengkap, berikut ditunjukkan blok diagram dari *subsystem* Pembentuk Gundukan:



Subsystem ini akan memakan waktu sebesar waktu untuk mengisi 1 cawan. Artinya, kita masih membutuhkan waktu *delay* sebesar 9 kali proses seperti ini lagi. Nilai *delay* sebesar itu adalah sama saja mengurangi waktu *delay* dari *subsystem* “*Keterangan Sistem*” dengan waktu proses yang dipakai oleh blok *subsystem* “*Pembentuk Gundukan*”. Untuk memberikan *delay*, kita dapat menggunakan sebuah blok *Variable Time Delay* sebelum gundukan-gundukan material dibawa keluar *subsystem*, dengan nilai *delay* dari hasil pengurangan tersebut. Berikut ini adalah gambaran blok diagram *conveyor* secara lengkap:



Berikut ini respon keluaran dari *conveyor* dengan kecepatan motor 250rpm:

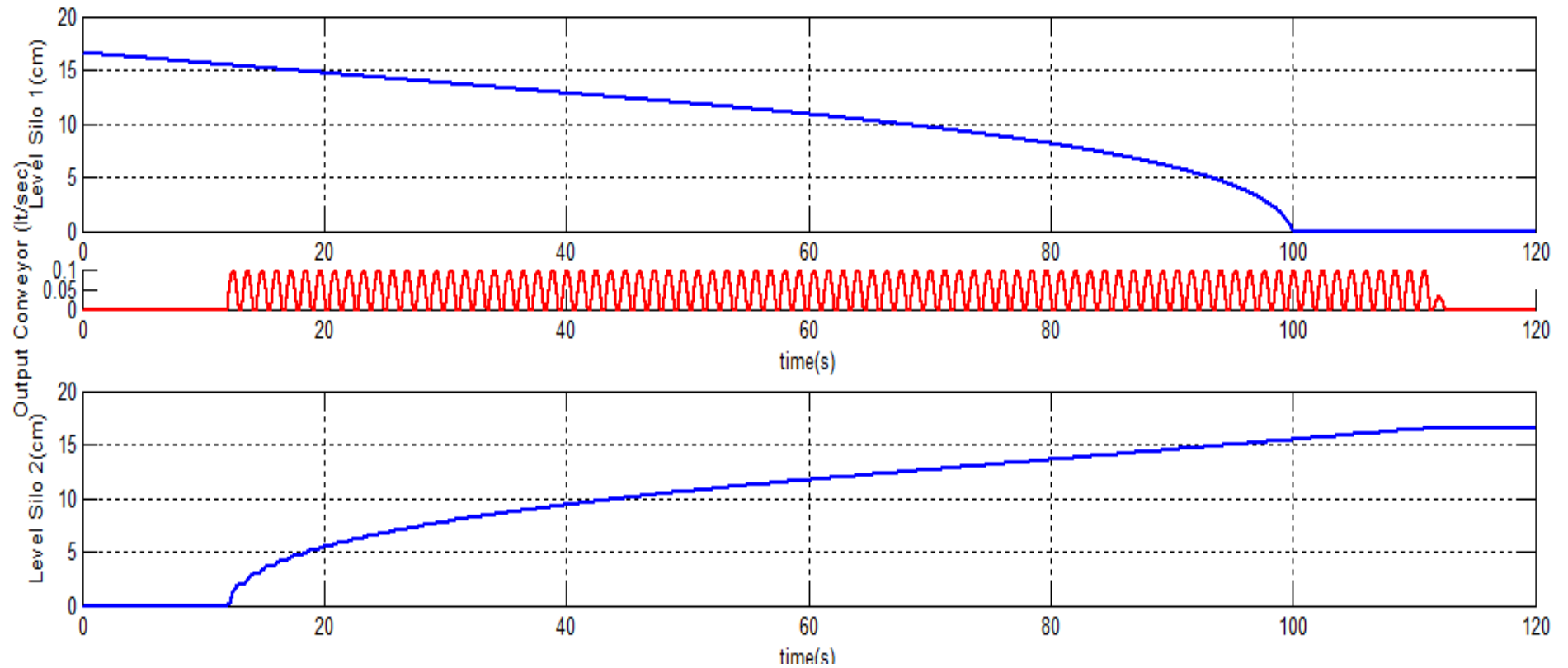


Terlihat bahwa respon keluaran dari *conveyor* berbentuk gundukan-gundukan material yang amplitudonya menyesuaikan dengan jumlah material yang masuk.

3. SILO2

Secara umum, blok diagram Silo2 akan sama saja dengan blok diagram pada Silo1, hanya saja Silo2 hanya dapat menampung material yang lebih sedikit dari Silo1. Untuk tinggi tabung pada Silo2, hanya mampu menampung material hingga 8.1296 liter. Untuk itu, perbedaan dari *subsystem* pada Silo2 hanya terletak pada batasan saturasi dari blok *integrator*-nya.

Berikut adalah grafik respon pemindahan material dari model Silo_to_Silo ini:



Respon yang baik adalah apabila ketinggian awal pada Silo1 akan sama dengan ketinggian akhir pada Silo2.

BAB IV

PENUTUP

KESIMPULAN

Sebagai mahasiswa elektro di bidang teknik kendali, mempelajari pemodelan sistem akan sangat penting. Selain untuk memonitoring sistem, pemodelan dapat membantu dalam melakukan analisis terhadap kesalahan-kesalahan sistem yang mungkin terjadi untuk meminimalisir kerugian dalam suatu proses. Namun, dibutuhkan keahlian yang baik dalam memodelkan sistem secara benar, karena pemodelan yang kurang tepat tentu saja dapat mengakibatkan kesalahan dalam memonitoring suatu proses.

2014

Tidak diperlukan kertas tambahan, tuliskan semua jawaban pada tempatnya, gunakan halaman kosong di sebalik.

1. Teknologi Kendali Proses dimanfaatkan dalam Sistem Kendali Proses. Terangkan apa yang dimaksud dengan istilah "PROSES" di sini, sebutkan 2 (dua) kategori PROSES yang ada, terangkan apa bedanya antara keduanya! (15 points)
2. Sebutkan 3 (tiga) keadaan (*state*) dari material yang di-proses pada kondisi lingkungan tertentu:
(5 points)
(1) Keadaan _____ (*solid state*)
(2) Keadaan _____ ()
(3) Keadaan _____ ()
3. Gambarkan di bawah ini bagan kotak dari suatu daur kendali proses (*Process Control Loop*) sehingga tampak jelas yang mana: *setpoint (SP)*, peubah kendali (*MV*), peubah proses (*PV*) dan gangguan(10 points):
4. Terangkan apa yang dimaksud dengan "tetapan waktu" (*time-constant, τ*), gunakan contoh rangkaian pemuatan kapasitor *RC-series* sebagai penjelasan! (10 points):

Teknologi Kendali Proses
TEST, OPEN BOOK, LAP-TOP is OK (100 menit)

Nama: _____ NIM: _____ Paraf: _____

Tidak diperlukan kertas tambahan, tuliskan semua jawaban pada tempatnya, gunakan halaman kosong di sebalik.

5. Dengan suatu persamaan differensial order kedua, terangkan yang mana dikatakan sebagai tetapan waktu (*time constant T*), nisbah redaman (*damping ratio, ξ*) dan penguatan (bati) sistem (*the system gain, K_p*) untuk suatu proses dengan keluaran **$y(t)$** dan masukan **$u(t)$** (15 points) :

(a) Persamaan Differensial: _____

(b) Lengkapilah tabel berikut:

Nisbah Redaman	Karakteristik Proses
$\xi = 0$	Osilasi tak teredam (<i>undamped oscillation</i>)
$0 < \xi < 1$	
$\xi = 1$	
$\xi > 1$	

6. Dengan menggambarkan tanggapan undak-(*step-response*)-nya masing-masing, jelaskan bedanya "*time-(transport) delay*" dan "*lagging*"! (10 point)

7. Pengendali **P** : *Proportional controller*, metode dasar pengendalian, isyarat kendali sebanding dengan galat (*error*), sedangkan (lengkapi, 15 points):

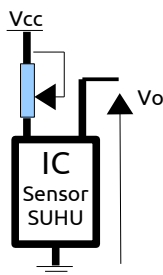
(a) Pengendali **PI** : _____

(b) Pengendali **PD**: _____

(c) Pengendali **PID**: _____

8. (a) Suatu sensor suhu berbasis rangkaian terintegrasi (**IC**) di-*setting* sedemikian rupa sehingga menghasilkan tegangan **2,6 Volt** pada suhu ruangan **26°C**. Jika diketahui karakteristik **IC** tersebut *linear-proportional* dengan kenaikan **10 mV/°C**, tentukanlah tegangan yang dihasilkan jika **IC** tersebut dicelupkan dalam air mendidih **100°C** (10 points):

Jawab:



- (b) Gambarkan konfigurasi rangkaian *3-Wire RTD* untuk suatu sistem instrumentasi digital. Jika digunakan sumber arus **1 mA** dan **$R_L = 1 \text{ k}\Omega$** , tentukan **R_x** supaya tegangan yang terpantau (**V_B**) **2,6 Volt** (10 points).

Jawab: