

repository.ub.ac.id

**SIMULASI SISTEM PENGENDALIAN KETINGGIAN AIR PADA  
TANGKI TERHUBUNG DENGAN STRUKTUR *OUTPUT*  
*FEEDBACK CONTROL***

**SKRIPSI**

**TEKNIK ELEKTRO KONSENTRASI TEKNIK KONTROL**

Diajukan untuk memenuhi persyaratan  
memperoleh gelar Sarjana Teknik



**FARIS NAUFAL MUSTHAF A**  
**NIM. 125060300111021**

**UNIVERSITAS BRAWIJAYA**

**FAKULTAS TEKNIK**

**MALANG**

**2017**

UNIVERSITAS BRAWIJAYA



## PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas di dalam Naskah Skripsi ini adalah asli dari pemikiran saya. Tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu Perguruan Tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur jiplakan, saya bersedia Skripsi dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

Malang, 27 Desember 2016

**Mahasiswa,**

**FARIS NAUFAL MUSTHAFA**

**NIM. 125060300111021**





## RINGKASAN

**Faris Naufal Musthafa**, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, April 2016, Simulasi Sistem Pengendalian Ketinggian Air Pada Tangki Terhubung Dengan Struktur *Output Feedback Control*, Dosen Pembimbing: Erni Yudaningtyas dan Mochammad Rusli.

*Plant* tangki terhubung merupakan gabungan dari 2 *plant* tangki. *Plant* tangki 2 memiliki *error steady state* lebih besar daripada *plant* tangki 1. Sistem pengontrolan ketinggian air bertujuan agar ketinggian air pada 2 *plant* tangki tetap sesuai dengan setpoint yang diinginkan. Ada banyak metode dalam perancangan sistem kontrol, salah satunya adalah perancangan sistem kontrol menggunakan metode penempatan kutub (*pole placement*). Setelah model matematika dapat diperoleh, langkah selanjutnya adalah menentukan performansi sistem yang diinginkan dan akhirnya dapat dihitung besarnya gain umpan balik keadaan. Salah satu struktur sistem kontrol dengan metode *pole placement* adalah *Output Feedback Control*. OFC disertai dengan *observer/estimator* yang digunakan untuk mengestimasi *state-state* pada pengontrolan ketinggian air berdasarkan model *plant*. *Output plant* setelah diberi kontroler dapat meminimalkan *error steady state*.

**Kata kunci** : *Plant* tangki terhubung, *Error Steady State*, *Pole Placement*, OFC.

## SUMMARY

**Faris Naufal Musthafa**, *Department of Electrical Engineering, Faculty of Engineering University of Brawijaya, April 2016, Simulation of Water Level Control System in Coupled Tanks with Structure Output Feedback Control, Academic Supervisor: Erni Yudaningtyas and Mochammad Rusli.*

*Coupled Tank plant forms combination two tanks plant which have more error steady state than one tank plant. Water height control system as a purpose to make water height on two tanks plant appropriate with setpoint that desired. There are many methods in designing control systems, one of them is pole placement method. After mathematical model is obtained, the next step is to determine the system's performances that is wanted and finally, the values of the state feedback gain can be calculated. One structure pole placement method control system is Output Feedback Control. OFC accompanied by observer/estimator is used to estimate states of controlling the water height based on the plant model. Plant output after being given controller can minimize error steady state.*

**Keywords :** *Coupled Tanks, Error Steady State, Optimal Control, Pole Placement, OFC.*





## PENGANTAR

*Bismillahirrohmanirrohim.* Alhamdulillah, puji syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Sistem Pengaturan Ketinggian Level Air pada Tangki Terhubung dengan Metode *Output Feedback Control* di Laboratorium Sistem Kontrol FTUB” dengan baik. Tak lepas shalawat serta salam tercurahkan kepada junjungan kita Nabi Muhammad SAW yang telah menjadi suri tauladan bagi yang mengharapkan rahmat dan hidayah-Nya.

Penulis menyadari bahwa penyusunan skripsi ini tidak lepas dari bantuan, bimbingan serta dorongan dari berbagai pihak. Pada kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar – besarnya kepada:

- Allah SWT yang telah memberikan kelancaran, kemudahan dan hidayah-Nya.
- Keluarga tercinta, kedua orang tua Badrul Musthafa dan Agustini Asriningtyas yang selalu memberikan kasih sayang dan doanya yang tiada akhir. Adik-adik tercinta Raihan, Faza, Irham yang selalu memberikan dukungan.
- Bapak M. Aziz Muslim, ST., MT., Ph.D. selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya.
- Bapak Hadi Suyono, ST., MT., Ph.D. selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya
- Bapak Ali Mustofa, ST., MT. selaku Ketua Program Studi S1 Jurusan Teknik Elektro Universitas Brawijaya
- Ibu Dr. Ir. Erni Yudaningtyas, MT. selaku Ka. Lab Sistem Kontrol sekaligus sebagai dosen pembimbing yang telah banyak memberikan kesempatan, nasehat, pengarahan, motivasi, saran dan masukan yang telah diberikan.
- Bapak Ir. Mochammad Rusli, Dipl. Ing. sebagai dosen pembimbing yang telah banyak memberikan kesempatan, nasehat, pengarahan, motivasi, saran dan masukan yang telah diberikan.
- Laboratorium Sistem Kontrol Teknik Elektro Universitas Brawijaya atas sarana dan prasarana yang dimanfaatkan penulis dalam melakukan penelitian.
- Pranata Laboratorium, Mbak Eka dan Keluarga besar asisten Laboratorium Sistem Kontrol, Yudha, Andri, Dennis, Avif, Suro, Hilmy, Indrut, Yudha Nur, Roni, Diana,

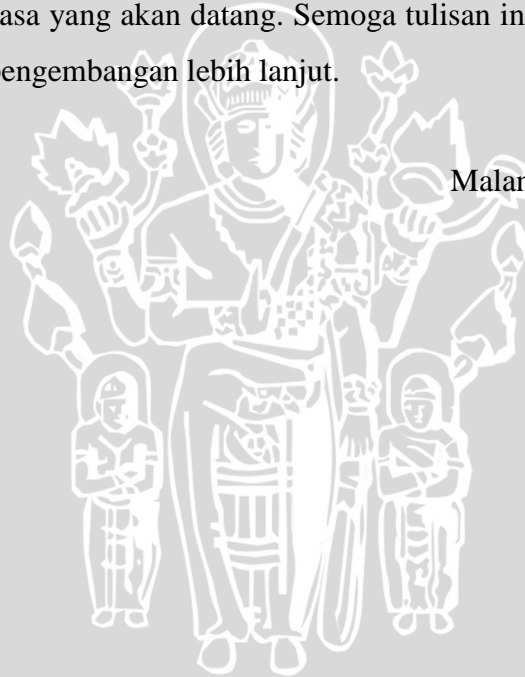
Ana, Zaini, Iqbal, Rifan terima kasih telah memberikan banyak bantuan, dukungan dan canda tawa.

- Kawan-kawan SMP-IT Al-Kahfi Wicak, Basum, Qide yang telah banyak berdiskusi dan berdialog tentang kehidupan pasca kampus
- Sahabat-sahabat dari maba, Fajar, Rizky, Rio, Wildan, Guntoro
- Keluarga besar Sistem Kontrol angkatan 2012, teman-teman angkatan 2012 “Voltage” atas do'a, semangat, serta dukungan yang diberikan pada penulis.
- Semua pihak yang telah memberikan bantuan serta dukungan baik secara langsung maupun tidak langsung atas penyusunan skripsi ini.

Dalam penyusunan skripsi ini, penulis menyadari bahwa skripsi ini belumlah sempurna, karena keterbatasan ilmu dan kendala-kendala lain yang terjadi selama pengerjaan skripsi ini. Oleh karena itu, penulis berharap kritik dan saran untuk penyempurnaan tulisan di masa yang akan datang. Semoga tulisan ini dapat bermanfaat dan dapat digunakan untuk pengembangan lebih lanjut.

Malang, Desember 2016

Penulis



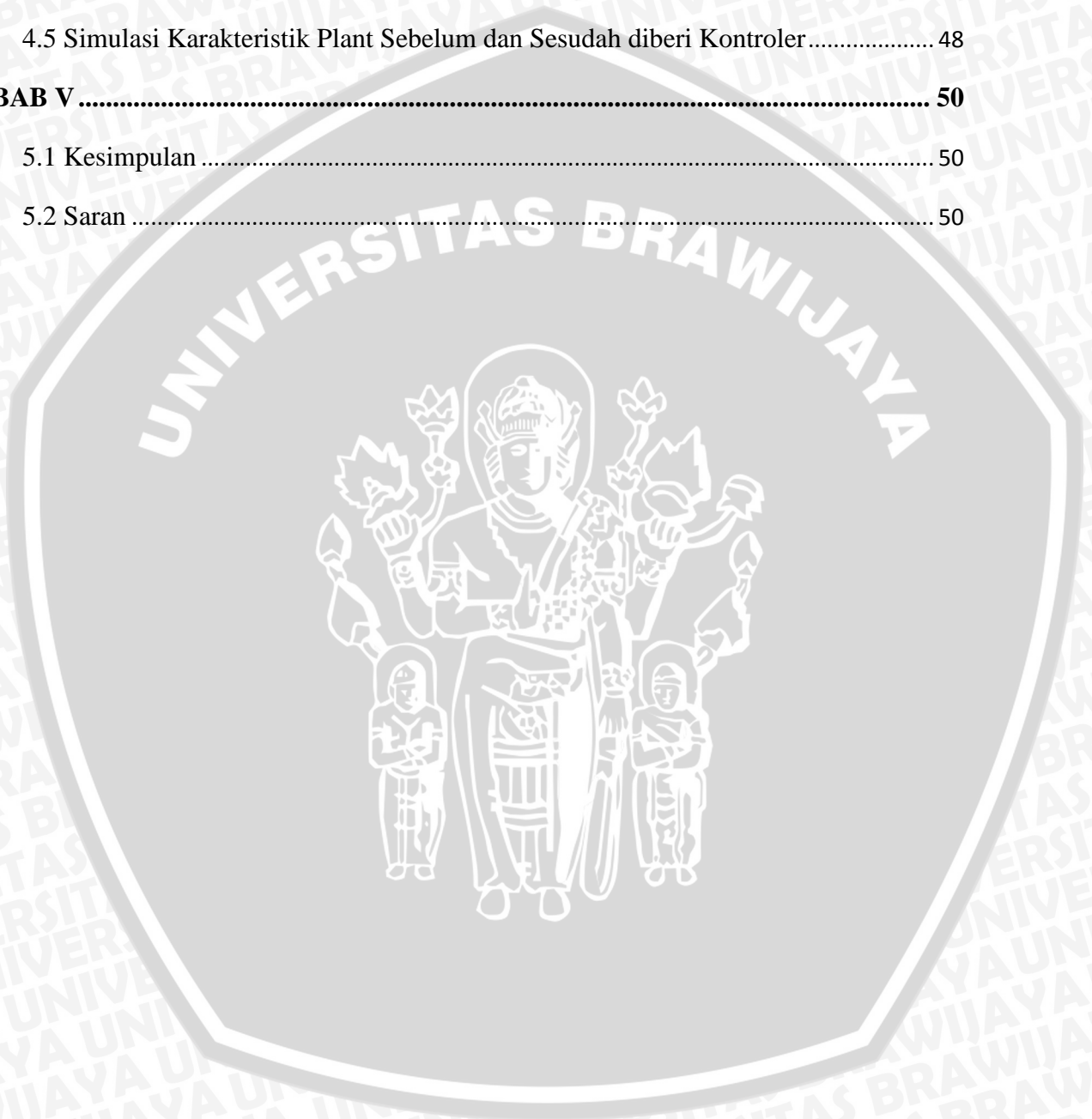
## DAFTAR ISI

|  |                              |
|--|------------------------------|
| <b>LEMBAR PERSETUJUAN.....</b>   | <b>i</b>                     |
| <b>PERNYATAAN ORISINALITAS SKRIPSI .....</b>   | <b>i</b>                     |
| <b>RINGKASAN.....</b>  | <b>ii</b>                    |
| <b>SUMMARY .....</b>   | <b>iii</b>                   |
| <b>PENGANTAR.....</b>  | <b>iv</b>                    |
| <b>DAFTAR ISI.....</b>   | <b>vi</b>                    |
| <b>DAFTAR TABEL.....</b>   | Error! Bookmark not defined. |
| <b>DAFTAR GAMBAR .....</b>   | Error! Bookmark not defined. |
| <b>BAB I PENDAHULUAN .....</b>   | <b>1</b>                     |
| 1.1 Latar Belakang .....   | 1                            |
| 1.2 Rumusan Masalah.....   | 2                            |
| 1.3 Batasan Masalah.....   | 2                            |
| 1.4 Tujuan.....  | 3                            |
| 1.5 Manfaat.....   | 3                            |
| <b>BAB II TINJAUAN PUSTAKA .....</b>   | <b>4</b>                     |
| 2.1 Mikrokontroler Arduino Mega.....   | 4                            |
| 2.2 Sistem Pengendalian.....   | 4                            |
| 2.3 Loop Pengendalian .....  | 5                            |
| 2.4 High Current 30 A DC Motor Driver dengan half bridge infineon BTS 7960 chip<br>..... | 7                            |
| 2.5 Tangki Terhubung.....  | 8                            |
| 2.6 Output Sistem Orde Satu.....   | 9                            |
| 2.7 Output Sistem Orde Dua .....   | 10                           |
| 2.7.1 Keadaan Kurang Teredam / Underdamped ( $0 < \xi < 1$ ) .....                       | 11                           |
| 2.7.2 Teredam Kritis / Critically Damped ( $\xi = 1$ ).....                              | 13                           |



|  |                                     |
|--|-------------------------------------|
| 2.7.3 Terlalu Teredam / Overdamped ( $\xi > 1$ ) .....   | 13                                  |
| 2.8 Tanggapan Peralihan .....  | 13                                  |
| 2.9 Kontroler .....  | 15                                  |
| 2.9.1 Kontroler Proporsional (P).....  | 16                                  |
| 2.9.2 Kontroler Integral (I) .....   | 17                                  |
| 2.9.3 Kontroler Proporsional Integral (PI).....  | 18                                  |
| 2.9.4 Kontroler Proporsional Integral Diferensial (PID).....   | 18                                  |
| 2.10 Konsep Ruang Keadaan ( <i>State Space</i> ).....  | 19                                  |
| 2.11 Konsep Sistem Kontrol Optimal.....  | 20                                  |
| 2.12 <i>Pole Placement</i> .....   | 21                                  |
| 2.13 <i>State Observer</i> .....   | 23                                  |
| 2.14 <i>Output Feedback Control</i> .....  | 24                                  |
| 2.15 Desain Pra Filter.....  | 25                                  |
| <b>BAB III METODE PENELITIAN.....</b>  | <b>27</b>                           |
| 3.1 Perancangan Blok Diagram Sistem .....  | 27                                  |
| 3.2 Karakterisasi <i>Ford Water Pump 24V DC</i> .....  | 28                                  |
| 3.3 Karakterisasi Driver Motor (High Current 30 A DC Motor Driver dengan half<br>bridge infineon BTS 7960 chip)..... | 30                                  |
| 3.4 Karakterisasi Plant Tangki Terhubung.....  | 33                                  |
| 3.5 Algoritma Kontrol Optimal .....  | <b>Error! Bookmark not defined.</b> |
| 3.6 Perancangan Algoritma .....  | 34                                  |
| 3.7 Penentuan <i>State Space Plant</i> .....   | 35                                  |
| 3.8 Desain Kontrol Struktur Output Feedback Control.....   | 366                                 |
| 3.9 Diagram Blok Perancangan Desain .....  | 388                                 |
| <b>BAB IV HASIL DAN PEMBAHASAN.....</b>  | <b>40</b>                           |
| 4.1 Simulasi Penentuan Matriks R dan L dari Perubahan Pole Kontroler dan Perubahan<br>Pole Observer.....             | 39                                  |
| 4.2 Simulasi Perubahan Pole Observer terhadap perubahan nilai matriks L.....   | 41                                  |
| 4.2.1 Perubahan Nilai $op_1$ .....   | 41                                  |

|   |           |
|---|-----------|
| 4.2.2 Perubahan Nilai $op_2$ .....  | 42        |
| 4.3 Simulasi Perubahan Pole Kontroler terhadap perubahan nilai matriks R..... | 44        |
| 4.3.1 Perubahan Nilai $rp_1$ .....  | 44        |
| 4.3.2 Perubahan Nilai $rp_2$ .....  | 46        |
| 4.4 Simulasi dengan Gangguan.....   | 48        |
| 4.5 Simulasi Karakteristik Plant Sebelum dan Sesudah diberi Kontroler.....    | 48        |
| <b>BAB V</b> .....  | <b>50</b> |
| 5.1 Kesimpulan .....  | 50        |
| 5.2 Saran .....   | 50        |



## DAFTAR TABEL

|   |    |
|---|----|
| Tabel 2.1 Fungsi Pin <i>High Current 30 A DC Motor Driver</i> .....   | 8  |
| Tabel 3.1 Data Pengujian Kecepatan <i>Ford Water Pump 1</i> terhadap <i>Dutycycle</i> dan Tegangan Keluaran <i>Driver</i> ..... | 29 |
| Tabel 3.2 Data Pengujian <i>Driver</i> terhadap PWM Arduino, Tegangan Masukan, dan Tegangan Keluaran .....                      | 31 |
| Tabel 3.3 Hasil Pengujian Sensor Ultrasonik HC-SR04 .....   | 33 |
| Tabel 4.1 Nilai Matriks R, dan L terhadap Perubahan Nilai Observer Pole op1 .....   | 41 |
| Tabel 4.2 Nilai Matriks R, dan L terhadap Perubahan Nilai Observer Pole op2 .....   | 42 |
| Tabel 4.3 Nilai Matriks R, dan L terhadap Perubahan Nilai Kontroler Pole rp1 .....  | 44 |
| Tabel 4.4 Nilai matriks R, dan L terhadap Perubahan Nilai Kontroler Pole rp2 .....  | 45 |
| Tabel 4.5 Nilai Matriks R, L, M untuk Simulasi dengan Gangguan .....  | 47 |

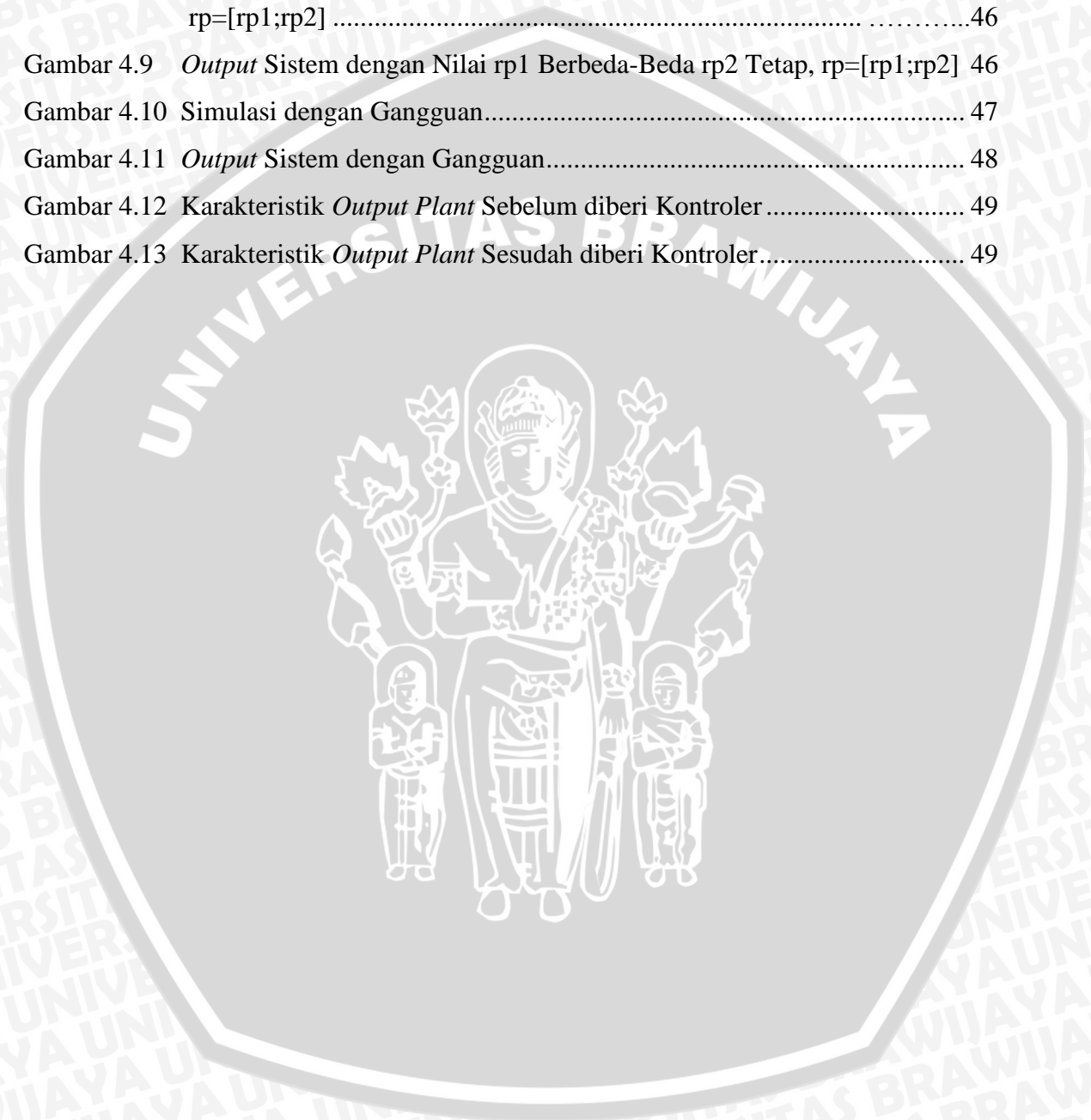




## DAFTAR GAMBAR

|             |   |    |
|-------------|---|----|
| Gambar 2.1  | Mikrokontroler Arduino Mega .....   | 4  |
| Gambar 2.2  | Diagram Proses Pengukuran .....   | 5  |
| Gambar 2.3  | Blok Diagram Sistem <i>Loop</i> Terbuka .....   | 5  |
| Gambar 2.4  | Blok Diagram <i>Close Loop</i> .....  | 6  |
| Gambar 2.5  | <i>Cascade Control System</i> .....   | 7  |
| Gambar 2.6  | <i>High Current 30 A DC Motor Driver</i> .....  | 7  |
| Gambar 2.7  | Rancangan Konstruksi <i>Plant</i> Tangki Terhubung.....   | 8  |
| Gambar 2.8  | Sistem Orde Satu.....   | 9  |
| Gambar 2.9  | Output Unit Step Sistem Orde Satu .....   | 10 |
| Gambar 2.10 | Sistem Ode Dua .....  | 10 |
| Gambar 2.11 | Output Unit Step Sistem Orde Dua.....   | 15 |
| Gambar 2.12 | Diagram Blok Sistem dengan Kontroler Otomatis .....   | 16 |
| Gambar 2.13 | Diagram Blok Kontroler Proporsional (P).....  | 17 |
| Gambar 2.14 | Diagram Blok Kontroler Integral (I).....  | 17 |
| Gambar 2.15 | Diagram Blok Kontroler Proporsional Integral (PI) .....   | 18 |
| Gambar 2.16 | Diagram Blok Kontroler Proporsional Integral Diferensial (PID) .....                                  | 19 |
| Gambar 2.17 | Representasi Sistem dalam Bentuk Persamaan State.....   | 20 |
| Gambar 2.18 | <i>Full Order State Observer</i> .....  | 23 |
| Gambar 2.19 | Struktur <i>Output Feedback Control</i> .....   | 25 |
| Gambar 2.20 | Desain Pra Filter .....   | 26 |
| Gambar 3.1  | Diagram Blok Sistem Pengendalian Ketinggian Air pada Tangki Terhubung<br>.....                        | 27 |
| Gambar 3.2  | Grafik Perubahan Debit Air (l/menit) terhadap <i>Dutycycle</i> .....                                  | 30 |
| Gambar 3.3  | Grafik Pengujian <i>Driver</i> terhadap PWM Arduino, Tegangan Masukan, dan<br>Tegangan Keluaran ..... | 32 |
| Gambar 3.4  | Grafik Hasil Pengujian Sensor Ultrasonik HC-SR04 .....  | 34 |
| Gambar 3.5  | Diagram Blok Perancangan Desain .....   | 38 |
| Gambar 4.1  | Simulasi Penggunaan Matriks R, L, dan M .....   | 40 |
| Gambar 4.2  | Grafik U dengan Nilai op1 Berbeda-Beda op2 Tetap, $op=[op1;op2]$ .....                                | 41 |
| Gambar 4.3  | <i>Output</i> Sistem dengan Nilai op1 Berbeda-Beda op2 Tetap, $op=[op1;op2]$                          | 42 |

|             |  |    |
|-------------|--|----|
| Gambar 4.4  | Grafik U dengan Nilai $op_2$ Berbeda-Beda $op_1$ Tetap, $op=[op_1;op_2]$ .....       | 43 |
| Gambar 4.5  | <i>Output</i> Sistem dengan Nilai $op_2$ Berbeda-Beda $op_1$ Tetap, $op=[op_1;op_2]$ | 43 |
| Gambar 4.6  | Grafik U dengan Nilai $rp_1$ Berbeda-Beda $rp_2$ Tetap, $rp=[rp_1;rp_2]$ .....       | 44 |
| Gambar 4.7  | <i>Output</i> Sistem dengan Nilai $rp_1$ Berbeda-Beda $rp_2$ Tetap, $rp=[rp_1;rp_2]$ | 45 |
| Gambar 4.8  | Grafik U dengan Nilai $rp_2$ Berbeda-Beda $rp_1$ Tetap, $rp=[rp_1;rp_2]$ .....       | 46 |
| Gambar 4.9  | <i>Output</i> Sistem dengan Nilai $rp_1$ Berbeda-Beda $rp_2$ Tetap, $rp=[rp_1;rp_2]$ | 46 |
| Gambar 4.10 | Simulasi dengan Gangguan.....  | 47 |
| Gambar 4.11 | <i>Output</i> Sistem dengan Gangguan.....  | 48 |
| Gambar 4.12 | Karakteristik <i>Output Plant</i> Sebelum diberi Kontroler .....                     | 49 |
| Gambar 4.13 | Karakteristik <i>Output Plant</i> Sesudah diberi Kontroler.....                      | 49 |



UNIVERSITAS BRAWIJAYA





# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Sistem kontrol adalah sebuah sistem yang terdiri atas beberapa peralatan yang berfungsi untuk mengendalikan sistem lain yang berhubungan dengan sebuah proses. Pada sistem kontrol dasar, telah dikenal kontroler proporsional, integral, dan diferensial yang dalam perkembangannya kontroler tersebut digabung menjadi kontroler PID (proporsional-integral-diferensial).

Dewasa ini, penggunaan dua tangki air untuk satu rumah ataupun gedung berskala besar bukanlah hal baru. Banyak rumah telah menerapkannya, terutama yang memiliki kondisi distribusi volume air PAM tidak teratur atau sangat kecil. Secara umum, mayoritas pemasangan dua tangki air bertujuan untuk menyediakan kapasitas cadangan lebih besar guna menampung air yang didistribusikan oleh PAM. Dengan demikian, volume air yang di distribusikan ke jaringan pipa dalam rumah memiliki konsistensi sama selama selang waktu tertentu.

Dalam proses industri banyak yang menggunakan tangki horizontal untuk menampung minyak atau cairan kimia. Pengaturan level digunakan untuk menjaga titik set level terhadap nilai yang diberikan sehingga mampu menerima nilai-nilai set point secara dinamis. Pada proses industri, fluida merupakan salah satu permasalahan yang terjadi pada saat pengaturan level coupled tank.

Sebenarnya, penggunaan dua tangki air tidak harus selalu untuk memenuhi kepentingan dari tujuan seperti itu saja. Bisa juga dimanfaatkan untuk menormalkan kekuatan tekanan air (baik yang terlalu lemah maupun terlalu kuat) atau sebagai model dasar untuk membuat fasilitas penyaringan air. Bahkan, bisa juga dimanfaatkan sekaligus untuk penanganan dari kedua kepentingan tersebut secara bersamaan.

*Pole placement* adalah suatu metode desain kontrol dimana menentukan letak dari kutub sistem *closed loop* pada bidang kompleks dengan mengatur *gain kontroler*. Penetapan kutub sistem dapat digunakan untuk mengoptimasi performansi sistem *closed loop* dengan meminimalisasi energi dengan *output* mencapai kondisi *steady state* secepat mungkin. Salah satu desain kontroler adalah *Output Feedback Control* (OFC). *OFC* disertai *observer/estimator* untuk mengestimasi *state-state* yang tidak dapat diukur



berdasarkan model *plant* dan *output* yang dapat diukur. Dengan OFC, *state-state* yang ada dapat diestimasi hanya dengan menggunakan *input* dan *output* sistem.

Dalam skripsi ini akan dibahas mengenai bagaimana cara mendesain dan menggunakan OFC pada *plant* tangki terhubung. Hal tersebut bertujuan agar ketinggian *output* tangki terhubung bernilai hampir konstan walaupun terdapat gangguan. Dengan kata lain, *error steady state* sekecil mungkin walau terdapat gangguan pada *plant* tangki terhubung. Selain itu, juga bertujuan untuk mendapatkan *settling time* secepat mungkin dengan energi seminimum mungkin.

## 1.2 Rumusan Masalah

1. Bagaimana cara menentukan *state space* sistem?
2. Bagaimana merancang algoritma pada pengontrolan ketinggian air pada tangki terhubung dengan struktur *output feedback control*?
3. Bagaimana hasil *ouput plant* tangki terhubung setelah menggunakan struktur *output feedback control* pada simulasi?
4. Bagaimana hasil *ouput plant* tangki terhubung yang menggunakan struktur *output feedback control* saat sistem diberi gangguan?

## 1.3 Batasan Masalah

1. Pada skripsi ini menggunakan mikrokontroler Arduino Mega 2560.
2. Pola gangguan yang diberikan bersifat deterministic yaitu berupa penurunan ketinggian air pada tangki saat dalam kondisi *steady state* akibat dari dibukanya lubang keluaran tangki 2.
3. Plant tangki terhubung yang digunakan bersifat *Single Input Single Output* (SISO) dimana input berupa debit air yang masuk ke tangki pertama melalui pompa dan *output* berupa ketinggian air pada tangki 2.
4. Pembahasan yang ditekankan pada cara mendesain dan penggunaan struktur *state feedback control*.
5. Penelitian ini hanya bersifat simulasi pada *software* MATLAB R2016b
6. Dimensi masing-masing tangki yang digunakan memiliki panjang 25cm, lebar 5cm, dan tinggi 17cm



#### 1.4 Tujuan

Tujuan skripsi ini adalah :

1. Menentukan *state space* sistem
2. Merancang algoritma pada pengontrolan ketinggian air pada tangki terhubung dengan struktur *output feedback control*
3. Mengetahui hasil *ouput plant* tangki terhubung setelah menggunakan struktur *output feedback control* pada simulasi
4. Mengetahui hasil *ouput plant* tangki terhubung yang menggunakan struktur *output feedback control* saat sistem diberi gangguan

#### 1.5 Manfaat

Manfaat skripsi ini adalah dapat dipahaminya sistem pengontrolan ketinggian air dengan menggunakan metode *output feedback control* pada sistem kontrol optimal.





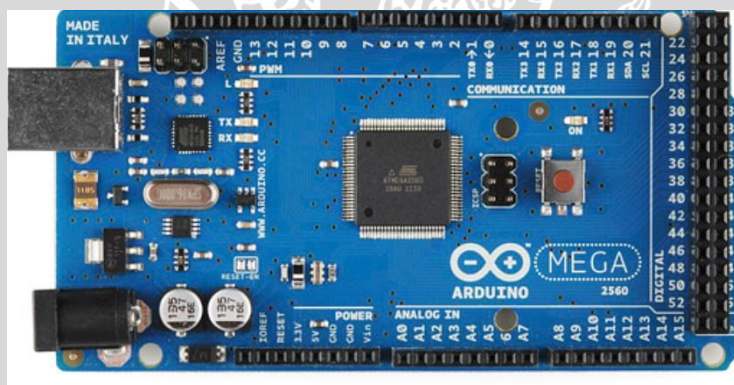
## BAB II

### TINJAUAN PUSTAKA

Bab ini menguraikan teori-teori pendukung skripsi, yang terdiri atas :

#### 2.1 Mikrokontroler Arduino Mega

Arduino adalah pengendali mikro single-board yang bersifat open-source, diturunkan dari Wiring platform, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Hardwarenya memiliki prosesor Atmel AVR dan softwarenya memiliki bahasa pemrograman sendiri. Saat ini Arduino sangat populer di seluruh dunia. Banyak pemula yang belajar mengenal robotika dan elektronika lewat Arduino karena mudah dipelajari. Tapi tidak hanya pemula, para hobbyist atau profesional pun ikut senang mengembangkan aplikasi elektronik menggunakan Arduino. Bahasa yang dipakai dalam Arduino bukan assembler yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (libraries) Arduino. Arduino juga menyederhanakan proses bekerja dengan mikrokontroler. Gambar Arduino Mega dapat dilihat dalam Gambar 2.1.



Gambar 2.1 Mikrokontroler Arduino Mega

Sumber: <http://www.robotshop.com/en/arduino-mega-microcontroller.html>

#### 2.2 Sistem Pengendalian

Sistem pengendalian berfungsi untuk mengendalikan jalannya proses agar variabel proses yang sedang diukur dapat dikendalikan dan diatur sesuai dengan nilai yang dikehendaki (*setpoint*). Gambar 2.2 memberikan kegiatan yang dilakukan dalam pengukuran, antara lain:



**Gambar 2.2** Diagram Proses Pengukuran

Sumber: K. Ogata, 2010

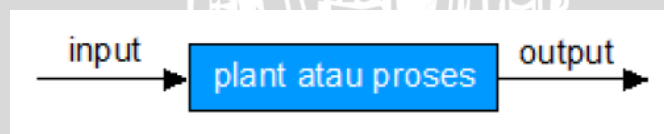
1. Mengukur yaitu mengukur proses variabel, kemudian sinyal diubah menjadi sinyal standar.
2. Membandingkan yaitu membandingkan antara *process variabel* (PV) dengan *setpoint value* (SV).
3. Menghitung yaitu menghitung perbedaan antara PV dengan SV, hasilnya disebut *error*.
4. Mengoreksi yaitu setelah mendapat nilai *error* maka dikoreksi untuk menentukan perubahan pada *final element*.

### 2.3 Loop Pengendalian

Secara umum bentuk *loop* sistem pengendalian dibagi menjadi dua macam, yaitu:

1. Sistem Pengendalian *Loop* Terbuka (*Open Loop Control System*)

Sistem pengendalian *loop* terbuka adalah sistem pengendalian yang keluarannya tidak dapat mempengaruhi aksi dari pengendaliannya. Jadi, pada sistem ini keluaran dari kontrolernya tidak diukur atau diumpanbalikkan untuk membandingkan dengan masukannya (K. Ogata, 2010), seperti ditampilkan dalam Gambar 2.3.



**Gambar 2.3** Blok Diagram Sistem *Loop* Terbuka

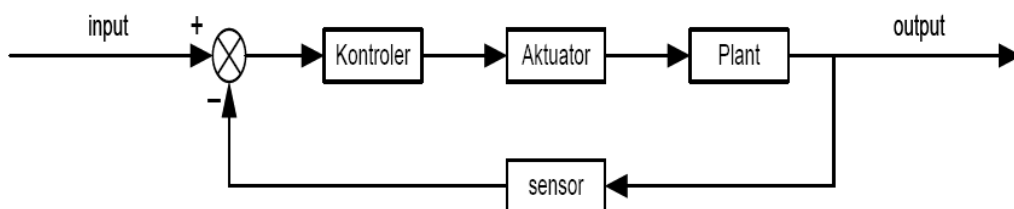
Sumber: K. Ogata, 2010

Salah satu contoh sistem *loop* terbuka adalah sistem pengaturan temperatur ruangan. Untuk mendapatkan temperatur yang diinginkan, operator menggunakan pengalamannya untuk mengeset daya yang dibutuhkan sistem agar keluaran sistem yang berupa temperatur ruangan sesuai dengan temperatur ruangan yang diinginkan.

2. Sistem Pengendalian *Loop* Tertutup (*Closed Loop Control System*)



Sistem pengendalian loop tertutup adalah sistem pengendalian yang sinyal keluarannya mempunyai pengaruh langsung pada aksi pengendaliannya (K. Ogata, 2010). Sinyal kesalahan yang bekerja yaitu antara sinyal masukan dan sinyal umpan balik yang disajikan ke kontroler disajikan sedemikian rupa untuk mengurangi kesalahan dan membawa keluaran sistem ke nilai yang dikehendaki. Diagram balok sistem *loop* tertutup ditampilkan dalam Gambar 2.4.



**Gambar 2.4** Block Diagram Close Loop

Sumber: K. Ogata, 2010

Contohnya *thermostat* pada ruangan untuk menjalankan atau mematikan alat pemanas atau pendingin agar suhu tetap nyaman. Biasanya berupa pengukur tidak langsung seperti pengukur *level* dengan radar dan ultrasonik. Di dalam pengendalian *loop* tertutup terdapat tiga macam metode pengendalian, yaitu:

a. *Feed Back Control System*

*Feed back control system* adalah sistem pengendalian dimana besaran proses yang diatur dan diukur (PV) dibandingkan dengan nilai yang dikehendaki (SV) dan perbedaannya digunakan sebagai dasar untuk mengeliminir perbedaan yang ada (membuka dan menutup *control valve*).

b. *Feed Forward Control System*

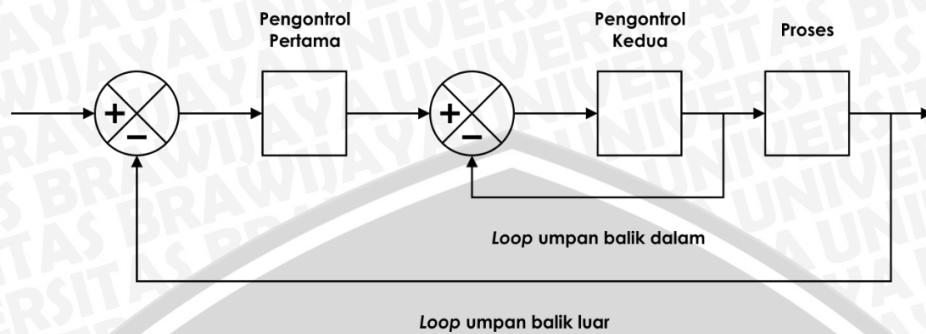
Pada metode ini beban proses pengaturan diukur kemudian dibandingkan dengan beban normal dan bila ada perbedaan, maka perbedaan tersebut digunakan sebagai dasar untuk melakukan aksi antisipasi agar tidak terjadi penyimpangan pada *primary* proses variabel yang diatur.

c. *Cascade Control System*

Kontrol *cascade* adalah kontrol yang melibatkan penggunaan dua buah pengontrol dengan keluaran dari pengontrol pertama merupakan titik pengaturan bagi pengontrol kedua. *Loop* umpan balik untuk salah satu pengontrol berada di dalam *loop* umpan balik untuk pengontrol yang lain, seperti yang ditampilkan dalam Gambar 2.5. Sistem seperti ini dapat



menghasilkan perbaikan karakteristik respon sistem terhadap gangguan yang muncul (W. Bolton, 2006: 290).



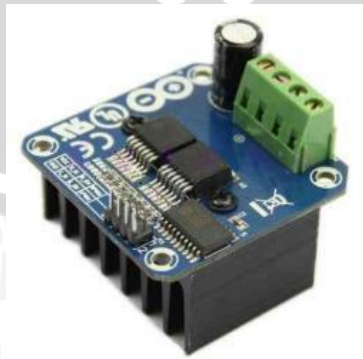
**Gambar 2.5** Cascade Control System

Sumber: K. Ogata, 2010

Kontrol kaskade meliputi penggabungan dua kontroler menjadi satu yang difungsikan secara bersamaan dan dikontrol secara bersama pula.

#### 2.4 High Current 30 A DC Motor Driver dengan half bridge infineon BTS 7960 chip

High Current 30 A DC Motor Driver (Gambar 2.6) merupakan driver H-Bridge yang didesain untuk menghasilkan drive 2 arah gerakan *clockwise* (cw) dan *counterclockwise* (ccw) dengan arus kontinyu sampai dengan 30 A pada tegangan 5 Volt sampai 27 Volt. Modul ini dilengkapi dengan rangkaian sensor arus beban yang dapat digunakan sebagai umpan balik ke pengontrol. Modul ini mampu men-drive (menggerakkan) motor DC yang menjadi penggerak generator. Rangkaian ini dihubungkan dengan Arduino Uno R3.



**Gambar 2.6** High Current 30 A DC Motor Driver

Sumber: datasheet High Current 30 A DC Motor Driver

Fungsi masing-masing pin pada *Interface Header* dapat dilihat dalam Tabel 2.1.

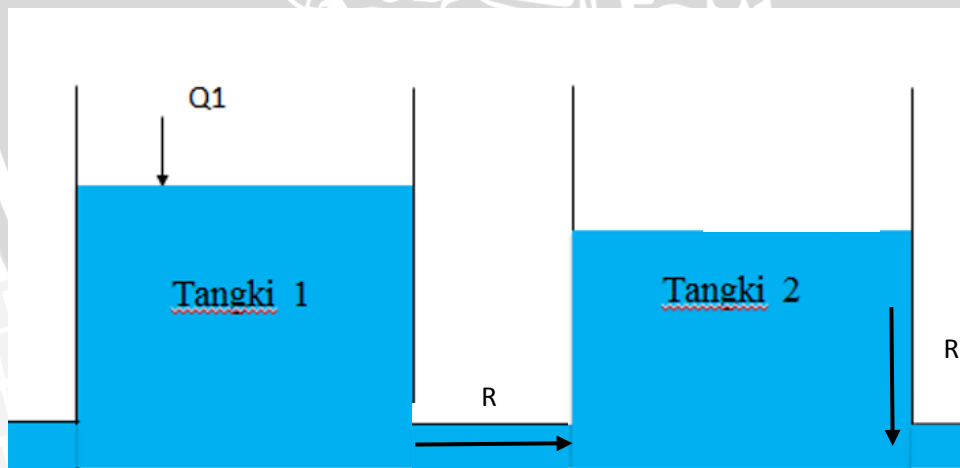
**Tabel 2. 1 Fungsi Pin *High Current 30 A DC Motor Driver***

| No | Pin  | Fungsi   |
|----|------|--|
| 1  | RPWM | Forward level atau input sinyal PWM, aktif <i>HIGH</i> |
| 2  | LPWM | Reverse level atau input sinyal PWM, aktif <i>LOW</i>  |
| 3  | R_EN | Forward drive enable input, HIGH enable, LOW close     |
| 4  | L_EN | Reverse drive enable input, HIGH enable, LOW close     |
| 5  | R_IS | Forward drive-side current alarm output                |
| 6  | L_IS | Reverse drive-side current alarm output                |
| 7  | VCC  | 5V power input dari board arduino                      |
| 8  | GND  | Ground   |

Sumber: *datasheet High Current 30 A DC Motor Driver*

## 2.5 Tangki Terhubung

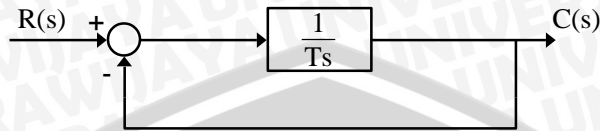
Tangki yang digunakan pada skripsi ini adalah tangki terhubung, dimana 2 tangki dengan ketinggian yang sama saling dihubungkan dengan valve yang membatasi 2 tangki tersebut. Pada tangki kedua, terdapat output untuk mengeluarkan air sebagai *disturbance* dalam sistem ini. Untuk rancangan konstruksinya bisa dilihat pada gambar 2.7



**Gambar 2.7** Rancangan konstruksi plant tangki terhubung

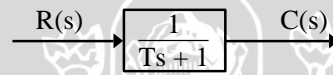
## 2.6 Output Sistem Orde Satu

Diagram blok sistem orde satu dapat dilihat dalam Gambar 2.8 yang mempunyai fungsi alih sebagai berikut



Gambar 2.8 Sistem orde satu

$$\begin{aligned} \frac{C(s)}{R(s)} &= \frac{1/Ts}{1 + 1/Ts} \quad (2-10) \\ &= \frac{1}{Ts + 1} \end{aligned}$$



Jika sistem orde satu diberi masukan unit step  $r(t) = 1$ , dari Transformasi Laplace fungsi ramp adalah  $R(s) = 1/s$ . Keluaran sistem adalah sebagai berikut

$$\begin{aligned} C(s) &= \frac{1}{Ts + 1} R(s) \\ C(s) &= \frac{1}{Ts + 1} \frac{1}{s} \\ &= \frac{1}{s(Ts + 1)} \quad (2-11) \end{aligned}$$

$$c(t) = \mathcal{L}^{-1}[C(s)]$$

$$c(t) = 1 - e^{-t/T} \quad (t \geq 0) \quad (2-12)$$

Persamaan keluaran tersebut menyatakan bahwa keluaran  $c(t)$  mula-mula nol kemudian akhirnya menjadi satu (dapat dilihat dalam Gambar 2.9). Salah satu karakteristik penting output eksponensial  $c(t)$  tersebut adalah bahwa pada  $t = T$ , maka

$$c(t) = 1 - e^{-1} \cong 0.632 = \frac{2}{3}$$

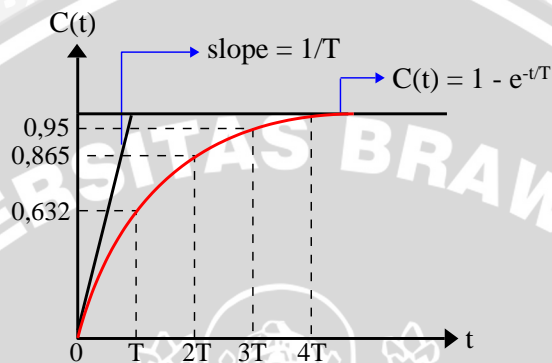
$T$  = *time constant* / konstanta waktu sistem

$$= \frac{2}{3} \text{ harga akhir}$$



Konstanta waktu  $T$  yang lebih kecil akan mempercepat output sistem. Karakteristik penting lainnya pada kurva output eksponensial adalah kemiringan garis singgung / gradien pada  $t = 0$  adalah  $1/T$ , karena

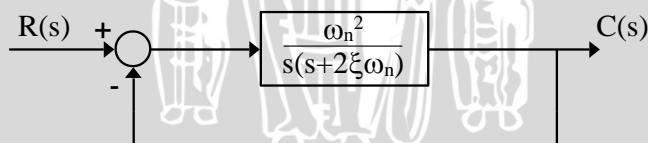
$$\frac{dc}{dt} = \frac{1}{T} e^{-t/T} \Big|_{t=0} = \frac{1}{T}$$



Gambar 2.9 Output unit step sistem orde satu

## 2.7 Output Sistem Orde Dua

Diagram blok sistem orde dua (lihat Gambar 2.10), dengan fungsi alihnya adalah sebagai berikut:



Gambar 2.10 Sistem orde dua

$$\begin{aligned} \frac{C(s)}{R(s)} &= \frac{\omega_n^2}{s(s+2\xi\omega_n)} \cdot \frac{1}{1 + \frac{\omega_n^2}{s(s+2\xi\omega_n)}} \\ &= \frac{\omega_n^2}{s(s+2\xi\omega_n) + \omega_n^2} \\ &= \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (2-13) \end{aligned}$$

Akar-akar penyebut fungsi alih atau persamaan karakteristik adalah

$$\begin{aligned}
 s_{1,2} &= \frac{-2\xi\omega_n \pm \sqrt{(2\xi\omega_n)^2 - 4\omega_n^2}}{2} \\
 s_{1,2} &= -\xi\omega_n \pm \omega_n\sqrt{\xi^2 - 1} \\
 &= -\xi\omega_n \pm j\omega_n\sqrt{1 - \xi^2} \\
 s_{1,2} &= -\xi\omega_n \pm j\omega_d \quad (2-14)
 \end{aligned}$$

dimana

- $\xi$  = rasio peredaman sistem (*damping ratio*)
- $\omega_n$  = frekuensi natural/alamiah tak teredam
- $\omega_d$  = frekuensi natural/alamiah teredam

Kelakuan dinamik sistem orde dua dapat digambarkan dalam suku dua parameter  $\xi$  dan  $\omega_n$ . Jika ( $0 < \xi < 1$ ), maka pole loop tertutup merupakan konjugat kompleks dan berada pada bidang  $s$  sebelah kiri. Dalam hal ini, sistem dikatakan dalam peredaman dan tanggapan peralihan berosilasi. Jika ( $\xi = 1$ ), maka sistem dikatakan teredam kritis. Sistem terlalu teredam berhubungan dengan ( $\xi > 1$ ). Tanggapan peralihan sistem teredam kritis dan sistem terlalu teredam tidak berosilasi. Jika  $\xi = 0$ , tanggapan peralihan tidak muncul.

Pada sistem orde dua seperti terlihat dalam Gambar 2.11, berdasarkan output sistem dengan masukan unit step akan terdapat tiga keadaan yang berbeda yaitu keadaan teredam ( $0 < \xi < 1$ ), teredam kritis ( $\xi = 1$ ), dan sistem terlalu teredam ( $\xi > 1$ ).

### 2.7.1 Keadaan Kurang Teredam / Underdamped ( $0 < \xi < 1$ )

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{(s^2 + 2\xi\omega_n s + \omega_n^2)s} \quad (2-15)$$

Jika sistem diberi input berupa unit step atau  $R(s) = \frac{1}{s}$ , maka:

$$C(s) = \frac{\omega_n^2}{(s^2 + 2\xi\omega_n s + \omega_n^2)s} \quad (2-16)$$

Dari Tabel Transformasi Laplace didapatkan

$$c(t) = 1 - \frac{1}{\sqrt{1-\xi^2}} e^{-\xi\omega_n t} \sin(\omega_n \sqrt{1-\xi^2} t + \phi)$$

$$\phi = \arctan \frac{\sqrt{1-\xi^2}}{\xi}$$

Jika  $\omega_d = \omega_n \sqrt{1-\xi^2}$ ; maka

$$c(t) = 1 - \frac{e^{-\xi\omega_n t}}{\sqrt{1-\xi^2}} \sin\left(\omega_d t + \arctan \frac{\sqrt{1-\xi^2}}{\xi}\right) \quad t \geq 0 \quad (2-17)$$

Output sistem tersebut juga bisa diperoleh dengan menggunakan Transformasi Laplace balik jika  $C(s)$  ditulis dalam bentuk berikut:

$$C(s) = \frac{1}{s} - \frac{s + 2\xi\omega_n}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (2-18)$$

$$= \frac{1}{s} - \frac{s + \xi\omega_n}{(s + \xi\omega_n)^2 + \omega_d^2} - \frac{\xi\omega_n}{(s + \xi\omega_n)^2 + \omega_d^2}$$

$$\mathcal{L}^{-1}\left[\frac{s + \xi\omega_n}{(s + \xi\omega_n)^2 + \omega_d^2}\right] = e^{-\xi\omega_n t} \cos \omega_d t$$

$$\mathcal{L}^{-1}\left[\frac{\xi\omega_n}{(s + \xi\omega_n)^2 + \omega_d^2}\right] = e^{-\xi\omega_n t} \sin \omega_d t \quad (2-19)$$

oleh karena itu, transformasi laplace balik dari persamaan

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

diperoleh sebagai

$$\mathcal{L}^{-1}[C(s)] = c(t)$$

$$c(t) = 1 - e^{-\xi\omega_n t} \left( \cos \omega_d t + \frac{\xi}{\sqrt{1-\xi^2}} \sin \omega_d t \right)$$

$$c(t) = 1 - \frac{e^{-\xi\omega_n t}}{\sqrt{1-\xi^2}} \sin\left(\omega_d t + \arctan \frac{\sqrt{1-\xi^2}}{\xi}\right) \quad t \geq 0 \quad (2-20)$$

Sinyal kesalahan / error adalah  $e(t) = r(t) - c(t)$ , dimana

$$r(t) = 1$$

dan



$$c(t) = 1 - e^{-\xi\omega_n t} \left( \cos \omega_d t + \frac{\xi}{\sqrt{1-\xi^2}} \sin \omega_d t \right)$$

sehingga

$$e(t) = e^{-\xi\omega_n t} \left( \cos \omega_d t + \frac{\xi}{\sqrt{1-\xi^2}} \sin \omega_d t \right) \quad t \geq 0$$

$$\text{jika } \xi = 0 \Rightarrow c(t) = 1 - \cos \omega_n t$$

### 2.7.2 Teredam Kritis / Critically Damped ( $\xi = 1$ )

Dalam hal ini apabila dua pole  $C(s)/R(s)$  hampir sama, maka sistem dapat didekati dengan bentuk teredam kritis. Jika input berupa unit step dimana  $R(s) = 1/s$  maka:

$$C(s) = \frac{\omega_n^2}{(s + \omega_n)^2 s} \quad (2-21)$$

$$c(t) = 1 - e^{-\omega_n t} (1 + \omega_n t) \quad t \geq 0$$

### 2.7.3 Terlalu Teredam / Overdamped ( $\xi > 1$ )

Dalam hal ini pole  $C(s)/R(s)$  adalah bilangan nyata / real negatif yang tidak sama.

Jika input berupa unit step dimana  $R(s) = 1/s$  dan  $C(s)$  dapat ditulis dengan :

$$C(s) = \frac{\omega_n^2}{(s + \omega_n)^2 s} \quad (2-22)$$

$$C(s) = \frac{\omega_n^2}{(s + \xi\omega_n + \omega_n\sqrt{\xi^2 - 1})(s + \xi\omega_n - \omega_n\sqrt{\xi^2 - 1})s}$$

$$c(t) = 1 + \frac{\omega_n}{2\sqrt{\xi^2 - 1}} \left( \frac{e^{-s_1 t}}{s_1} - \frac{e^{-s_2 t}}{s_2} \right) \quad t \geq 0$$

$$\text{dengan } s_1 = (\xi + \sqrt{\xi^2 - 1})\omega_n$$

$$s_2 = (\xi - \sqrt{\xi^2 - 1})\omega_n \quad (2-23)$$

Tanggapan  $c(t)$  terdiri dari dua suku eksponensial menurun.

## 2.8 Tanggapan Peralihan

Sistem dengan tenaga tidak dapat memberikan tanggapan seketika dan akan menunjukkan tanggapan peralihan walaupun diberi masukan ataupun gangguan.

Karakteristik unjuk kerja sistem kontrol yang diinginkan dicirikan oleh suku tanggapan peralihan terhadap masukan unit step karena hal itu mudah dilakukan dan cukup drastis. Jika tanggapan terhadap masukan unit step diketahui, secara matematis dapat dihitung tanggapan untuk masukan yang lain.

Tanggapan peralihan sistem kontrol selalu menunjukkan osilasi teredam sebelum mencapai keadaan mantapnya, hal ini juga menunjukkan bahwa sistem tersebut mempunyai rasio peredaman ( $0 < \xi < 1$ ) yang juga berarti bahwa sistem tersebut merupakan sistem yang kurang teredam / underdamped.

Tanggapan peralihan sistem kontrol terhadap masukan unit step umumnya dikelompokkan sebagai berikut (lihat Gambar 2.12):

1) Delay Time / Waktu Tunda,  $t_d$

Waktu yang dibutuhkan oleh outputs untuk mencapai setengah harga akhir pada saat lonjakan pertama

2) Rise Time / Waktu Naik,  $t_r$

Waktu yang dibutuhkan oleh outputs agar bertambah dari 10% menjadi 90% dari nilai akhir

3) Peak Time / Waktu Puncak,  $t_p$

Waktu yang dibutuhkan oleh outputs untuk mencapai puncak pertama lonjakan (maksimum)

4) Maximum Overshoot / Lonjakan Maksimum,  $M_p$

Merupakan nilai puncak kurva outputs diukur dari satu

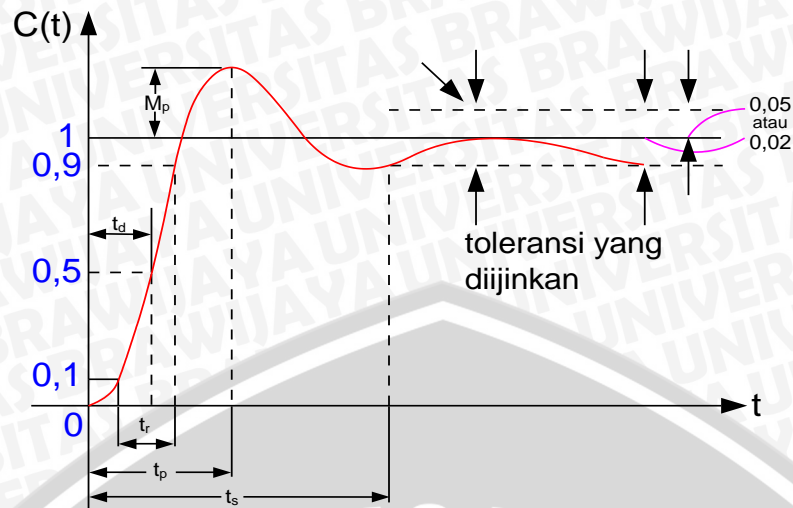
$$M_p = \frac{c(t_p) - c(\infty)}{c(\infty)} \times 100\%$$

dengan  $c(t_p)$  = nilai *output* pada saat lonjakan maksimum.

$c(\infty)$  = nilai *output* pada saat keadaan mantap.

5) Settling Time / Waktu Turun,  $t_s$

Waktu yang dibutuhkan oleh *output* untuk mencapai harga tertentu dan tetap dalam range nilai akhir (biasanya 5% atau 2%) dapat dilihat dalam Gambar 2.11.



Gambar 2.11 Output Unit Step Sistem Orde Dua

## 2.9 Kontroler

Kontroler seringkali juga disebut dengan istilah kompensator atau pengontrol. Kontroler adalah suatu sistem dinamis yang sengaja ditambahkan untuk mendapatkan karakteristik sistem keseluruhan yang diinginkan (Ogata K., 2010). Fungsi kontroler pada umumnya adalah sebagai berikut:

- 1) Membandingkan nilai *input* dan *output* sistem secara keseluruhan (*plant*).
- 2) Menentukan penyimpangan (*error*).
- 3) Menghasilkan sinyal kontrol (mengurangi penyimpangan (*error*) menjadi nilai nol/ nilai yang kecil).

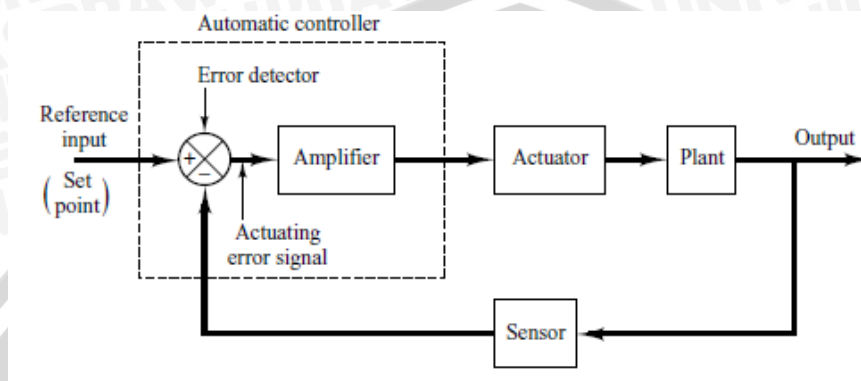
Adapun tujuan kontrol secara khusus adalah sebagai berikut:

- 1) Meminimumkan error steady state.
- 2) Meminimumkan *settling time*.
- 3) Mencapai spesifikasi transien yang lain, misalnya meminimumkan *maximum overshoot*.

Sistem loop tertutup (Gambar 2.12) menggunakan sinyal *output* yang diumpanbalikkan terhadap *automatic controller* (kontroler otomatis) (Ogata, K., 2010), yang akan membuat perubahan terhadap sistem agar *output* sistem seperti yang diinginkan atau sesuai *set point*. Sensor/transduser digunakan sebagai elemen yang langsung mengadakan kontak dengan objek yang diukur. Transduser berfungsi untuk mengubah besaran fisis yang diukur menjadi besaran fisis lainnya, seperti mengubah besaran tekanan, temperatur, aliran, posisi menjadi besaran listrik. *Actuating error signal* merupakan sinyal kesalahan (*error*) yang merupakan selisih antara sinyal set point dan



sinyal output. *Actuator* (aktuator) berfungsi untuk mengontrol aliran energi ke sistem yang dikontrol. Sebagai contoh adalah motor listrik, katub pengontrol, pompa dan sebagainya. *Amplifier* merupakan unit yang dibutuhkan karena daya dari *error detector* tidak cukup kuat untuk menggerakkan elemen *output*. Karena fungsi pengontrolan adalah untuk mengendalikan *output* agar kesalahan (*error*) mendekati nol, maka diperlukan penguat daya (*power amplifier*).



Gambar 2.12 Diagram Blok Sistem dengan Kontroler Otomatis

Sumber: Ogata, K. (2010)

Cara bagaimana kontroler otomatis menghasilkan sinyal kontrol disebut dengan aksi kontrol. Aksi kontrol dasar yang sering digunakan dalam kontroler adalah

1. Kontroler proporsional (P)
2. Kontroler integral (I)
3. Kontroler proporsional integral (PI)
4. Kontroler proporsional integral diferensial (PID)

### 2.9.1 Kontroler Proporsional (P)

Kontroler proporsional adalah sebuah kontroler yang memiliki karakteristik mempercepat output. Hubungan antara output kontroler  $u(t)$  dan sinyal *error*  $e(t)$  ditunjukkan dalam persamaan berikut:

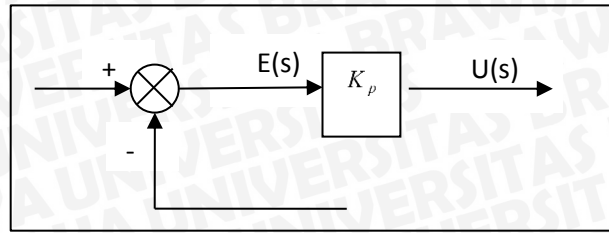
$$u(t) = K_p e(t) \quad (2-24)$$

atau, dalam fungsi alih

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

dimana  $K_p$  adalah penguatan.

Diagram blok kontroler proporsional (P) ditunjukkan dalam Gambar 2.13.



Gambar 2.13 Diagram blok kontroler proporsional (P)

Sumber: Ogata, K. (2010)

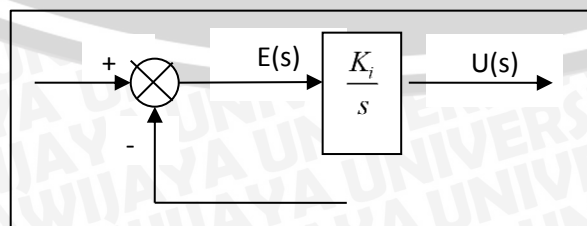
Apapun wujud mekanisme yang sebenarnya dan apapun bentuk daya penggerakannya, kontroler proporsional pada dasarnya merupakan penguat dengan penguatan yang dapat diatur (Ogata K., 2010).

### 2.9.2 Kontroler Integral (I)

Kontroler integral (I) memiliki kemampuan untuk mengurangi *offset* yang diakibatkan oleh kontroler proporsional. Output kontroler  $u(t)$  diubah dengan laju yang sebanding dengan *error*  $e(t)$ . Persamaan kontroler integral (I) ditunjukkan dalam persamaan berikut (Ogata K., 2010).

$$\begin{aligned} \frac{du(t)}{dt} &= K_i e(t) \\ u(t) &= K_i \int_0^t e(t) dt \\ \frac{U(s)}{E(s)} &= \frac{K_i}{s} \end{aligned} \quad (2-25)$$

yang merupakan fungsi alih kontroler integral (I), dengan  $K_i$  adalah konstanta integral yang dapat diubah nilainya. Jika  $e(t)$  bernilai nol, maka nilai  $m(t)$  tetap konstan. Aksi kontrol integral biasa disebut dengan kontrol reset. Gambar 2.14 menunjukkan diagram blok kontroler integral (I).



Gambar 2.14 Diagram blok kontroler integral (I)

Sumber: Ogata, K. (2010)

### 2.9.3 Kontroler Proporsional Integral (PI)

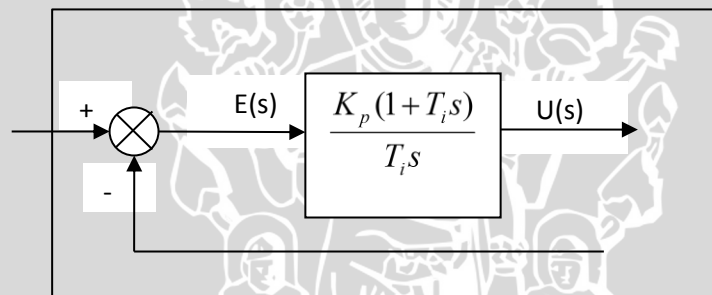
Kontroler proporsional integral (PI) memiliki kemampuan untuk mempercepat output dan mengurangi *offset*. Persamaan kontroler proporsional integral (PI) adalah

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (2-26)$$

Adapun fungsi alihnya adalah

$$\begin{aligned} \frac{U(s)}{E(s)} &= K_p \left( 1 + \frac{1}{T_i s} \right) \\ &= \frac{K_p (1 + T_i s)}{T_i s} \end{aligned}$$

dengan  $K_p$  penguatan proporsional dan  $T_i$  disebut waktu integral, yang keduanya dapat ditentukan. Waktu integral mengatur aksi kontrol integral sedangkan perubahan nilai  $K_p$  berakibat pada bagian aksi kontrol proporsional maupun integral. Gambar 2.15 menunjukkan diagram blok kontroler proporsional integral (PI).



Gambar 2.15 Diagram blok kontroler proporsional integral (PI)

Sumber: Ogata, K. (2010)

### 2.9.4 Kontroler Proporsional Integral Diferensial (PID)

Gabungan aksi kontrol proporsional, integral, dan differensial mempunyai keunggulan dibandingkan dengan masing-masing dari tiga aksi kontrol tersebut. Masing-masing kontroler P, I, maupun D berfungsi untuk mempercepat output sistem, menghilangkan *offset*, dan mendapatkan energi ekstra ketika terjadi perubahan beban (mempercepat *settling time*). Aksi kontroler proporsional integral diferensial (PID) didefinisikan dengan persamaan

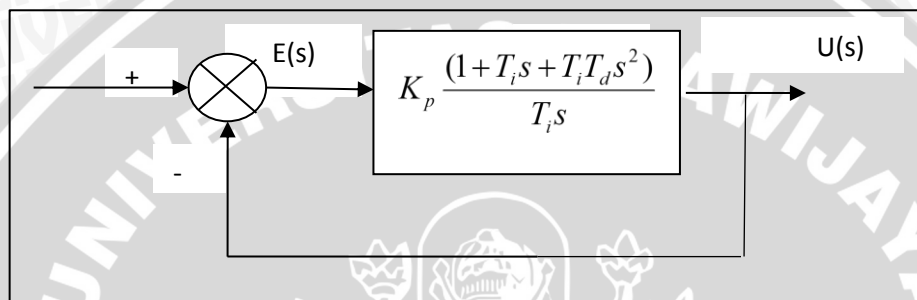
$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (2-27)$$



Sedangkan fungsi alihnya adalah

$$\begin{aligned}\frac{U(s)}{E(s)} &= K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) \\ &= K_p \frac{(1 + T_i s + T_i T_d s^2)}{T_i s}\end{aligned}$$

dimana  $K_p$  adalah penguatan proporsional,  $T_i$  adalah waktu integral dan  $T_d$  adalah waktu diferensial. Gambar 2.16 menunjukkan diagram blok kontroler proporsional integral diferensial (PID).



Gambar 2.16 Diagram blok kontroler proporsional integral diferensial (PID)

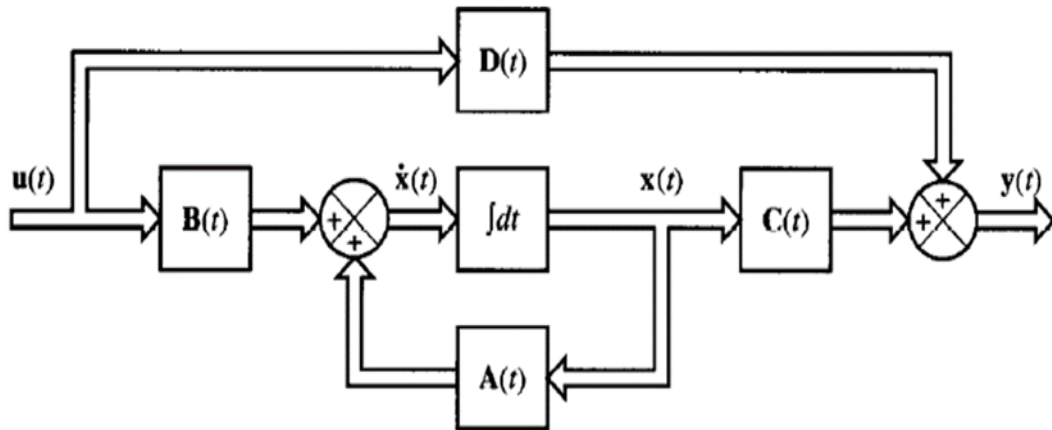
Sumber: Ogata, K. (2010)

Kombinasi ini mempunyai keuntungan dibandingkan masing-masing kontroler, biasanya dengan kontroler ini didapatkan overshoot yang rendah, cepat mencapai *steady state* (keadaan mantap) dan *error steady state* (kesalahan keadaan mantap) yang kecil bahkan nol. Pada perancangan sistem kontrol, seringkali kita harus melakukan *trial and error*. Contohnya pada desain PID, kita harus melakukan *trial and error* kepada 3 variabel yang berbeda (P, I, dan D). Desain PID bisa juga dilakukan dengan menggunakan metode Ziegler-Nichols.

## 2.10 Konsep Ruang Keadaan (*State Space*)

Proses pemodelan sistem akan selalu dilakukan dalam mendesain sistem kontrol. Model dapat berupa diagram blok, diagram aliran sinyal, maupun dalam bentuk persamaan matematis. Model matematis sebuah sistem didefinisikan sebagai kumpulan informasi dalam bentuk persamaan matematis yang mewakili sistem fisik yang ditinjau. Terdapat suatu metode untuk memodelkan sistem dengan multi *input* atau *multi output*, yang dikenal dengan metode ruang keadaan (*state space*). Model ruang keadaan merupakan sekumpulan persamaan diferensial orde satu yang berhubungan dan ditulis

dalam notasi matriks vektor. Bentuk umum ruang keadaan (*state space*) sebuah sistem dapat dilihat dari representasi sistem dalam bentuk persamaan *state* dalam Gambar 2.17.



Gambar 2.17 Representasi sistem dalam bentuk persamaan state

Sumber: Ogata, K. (2010)

Persamaan *state*:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (2-28)$$

Persamaan *output*:

$$y(t) = Cx(t) + Du(t) \quad (2-29)$$

dengan:

$x(t)$  = vektor state (n-vektor)

$u(t)$  = vektor kontrol (m-vektor)

$y(t)$  = vektor *output* (p-vektor)

$A$  = matriks keadaan orde  $n \times n$

$B$  = matriks *input* orde  $n \times m$

$C$  = matriks *output* orde  $p \times n$

$D$  = matriks transmisi langsung orde  $p \times n$

## 2.11 Konsep Sistem Kontrol Optimal

Sistem kontrol optimal adalah sistem kontrol yang mempunyai unjuk kerja terbaik (*best performance*) terhadap suatu acuan tertentu. Sistem kontrol mempunyai tolok ukur, artinya sistem tersebut sesuai terhadap suatu referensi (Houpis D'Azzo, 1995). Supaya suatu sistem dapat dikontrol, maka perlu dibuat model matematis yang menghubungkan *input* dan *output*. Pada sistem kontrol optimal, model yang banyak digunakan adalah persamaan keadaan (*state space*).



Cara untuk mengukur kualitas output transien pada sistem kontrol yang diberi *input* step yaitu dengan *control area*. Performansi sistem terbaik ditandai harga *output*  $y(t)$  mendekati *set point (input)*  $u(t)$  sehingga luasan *error*  $e(t) = u(t) - y(t)$  mendekati nol. Selain itu, waktu yang dibutuhkan oleh *output* untuk mencapai kondisi *steady state* secepat mungkin.

Cara paling sederhana untuk mencapai kondisi “*steady state*” secepat mungkin adalah dengan memperbesar parameter kontroler proporsional (P). Sedangkan untuk mengurangi *error* adalah dengan memperbesar parameter kontroler integral (I). Namun kedua cara tersebut akan dapat mengakibatkan ketidakstabilan sistem dan mengurangi nilai redaman ( $\xi$ ). Tujuan tersebut tidak dapat dicapai secara bersama-sama. Oleh karena itu diperlukan konsep kompromi atau pengambilan “jalan tengah” dari kedua tujuan itu. “Jalan Tengah” itu dikenal sebagai optimasi. Syarat proses optimasi adalah penetapan fungsi tujuan. Sinyal yang mungkin dapat digunakan untuk penggambaran itu adalah sinyal *error* (Rusli, M., 2015).

### 2.12 Pole Placement

*Pole placement* adalah suatu metode desain kontrol dimana menentukan letak dari *pole* sistem loop tertutup pada bidang kompleks dengan mengatur *gain kontroler*. Pada pendekatan secara konvensional, kita mengasumsikan bahwa pengaruh respon *pole* loop tertutup tak dominan dapat diabaikan. Pada pendekatan dengan penempatan *pole (pole placement)*, pendekatan penempatan *pole* yang diberikan menetapkan semua loop tertutup. Terdapat persyaratan pada bagian sistem untuk *pole-pole* loop tertutup yang akan ditempatkan pada lokasi yang dipilih secara sembarang yaitu bahwa sistem harus berupa keadaan lengkap yang dapat dikontrol.

Keterkontrolan keadaan secara sempurna dapat dilihat dari sistem waktu diskrit. Sistem diskrit yang dinyatakan oleh “  $x((k + 1)T) = Gx(kT) + Hu(kT)$  “

Dimana :

$x(kT)$  = vektor keadaan (vektor n dimensi)

$u(kT)$  = sinyal kontrol

G = matriks nonsingular n x n

H = matriks n x 1

T = perioda cacah

Kondisi keterkontrolan sempurna diturunkan berdasarkan kenyataan bahwa jika suatu sistem terkontrol sempurna, maka ada sinyal kontrol kontinu sepotong-sepotong



yang akan memindahkan setiap keadaan awal ke titik asal dalam sejumlah perioda cacah yang terhingga.

*Observability* atau keteramatan sistem linier dapat dinyatakan oleh persamaan berikut:

$$\begin{aligned}\dot{x}(t) &= Ax(t) \\ y &= Cx(t)\end{aligned}$$

dimana :

$x$  = vektor keadaan (vektor n-dimensi)

$y$  = vektor keluaran (vektor m-dimensi)

$A$  = matriks  $n \times n$

$C$  = matriks  $m \times n$

Keteramatan sempurna dari sistem waktu diskrit. Tinjau sistem yang dinyatakan oleh :

$$\begin{aligned}x((k+1)T) &= Gx(kT) \\ y(kT) &= Cx(kT)\end{aligned}$$

dimana:

$x(kT)$  = vektor keadaan (vektor n-dimensi)

$y(kT)$  = vektor keluaran (vektor m-dimensi)

$G$  = matriks  $n \times n$

$C$  = matriks  $m \times n$

$T$  = periode cacah

Persamaan ruang keadaan waktu berbentuk :

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (2-30)$$

Dan sinyal kontrol diberikan oleh :

$$u = -kx \quad (2-31)$$

Dan substitusi persamaan (2-31) ke persamaan (2-30) sehingga didapatkan

$$\dot{x}(t) = (A - BK)x(t) \quad (2-32)$$

Solusi dari persamaan (2-32)

$$x(t) = e^{(A-BK)t}x(0)$$

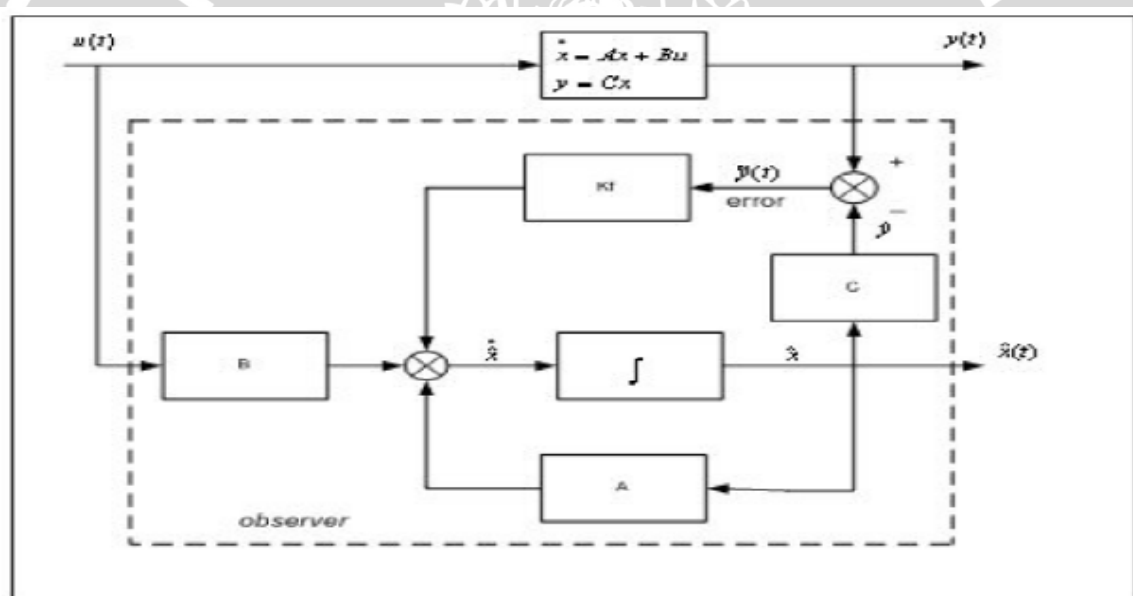
Kestabilan dari sistem (2-32) ditentukan oleh nilai eigen dari  $(A-BK)$ , artinya jika matriks  $K$  dapat dipilih secara tepat, maka bagian riil dari nilai eigen matriks  $(A-BK)$  terletak di sebelah kiri sumbu imajiner bidang  $s$ , hal itu berarti, untuk semua  $x(0) \neq 0$ ,  $x(t) \rightarrow 0$  untuk  $t \rightarrow \infty$ . Syarat perlu dan cukup untuk penempatan pole dinyatakan dalam teorema berikut. :

Teorema [Ogata Katsuhiko, 1997]: “Jika diberikan suatu sistem, maka syarat perlu dan cukup untuk penempatan sembarang pole yang diinginkan adalah bahwa sistem tersebut terkontrol secara lengkap”.

### 2.13 State Observer

Pada sistem dengan banyak state, adalah tidak mungkin untuk mengetahui seluruh informasi mengenai state. Tidak semua state dapat diindera oleh sensor, dan dapat diumpanbalikkan. Sistem dengan *state feedback* membutuhkan sensor ideal yang mempunyai *bandwidth* tak terhingga. Kenyataannya tidak ada sensor yang ideal dan juga sensor memiliki *bandwidth* yang terbatas.

Konsep pengamat (*observer*) sebenarnya adalah bila tidak memiliki seluruh state, adalah mungkin untuk mengestimasi state-state tersebut dengan menggunakan *input* dan *output* sistem. Konsepnya adalah seperti dalam Gambar 2.18.



Gambar 2.18 Full order state observer

Sumber: Gopal, M. (1987)

Untuk sistem,

$$\dot{x}(t) = Ax(t) + Bu(t); x(0) = x^0 \quad (2-33)$$

$$y = Cx \quad (2-34)$$

Dengan estimasi state  $\hat{x}(t)$  dinotasikan oleh

$$\dot{\hat{x}}(t) = A\hat{x} + Bu + K_f(y - C\hat{x}) \quad (2-35)$$

$$\hat{y} = \hat{C}\hat{x} \quad (2-36)$$

Nilai  $\tilde{y} = y - \hat{y}$  diusahakan bernilai sekecil mungkin, yang menandakan bahwa estimator bekerja dengan baik. Untuk membuat  $\hat{x}(t)$  sedekat mungkin dengan  $x(t)$ , didefinisikan *state estimator error*:

$$\begin{aligned}\tilde{x} &= x - \hat{x} \\ \dot{\tilde{x}} &= \dot{x} - \dot{\hat{x}} \\ \dot{\tilde{x}} &= Ax + Bu - A\hat{x} - B\dot{\hat{x}} - K_f C(x - \hat{x}) \\ \dot{\tilde{x}} &= A(x - \hat{x}) - K_f C(x - \hat{x}) \\ \dot{\tilde{x}} &= (A - K_f C) \tilde{x}\end{aligned}\quad (2-36)$$

Jika dapat dicari nilai matriks *observer gain*  $K_f$  sedemikian rupa sehingga  $(A - K_f C)$  stabil, maka nilai *estimator error* akan menuju nol untuk sembarang nilai awal. Hal ini dapat dilakukan hanya jika pasangan matriks  $(A, C)$  *observable* (Lewis and Syrmos, 1995:450).

Dengan:

$$K_f^T = R^{-1} C P$$

Atau

$$K_f^T = P C^T R^{-1}$$

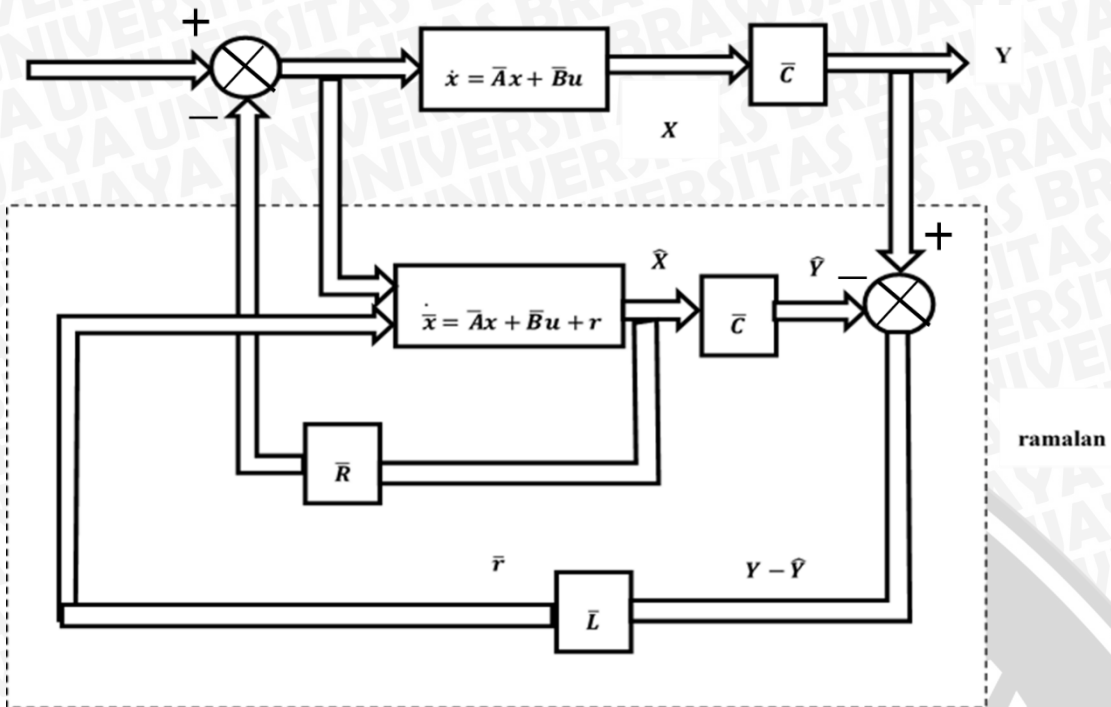
Dimana P adalah solusi dari persamaan Riccati:

$$AP + PA^T + Q - PC^T R^{-1} CP = 0 \quad (2-37)$$

## 2.14 Output Feedback Control

Salah satu desain kontroler adalah *Output Feedback Control* (OFC). Dengan menggunakan *OFC*, dapat diestimasi *state-state* yang ada hanya dengan menggunakan *input* dan *output* sistem. Salah satu metode yang digunakan pada *OFC* adalah dengan menggunakan estimasi Leunberger. Estimasi tersebut dapat dilihat dalam Gambar 2.19.





Gambar 2.19 Struktur Output Feedback Control

Sumber: Kwakernaak (2011)

$$\dot{\hat{x}} = \bar{A}\hat{x} + \bar{B}u + r$$

$$\bar{r} = \bar{L}(Y - \hat{Y})$$

$$\hat{Y} = \bar{C}\hat{x}$$

$$Y = \bar{C}x$$

distubtitusi:

$$\dot{\hat{x}} = \bar{A}\hat{x} + \bar{B}u + \bar{L}(Y - \hat{Y})$$

$$\dot{\hat{x}} = \bar{A}\hat{x} + \bar{B}u + \bar{L}(\bar{C}x - \bar{C}\hat{x})$$

$$\dot{\hat{x}} = \bar{A}\hat{x} + \bar{B}u + \bar{L}(x - \hat{x})$$

$(x - \hat{x}) = \tilde{x}$  dinamakan selisih error

$$\dot{\tilde{x}} = (\bar{A} - \bar{L}\bar{C})\tilde{x}$$

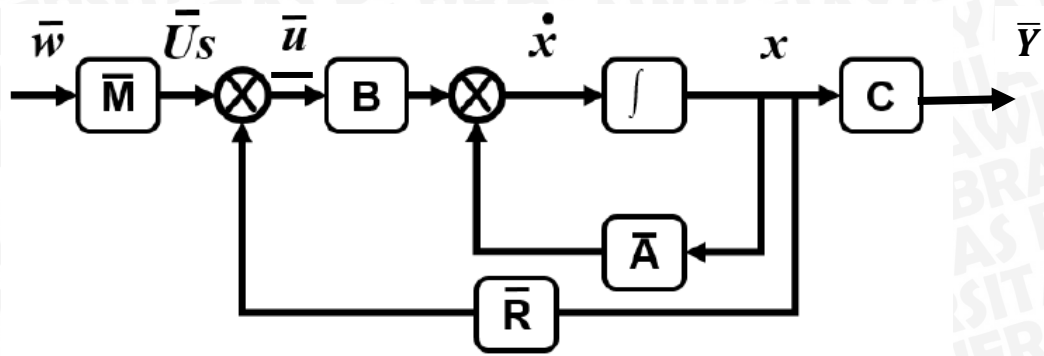
untuk mencari matriks L menggunakan rumus:

$$L = (-S^{-1}CP)^T \quad (2-38)$$

## 2.15 Desain Pra Filter

Pra filter digunakan untuk menunjang performa *output feedback control* (OFC).

Desain pra filter ditunjukkan dalam Gambar 2.20.



Gambar 2.20 Desain pra filter

Dimana:

$$\dot{x} = \bar{A}x + \bar{B}u$$

$$\bar{Y} = Cx$$

$$\bar{u} = -\bar{R}x + \bar{U}s$$

$$\bar{U}s = \bar{M}w$$

disubstitusi:

$$\dot{x} = \bar{A}x + \bar{B}(-\bar{R}x + \bar{U}s)$$

$$\dot{x} = (\bar{A} - \bar{B}\bar{R})x + \bar{B}\bar{U}s$$

ditransformasi Laplace:

$$s\bar{X}(s) = (\bar{A} - \bar{B}\bar{R})\bar{X}(s) + \bar{B}\bar{U}s(s)$$

$$\bar{X}(s) = [sI - (\bar{A} - \bar{B}\bar{R})]^{-1}\bar{B}\bar{U}s(s)$$

$$\bar{Y}(s) = C\bar{X}(s)$$

$$\bar{Y}(s) = C[sI - (\bar{A} - \bar{B}\bar{R})]^{-1}\bar{B}\bar{U}s(s)$$

$$\bar{M} = [-C(\bar{A} - \bar{B}\bar{R})^{-1}\bar{B}]^{-1} \quad (2-39)$$

## BAB III

### METODE PENELITIAN

Metodologi penelitian disini difungsikan demi tercapainya tujuan skripsi yang terdapat di bab pendahuluan. Metode penelitian pada skripsi ini meliputi :

1. Perancangan Diagram Blok Sistem
2. Karakterisasi setiap blok

Karakterisasi setiap blok dilakukan untuk mempermudah analisis sistem.

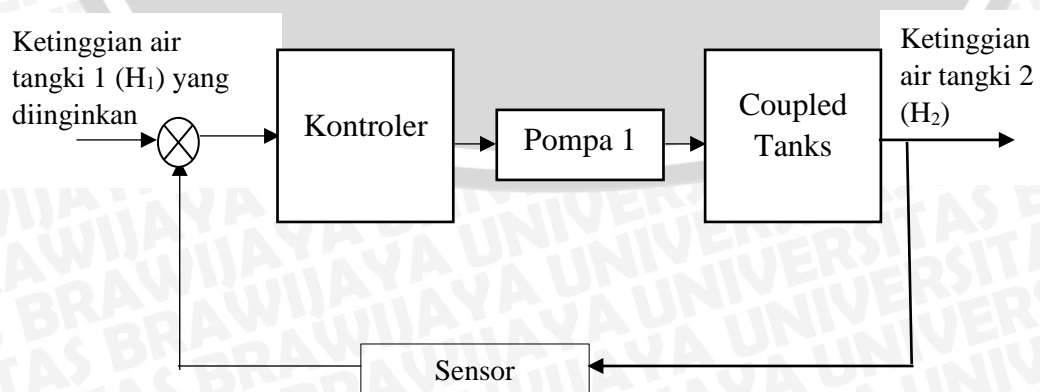
Karakterisasi dibagi menjadi beberapa bagian, yaitu:

1. Karakterisasi 2 buah pompa *Ford 24V DC*.
2. Karakterisasi driver motor (*High Current 30 A DC Motor Driver*)
3. Karakterisasi *plant* (dua tangki terhubung)
3. Perancangan algoritma *pole placement*
4. Penentuan observability pada diagram blok system
5. Simulasi sistem secara keseluruhan
6. Pengambilan kesimpulan dan saran

Perancangan algoritma meliputi algoritma kontrol optimal, *observability* dan *flowchart* program.

#### 3.1 Perancangan Blok Diagram Sistem

Perancangan diagram balok sistem berfungsi untuk menjelaskan sistem secara garis besar yang sangat diperlukan pada perancangan alat, dan diharapkan alat bisa bekerja sesuai yang diharapkan dan direncanakan. Perancangan blok diagram sistem dapat dilihat dalam Gambar 3.1.



**Gambar 3.1** Diagram Blok Sistem Pengaturan Ketinggian Level Air pada Tangki Terhubung



### 3.2 Karakterisasi *Ford Water Pump 24V DC*

Karakterisasi *Ford water pump* dilakukan untuk mengetahui karakter *Ford water pump*. Hal tersebut diperoleh dengan mengamati kecepatan *Ford water pump* terhadap perubahan dutycycle.

Peralatan yang digunakan terdiri atas:

1. Power Supply Unit
2. Mikrokontroler Arduino Mega 2560
3. *High Current 30 A DC Motor Driver* dengan *half bridge infineon BTS 7960 chip*
4. *Ford Water Pump 24V DC*
5. Plant tangki terhubung
6. Multimeter
7. Kabel penghubung

Langkah pengujian meliputi:

1. Menghubungkan Mikrokontroler Arduino Mega 2560 dengan seperangkat komputer
2. Memasukkan program duty cycle dari 0% sampai 100% ke dalam mikrokontroler Arduino Mega 2560 ke dalam
3. Menghubungkan Mikrokontroler Arduino Mega 2560 dengan port driver menggunakan kabel male to female.
4. Menghubungkan tegangan *output* Power Supply Unit (PSU) dengan *High Current 30 A DC Motor Driver* dengan *half bridge infineon BTS 7960 chip*.
5. Menghubungkan tegangan keluaran driver ke *Ford water pump*
6. Menghitung tegangan keluaran driver motor dengan multimeter

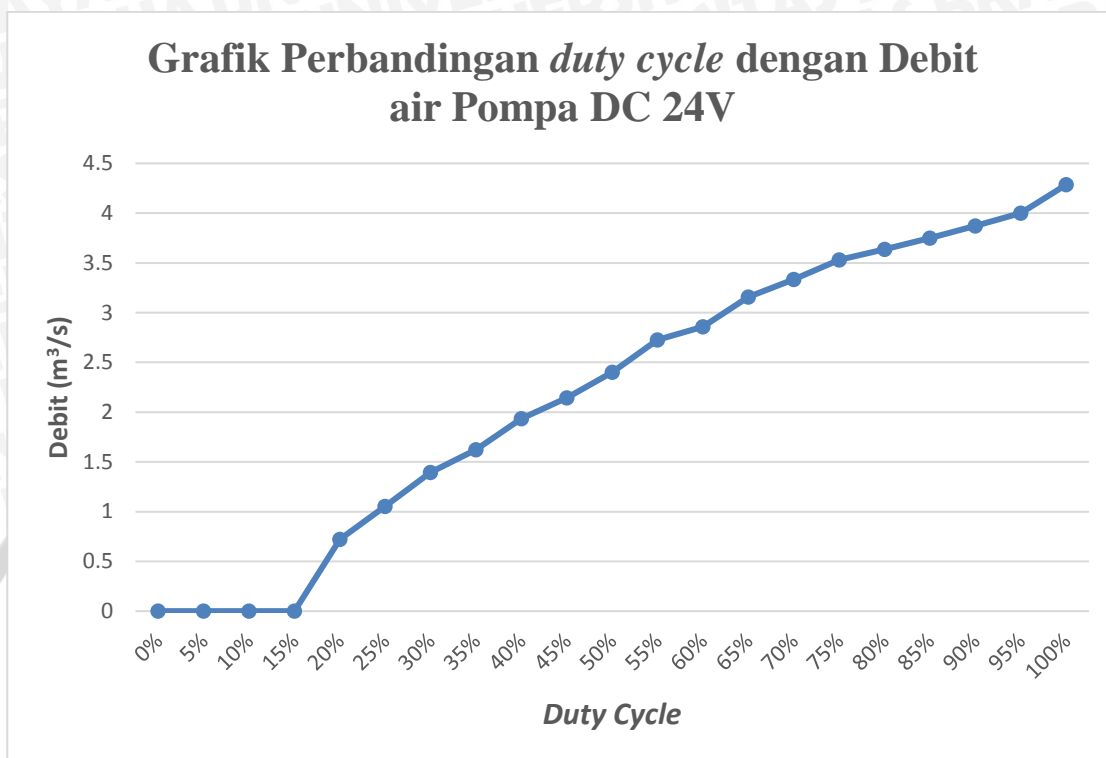
Data hasil pengujian kecepatan *Ford water pump 1* terhadap dutycycle ditunjukkan dalam Tabel 3.1

Tabel 3.1 Data pengujian kecepatan *Ford water pump* terhadap dutycycle dan tegangan keluaran driver

| No. | Duty Cycle | Pengukuran (m <sup>3</sup> /s) |
|-----|------------|--------------------------------|
| 1   | 0%         | 0                              |
| 2   | 5%         | 0                              |
| 3   | 10%        | 0                              |
| 4   | 15%        | 0                              |
| 5   | 20%        | $1.20482 \times 10^{-5}$       |
| 6   | 25%        | $1.75439 \times 10^{-5}$       |
| 7   | 30%        | $2.32558 \times 10^{-5}$       |
| 8   | 35%        | $2.7027 \times 10^{-5}$        |
| 9   | 40%        | $3.22581 \times 10^{-5}$       |
| 10  | 45%        | $3.57143 \times 10^{-5}$       |
| 11  | 50%        | 0.00004                        |
| 12  | 55%        | $4.54545 \times 10^{-5}$       |
| 13  | 60%        | $4.7619 \times 10^{-5}$        |
| 14  | 65%        | $5.26316 \times 10^{-5}$       |
| 15  | 70%        | $5.55556 \times 10^{-5}$       |
| 16  | 75%        | $5.88235 \times 10^{-5}$       |
| 17  | 80%        | $6.06061 \times 10^{-5}$       |
| 18  | 85%        | 0.0000625                      |
| 19  | 90%        | $6.45161 \times 10^{-5}$       |
| 20  | 95%        | $6.66667 \times 10^{-5}$       |
| 21  | 100%       | $7.14286 \times 10^{-5}$       |

Grafik perubahan debit air (l/menit) terhadap dutycycle dapat dilihat dalam Gambar

3.2



Gambar 3.2 Grafik perubahan debit air (l/menit) terhadap dutycycle

$$m = \frac{Y_{21} - Y_5}{X_{21} - X_5} = \frac{0.0000714286 - 0.0000120482}{1 - 0.2} = \frac{0.00005938}{0.8} = 0.00007423$$

### 3.3 Karakterisasi *Driver Motor (High Current 30 A DC Motor Driver dengan half bridge infineon BTS 7960 chip)*

Karakterisasi Motor Driver dilakukan untuk mengetahui karakter Motor Driver. Hal tersebut diperoleh dengan mengamati tegangan keluaran dan tegangan masukan Motor Driver terhadap perubahan dutycycle.

Peralatan yang digunakan terdiri atas:

1. Power Supply Unit
2. Mikrokontroler Arduino Mega 2560
3. *High Current 30 A DC Motor Driver dengan half bridge infineon BTS 7960 chip*
4. Plant tangki terhubung
5. Multimeter



## 6. Kabel penghubung

Langkah pengujian meliputi:

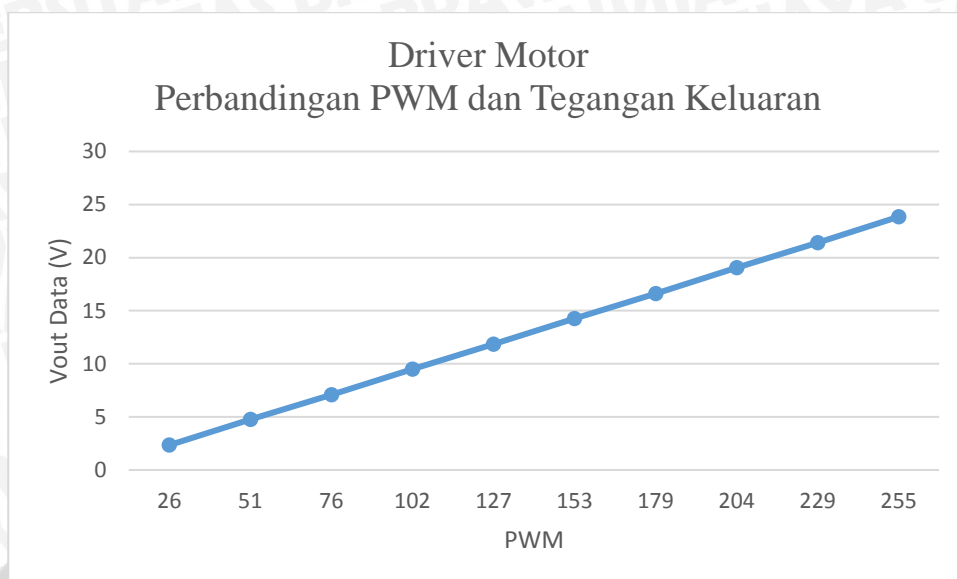
1. Menghubungkan Mikrokontroler Arduino Mega 2560 dengan seperangkat komputer
2. Memasukkan program duty cycle dari 0% sampai 100% ke dalam mikrokontroler Arduino Mega 2560 ke dalam
3. Menghubungkan Mikrokontroler Arduino Mega 2560 dengan port driver menggunakan kabel male to female.
4. Menghubungkan tegangan *output* Power Supply Unit (PSU) dengan *High Current* 30 A DC Motor *Driver* dengan *half bridge infineon* BTS 7960 *chip* .
5. Menghitung tegangan *output* driver motor dengan multimeter

Data hasil pengujian driver motor terhadap PWM arduino, tegangan masukan, dan tegangan keluaran ditunjukkan dalam Tabel 3.2.

**Tabel 3.2 Data pengujian *driver* terhadap PWM arduino, tegangan masukan, dan tegangan keluaran**

| No | PWM Arduino | DutyCycle | Vin data (V) | Vout data (V) | Vout teori (V) | Vin teori (V) |
|----|-------------|-----------|--------------|---------------|----------------|---------------|
| 1  | 26          | 10.2%     | 0.48         | 2.34          | 2.448          | 0.51          |
| 2  | 51          | 20%       | 0.99         | 4.75          | 4.8            | 1             |
| 3  | 76          | 29.8%     | 1.47         | 7.08          | 7.152          | 1.49          |
| 4  | 102         | 40%       | 1.98         | 9.5           | 9.6            | 2             |
| 5  | 127         | 49.8%     | 2.46         | 11.84         | 11.952         | 2.49          |
| 6  | 153         | 60%       | 2.97         | 14.27         | 14.4           | 3             |
| 7  | 179         | 70.2%     | 3.45         | 16.61         | 16.848         | 3.51          |
| 8  | 204         | 80%       | 3.96         | 19.05         | 19.2           | 4             |
| 9  | 229         | 89.8%     | 4.45         | 21.4          | 21.552         | 4.49          |
| 10 | 255         | 100%      | 4.96         | 23.85         | 24             | 5             |

Grafik pengujian driver motor terhadap PWM arduino, tegangan masukan, dan tegangan keluaran dapat dilihat dalam Gambar 3.3



Gambar 3.3 Grafik pengujian *driver* terhadap PWM arduino, tegangan masukan, dan tegangan keluaran

$$m = \frac{Vout_{10} - Vout_1}{Vin_{10} - Vin_1} = \frac{12.02 - 1.17}{4.96 - 0.48} = \frac{10.85}{4.48} = 2.421875$$

### 3.4 Karakterisasi Sensor Ultrasonik HC-SR04

Karakterisasi sensor ultrasonik HC-SR04 ini bertujuan untuk mengetahui kemampuan membaca ketinggian air pada *coupled tanks*. Hal tersebut dilakukan dengan cara membandingkan nilai *output* berupa tegangan keluaran sensor dengan *input* berupa ketinggian yang dibaca oleh sensor. Berikut adalah peralatan yang digunakan, langkah pengujian, dan hasil pengujian:

a. Peralatan yang digunakan

1. Arduino Mega 2560 R3
2. Sensor Ultrasonik HC-SR04
3. Kabel penghubung
4. Penggaris
5. Perekat
6. Perangkat Komputer

b. Langkah pengujian

1. Merangkai peralatan sensor ultrasonik HC-SR04, Arduino, dan perangkat computer menggunakan kabel penghubung.

2. Meletakkan sensor ultrasonik HC-SR04 pada tempat sensor di tangki pengisian.
  3. Mengisi tangki pengisian dengan air secara bertahap sesuai dengan ketinggian yang diinginkan.
  4. Mencatat dan membandingkan data ketinggian keluaran dari serial monitor Arduino dengan penggaris.
- c. Hasil pengujian

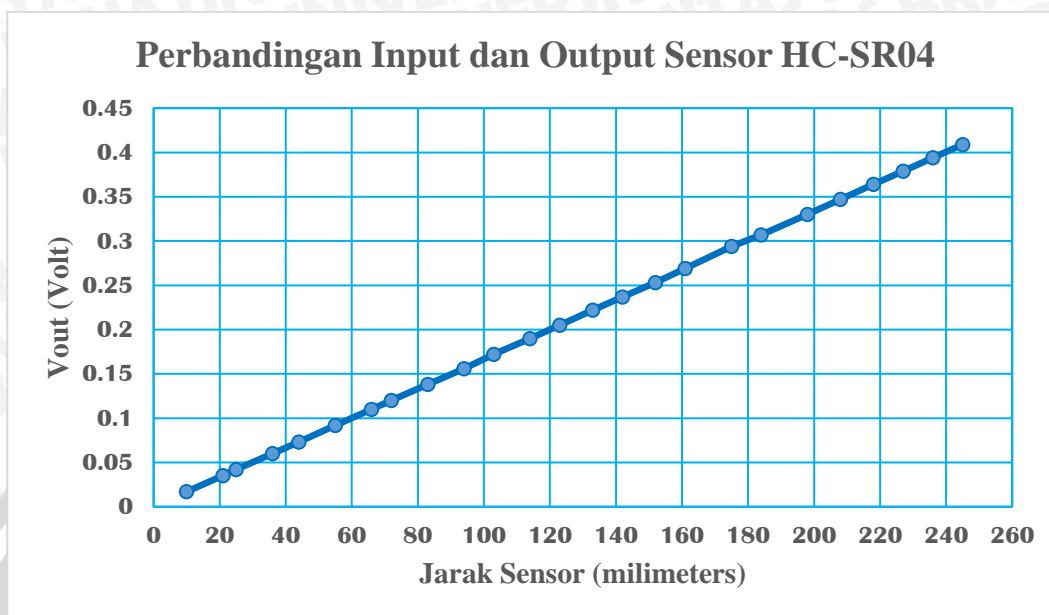
Data hasil pengujian sensor ultrasonik HC-SR04 ditunjukkan pada Tabel 3.3

**Tabel 3.3 Hasil Pengujian Sensor Ultrasonik HC-SR04**

| No. | Jarak Sebenarnya (mm) | Jarak Sensor (mm) | Vout (Volt) |
|-----|-----------------------|-------------------|-------------|
| 1   | 0                     | 0                 | 0           |
| 2   | 10                    | 10                | 0.017       |
| 3   | 20                    | 21                | 0.035       |
| 4   | 30                    | 25                | 0.042       |
| 5   | 40                    | 36                | 0.06        |
| 6   | 50                    | 44                | 0.073       |
| 7   | 60                    | 55                | 0.092       |
| 8   | 70                    | 66                | 0.11        |
| 9   | 80                    | 72                | 0.12        |
| 10  | 90                    | 83                | 0.138       |
| 11  | 100                   | 94                | 0.156       |
| 12  | 110                   | 103               | 0.172       |
| 13  | 120                   | 114               | 0.19        |
| 14  | 130                   | 123               | 0.205       |
| 15  | 140                   | 133               | 0.222       |
| 16  | 150                   | 142               | 0.237       |
| 17  | 160                   | 152               | 0.253       |
| 18  | 170                   | 161               | 0.269       |
| 19  | 180                   | 175               | 0.294       |
| 20  | 190                   | 184               | 0.307       |
| 21  | 200                   | 198               | 0.33        |
| 22  | 210                   | 208               | 0.347       |
| 23  | 220                   | 218               | 0.364       |
| 24  | 230                   | 227               | 0.379       |
| 25  | 240                   | 236               | 0.394       |
| 26  | 250                   | 245               | 0.409       |



Dari data hasil pengujian yang dilakukan pada sensor ultrasonik HC-SR04 didapatkan grafik seperti pada gambar 3.5



**Gambar 3.6** Grafik Hasil Pengujian Sensor Ultrasonik HC-SR04

Dari grafik diatas, didapatkan *gain* Pompa DC 12V dengan cara mencari gradien dari garis diatas yaitu sebesar:

$$m = \frac{Y_2 - Y_1}{X_2 - X_1} = \frac{0.409 - 0.017}{245 - 10} = \frac{0.392}{235} = 0.001668$$

### 3.5 Perancangan Algoritma

Dalam perancangan perangkat lunak, hal pertama yang dilakukan adalah mengetahui karakteristik motor generator DC. Setelah itu, dilakukan penghitungan untuk menentukan algoritma kontroler dengan *Output Feedback Control*. Setelah didapatkan algoritma, kemudian dibuat program untuk mikrokontroler dengan software program Arduino. Perancangan algoritma tersebut melalui beberapa tahap:

1. Mencari *state space* sistem
2. Mencari matriks R (matriks umpan balik)
3. Mencari matriks L (estimasi)
4. Mencari matriks M (pra filter)
5. Pembuatan simulasi pada software matlab.

### 3.6 Penentuan *State Space Plant*

Berdasarkan hasil pengamatan *plant coupled tanks* seperti yang ditunjukkan dalam Gambar 3.2, maka akan didapatkan persamaan dinamika fluida sebagai berikut:

$$\frac{H_1 - H_2}{R} = Q \quad (3.1)$$

$$A_1 \frac{dH_1}{dt} = Q_1 - Q \quad (3.2)$$

$$\frac{H_2}{R_2} = Q_2 \quad (3.3)$$

$$A_2 \frac{dH_2}{dt} = Q - Q_2 \quad (3.4)$$

dengan  $Q_1$  adalah debit air masukan tangki 1,  $Q_2$  adalah debit air keluaran tangki 2,  $Q$  adalah debit air pada saluran penghubung kedua tangki,  $H_1$  adalah ketinggian air pada tangki 1,  $H_2$  adalah ketinggian air pada tangki 2,  $A_1$  adalah luas alas tangki 1, dan  $A_2$  adalah luas alas tangki 2. Pada sistem ini, aliran air penghubung kedua tangki berupa aliran laminar. Aliran laminar ini analog dengan hukum Coulomb yang menyatakan bahwa arus berbanding lurus dengan beda potensial. Untuk itu diperoleh nilai resistansi ( $R$ ) sebagai berikut :

$$R = \frac{dH}{dQ} = \frac{H}{Q} \quad (3.5)$$

Gambaran tempat kedudukan untuk sistem ini adalah dimana  $Q_1$  ditinjau sebagai masukan dan  $H_2$  sebagai keluaran, maka variabel *state* sistem adalah sebagai berikut:

$$x_1 = H_2 \quad (3.6)$$

$$x_2 = H_1 \quad (3.7)$$

variabel masukan  $u$  sebagai

$$u = Q_1 \quad (3.8)$$

dan variabel keluaran  $y$  sebagai

$$y = H_2 \quad (3.9)$$

Kemudian gambaran *state* yang berkaitan dapat diperoleh dari persamaan (3.1) sampai (3.4), dan diperoleh:

$$A_2 \frac{dH_2}{dt} = \frac{H_1 - H_2}{R} - \frac{H_2}{R_2} \quad (3.10)$$

$$A_1 \frac{dH_1}{dt} = Q_1 - \frac{H_1 - H_2}{R} \quad (3.11)$$

substitusi

$$\frac{dH_2}{dt} = - \left( \frac{1}{R_{A_2}} + \frac{1}{R_2 A_2} \right) H_2 + \frac{1}{R A_2} H_1 \quad (3.12)$$

$$\frac{dH_1}{dt} = \frac{1}{R_{A_1}} H_2 - \frac{1}{R_{A_1}} H_1 + \frac{1}{A_1} Q_1 \quad (3.13)$$

dari persamaan diatas diubah ke dalam bentuk *state space* menjadi

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\left(\frac{1}{R_{A_2}} + \frac{1}{R_{2A_2}}\right) & \frac{1}{R_{A_2}} \\ \frac{1}{R_{A_1}} & -\frac{1}{R_{A_1}} \end{bmatrix} \begin{bmatrix} H_2 \\ H_1 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{A_1} \end{bmatrix} Q_1 \quad (3.14)$$

$$y = [1 \quad 0] \begin{bmatrix} H_2 \\ H_1 \end{bmatrix} \quad (3.15)$$

Dengan memasukkan nilai  $A_1 = A_2 = 0.00667 \text{ m}^2/\text{s}$ ,  $R = 25000 \text{ s/m}^2$ , dan  $R_2 = 8000 \text{ s/m}^2$  maka

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} -\left(\frac{1}{166.75} + \frac{1}{53.36}\right) & \frac{1}{166.75} \\ \frac{1}{166.75} & -\frac{1}{166.75} \end{bmatrix} \begin{bmatrix} H_2 \\ H_1 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{0.00667} \end{bmatrix} Q_1 \\ &= \begin{bmatrix} -(0.005997 + 0.0187) & 0.005997 \\ 0.005997 & -0.005997 \end{bmatrix} \begin{bmatrix} H_2 \\ H_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 149.92504 \end{bmatrix} Q_1 \end{aligned}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -0.024737 & 0.005997 \\ 0.005997 & -0.005997 \end{bmatrix} \begin{bmatrix} H_2 \\ H_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 149.92504 \end{bmatrix} Q_1 \quad (3.16)$$

$$y = [1 \quad 0] \begin{bmatrix} H_2 \\ H_1 \end{bmatrix} \quad (3.17)$$

### 3.7 Desain Kontrol Struktur *Output Feedback Control*.

Dalam mendesain kontroler struktur *Output Feedback Control* diperlukan nilai matriks A, B, C, R, L, M. Dimana matriks A adalah matriks keadaan orde  $n \times n$ , B adalah matriks masukan orde  $n \times m$ , C adalah matriks keluaran orde  $p \times n$ , R adalah matriks umpan balik, L adalah matriks estimasi, dan M adalah matriks pra filter. Langkah-langkah yang dilakukan untuk mencari matriks-matriks tersebut adalah sebagai berikut:

1. Fungsi alih *plant coupled tanks* didapatkan dengan cara menurunkan model matematik dari *plant* untuk kemudian diubah ke dalam bentuk *state space*. Bentuk *state space* yang dimaksud adalah sebagai berikut:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -0.024737 & 0.005997 \\ 0.005997 & -0.005997 \end{bmatrix} \begin{bmatrix} H_2 \\ H_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 149.92504 \end{bmatrix} Q_1$$

$$y = [1 \quad 0] \begin{bmatrix} H_2 \\ H_1 \end{bmatrix}$$

dari persamaan diatas didapatkan matriks A, B, C, dan D

$$A = \begin{bmatrix} -0.024737 & 0.005997 \\ 0.005997 & -0.005997 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 149.92504 \end{bmatrix}$$



$$C = [1 \quad 0]$$

2. Setelah diketahui matriks A, B, C, dan D dari plant selanjutnya adalah menentukan pole dari kontroler untuk mendapatkan nilai matriks umpan balik R. Matriks umpan balik R bisa didapatkan dengan menggunakan persamaan pada matlab yaitu:

$$r = \text{place}(a, b, rp);$$

dimana:

a= matriks A, matriks orde 2x2

b=matriks B, matriks orde 2x1

rp= pole kontroler yang nilainya bebas (rp=rp1;rp2), matriks orde 2x1

3. Setelah mendapatkan nilai matriks R, langkah selanjutnya adalah mencari matriks estimasi L. Matriks estimasi L bisa didapatkan dengan menggunakan persamaan pada matlab yaitu:

$$l = \text{place}(a', c', lp)';$$

dimana:

a= matriks A, matriks orde 2x2

c= matriks C, matriks orde 1x2

lp= pole observer yang nilainya bebas (lp=lp1;lp2), matriks orde 2x1

4. Setelah mendapatkan nilai matriks L, langkah selanjutnya adalah mencari matriks *pra filter* M menggunakan persamaan:

$$M = [-C(A - BR)^{-1}B]^{-1}$$

Dengan menggunakan matlab, kita bisa mencari nilai matriks M dengan menggunakan persamaan:

$$\text{dcg} = \text{dcgain}(at, bt, ct), dt);$$

$$m = 1/\text{dcg}$$

dimana:

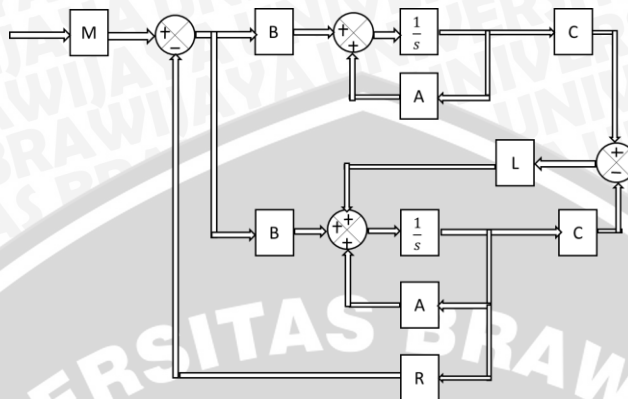
at=a-b\*gain;

bt=b;

ct=c;

### 3.8 Diagram Blok Perancangan Desain

Setelah dilakukan desain, didapatkan blok perancangan desain yang dapat dilihat dalam Gambar 3.4 :



Gambar 3.4 Diagram Blok Perancangan Desain



## BAB IV

### HASIL DAN PEMBAHASAN

Hasil dan pembahasan dilakukan dengan melakukan simulasi sistem dan analisis karakteristik *plant*. Simulasi dilakukan untuk mendapatkan letak kesalahan dan mempermudah analisis pada sistem. Analisis dilakukan untuk mengetahui karakteristik *plant* sebelum dan sesudah diberi kontroler.

Simulasi dibagi menjadi beberapa bagian, yaitu:

1. Simulasi penentuan matriks R, L, M dari perubahan pole kontroler dan perubahan pole observer.
2. Simulasi dengan gangguan.

#### 4.1 Simulasi Penentuan Matriks R dan L dari Perubahan Pole Kontroler (rp) dan Perubahan Pole Observer (op).

Simulasi penentuan matriks R dan L menggunakan software matlab dengan sintaks:

```
% program lqr
%Matriks plant
clear;
a=[-0.024737 0.005997 ; 0.005997 -0.005997];
b=[0;149.925];
c=[1 0];
%-----
bsize=size(b);
noinp=bsize(2);
csize=size(c);
noout=csize(1);
rr= 1; % dibuat r berubah
qq= 1; % elemen diagonal matrik q
[ gain,p]=lqr(a,b,qq*eye(size(a)),rr*eye(size(noinp)));
cgh=gain;
dgh=zeros(noinp);
%penetapan pole observer untuk mengetahui nilai matriks L
lp=[-2.6834; -0.0098];
l=place(a',c',lp)';
%penetapan pole kontroler untuk mengetahui nilai matriks R
rp=[-50.9912; -0.0746];
k=place(a,b,rp);
```



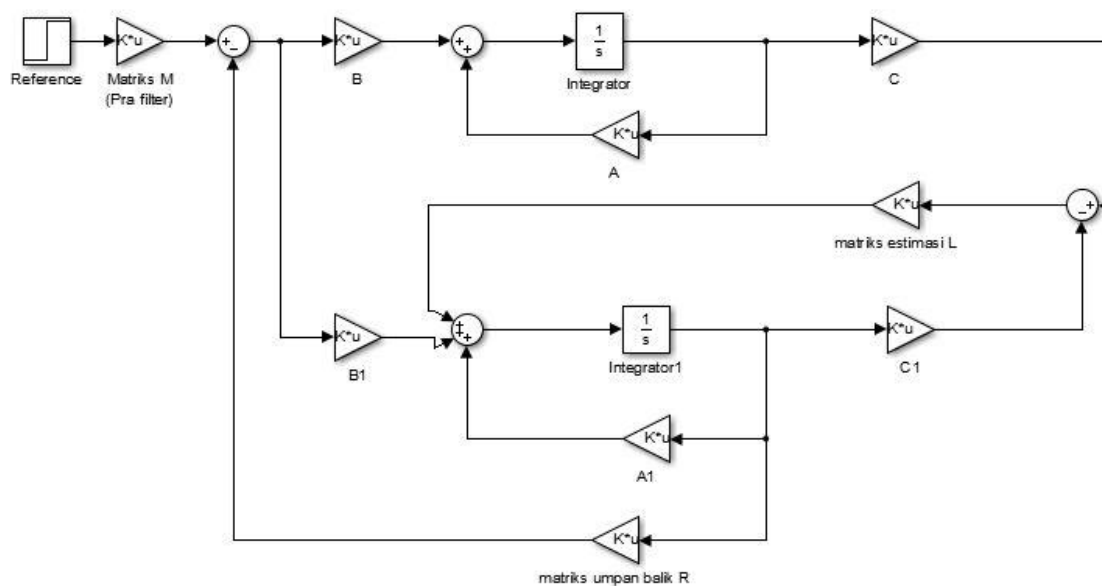


```

% a,b,cgh,dgh adalah matrik loop terbuka
at=a-b*gain;
bt=b;
ct=c;
dt=zeros(noout,noinp);
t=0:250;
woq=logspace(-1,3,100);
out=step(at,bt,ct,dt,1,t); % respon closed loop
out=out/out(length(out));
[mag,phase]=bode(a,b,cgh,dgh,1,woq);
[cmag,cphase]=bode(at,bt,ct,dt,1,woq);
kq=gain;
clpg=eig(a-b*gain);
yq=out;
dcg=dcgain(at,bt,ct,dt);
m=1/dcg;
e=eig(a-l*c); % nilai eigen value
w=logspace(-1,1,100);
figure(1);
plot(t,yq),grid
legend('r=1 q=1','r=0.001','r=1','r=10','r=10000');

```

Setelah mendapatkan nilai R dan L, simulasi dilakukan dengan menggunakan simulink pada matlab dengan struktur *output feedback control*. Simulasi penggunaan matriks R, L, dan M dapat dilihat dalam Gambar 4.1



Gambar 4.1 Simulasi penggunaan matriks R, L, dan M

## 4.2 Simulasi Perubahan Pole Observer terhadap perubahan nilai matriks L

Sebelum melakukan implementasi pada alat, kita perlu mengetahui pengaruh perubahan pada pole observer ( $op$ ) terhadap energi atau sinyal kontrol ( $U$ ) dan terhadap *output* sistem. Kita misalkan  $op = [op1; op2]$ .

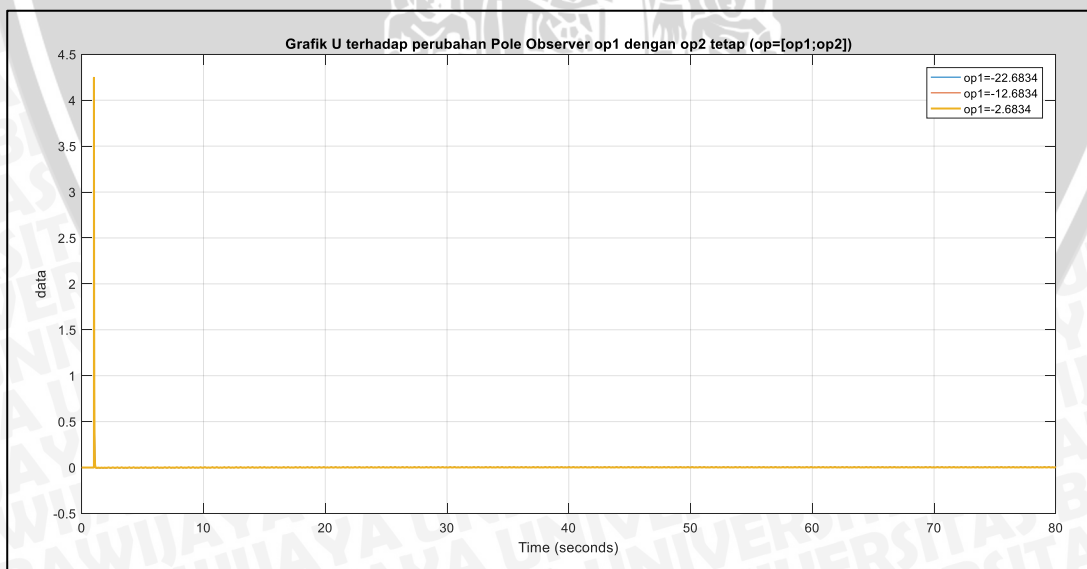
### 4.2.1 Perubahan Nilai $op1$

Simulasi perubahan nilai  $op1$  menggunakan 3 data (Lihat Tabel 4.1). Dari simulasi yang dilakukan akan dilihat pengaruh dari perubahan nilai  $op1$  terhadap nilai  $U$  (lihat Gambar 4.2) dan terhadap tegangan *output* sistem (lihat Gambar 4.3).

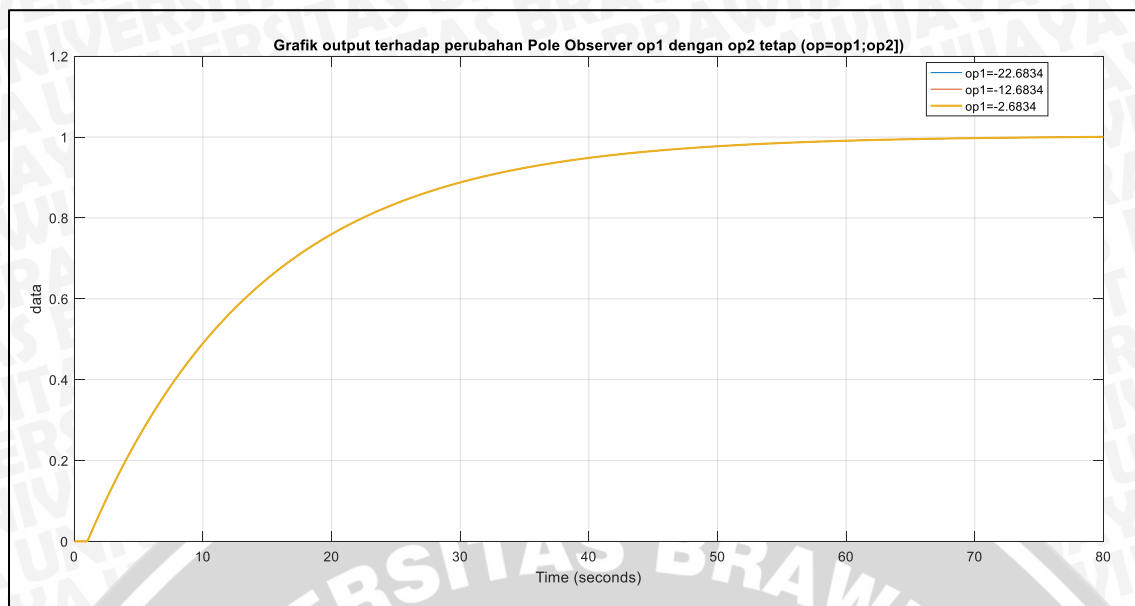
Tabel 4. 1 Nilai Matriks R dan L terhadap Perubahan Nilai Observer Pole  $op1$

| No | L pole observer     | R pole kontroler    | R               | L                  |
|----|---------------------|---------------------|-----------------|--------------------|
| 1  | [-22.6834; -0.0098] | [-50.9912; -0.0746] | [2.8266 0.3404] | [22.6625; 14.3869] |
| 2  | [-12.6834; -0.0098] | [-50.9912; -0.0746] | [2.8266 0.3404] | [12.6625; 8.0454]  |
| 3  | [-2.6834; -0.0098]  | [-50.9912; -0.0746] | [2.8266 0.3404] | [2.6625; 1.7039]   |

Dari data dalam Tabel 4.1 simulasi dengan menggunakan simulink dapat dilakukan untuk melihat perubahan  $U$  dan *output* sistem terhadap perubahan *observer pole*  $op1$ .



Gambar 4.2 Grafik U dengan nilai  $op1$  berbeda-beda  $op2$  tetap,  $op=[op1;op2]$



**Gambar 4.3** Output sistem dengan nilai  $op1$  berbeda-beda  $op2$  tetap,  $op=[op1;op2]$

Dari Gambar 4.2 dan Gambar 4.3 dapat dilihat bahwa perubahan nilai *observer pole*  $op1$  tidak mempengaruhi nilai  $U$ . Begitu juga dengan *output* sistem, perubahan nilai *observer pole*  $op1$  tidak mempengaruhi *output* sistem.

#### 4.2.2 Perubahan Nilai $op2$

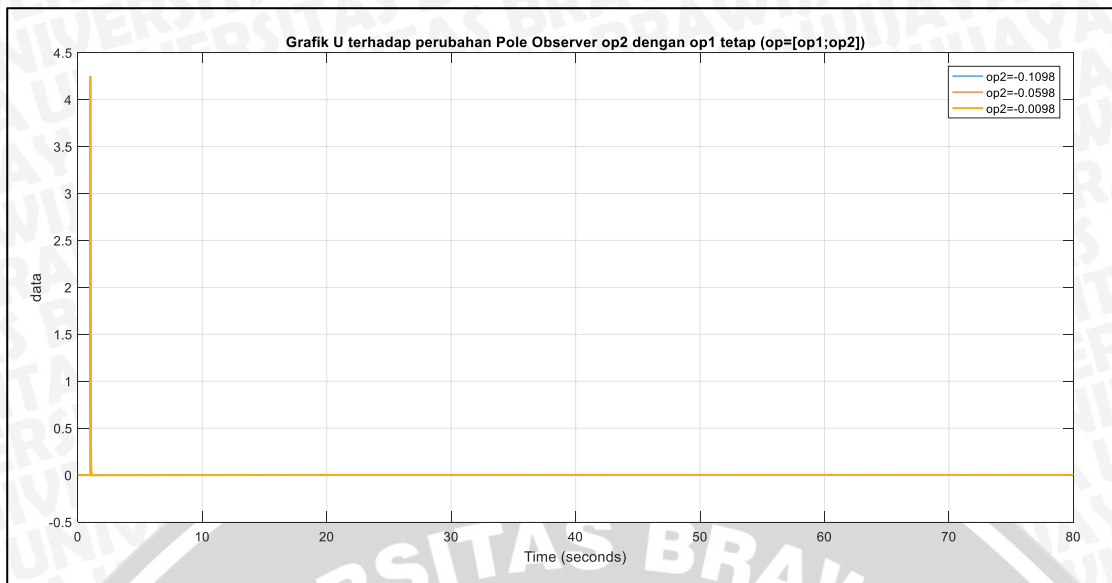
Simulasi perubahan nilai  $op2$  menggunakan 3 data (lihat Tabel 4.2). Dari simulasi yang dilakukan akan dilihat pengaruh dari perubahan nilai  $op1$  terhadap nilai  $U$  (lihat Gambar 4.4) dan terhadap *output* sistem (lihat Gambar 4.5).

**Tabel 4. 2** Nilai Matriks  $R$  dan  $L$  terhadap Perubahan Nilai Observer Pole  $op2$

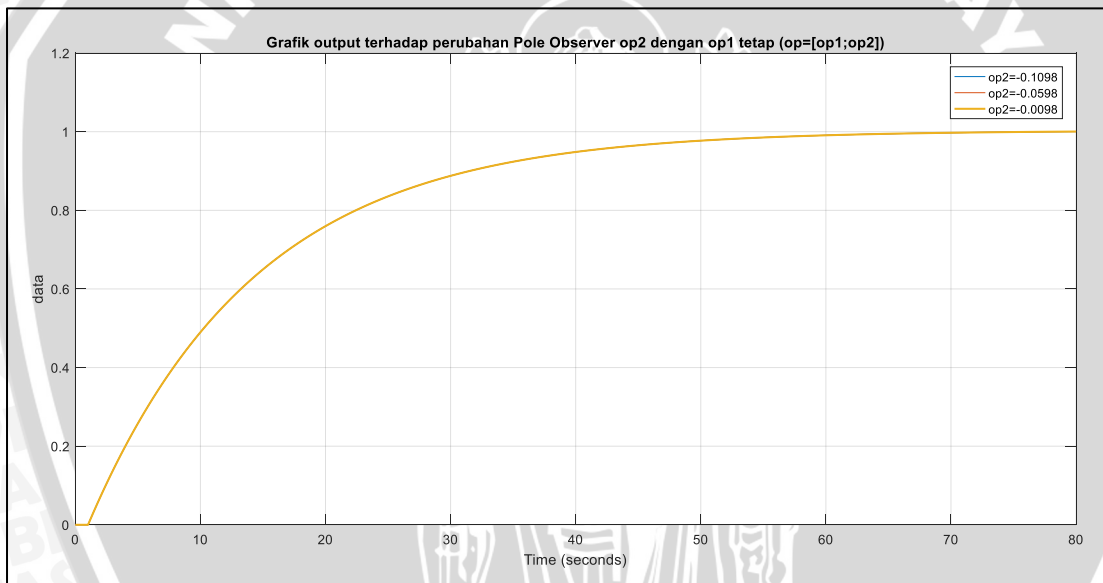
| No | L pole observer    | R pole kontroler    | R               | L                 |
|----|--------------------|---------------------|-----------------|-------------------|
| 1  | [-2.6834; -0.1098] | [-50.9912; -0.0746] | [2.8266 0.3404] | [2.7625; 46.3496] |
| 2  | [-2.6834; -0.0598] | [-50.9912; -0.0746] | [2.8266 0.3404] | [2.7125; 24.0267] |
| 3  | [-2.6834; -0.0098] | [-50.9912; -0.0746] | [2.8266 0.3404] | [2.6625; 1.7039]  |

Dari data tabel 4.2 simulasi dengan menggunakan simulink dapat dilakukan untuk melihat perubahan  $U$  dan *output* sistem terhadap perubahan *observer pole*  $op2$ .





Gambar 4.4 Grafik U dengan nilai  $op2$  berbeda-beda  $op1$  tetap,  $op=[op1;op2]$



Gambar 4.5 Output sistem dengan nilai  $op2$  berbeda-beda  $op1$  tetap,  $op=[op1;op2]$

Dari Gambar 4.4 dan Gambar 4.5 dapat dilihat bahwa perubahan nilai *observer pole*  $op2$  tidak mempengaruhi nilai U. Begitu juga dengan *output sistem*, perubahan nilai *observer pole*  $op2$  tidak mempengaruhi tegangan *output sistem*.

### 4.3 Simulasi Perubahan Pole Kontroler terhadap perubahan nilai matriks R.

Sebelum melakukan implementasi pada alat, kita perlu mengetahui pengaruh perubahan pada *pole observer* terhadap U dan terhadap *output* sistem. Kita misalkan  $rp = [rp1; rp2]$ .

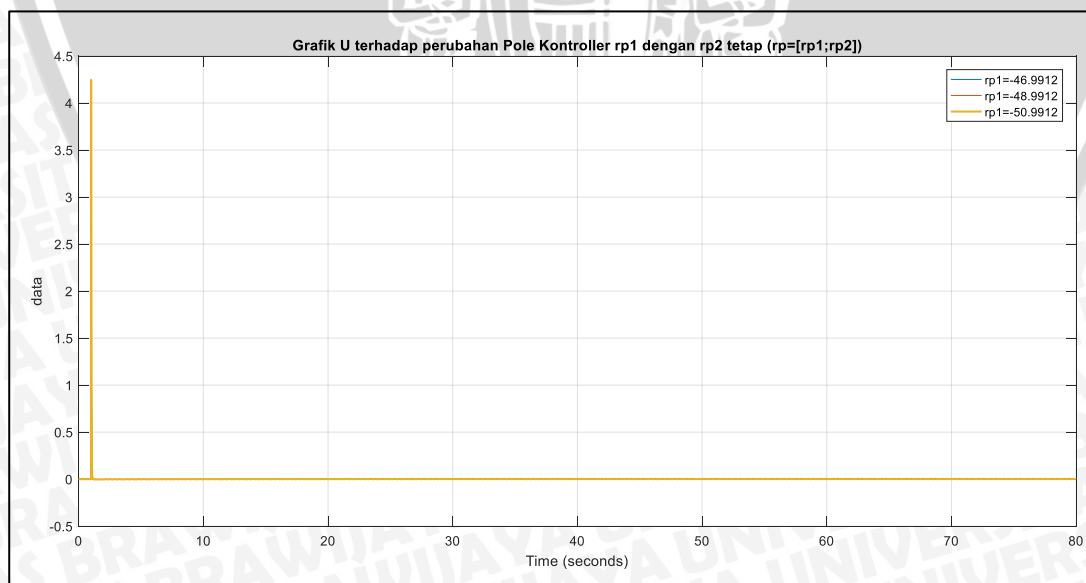
#### 4.3.1 Perubahan Nilai rp1

Simulasi perubahan nilai rp1 menggunakan 3 data (Lihat Tabel 4.3). Dari simulasi yang dilakukan akan dilihat pengaruh dari perubahan nilai rp1 terhadap nilai U (lihat Gambar 4.6) dan terhadap tegangan *output* sistem (lihat Gambar 4.7).

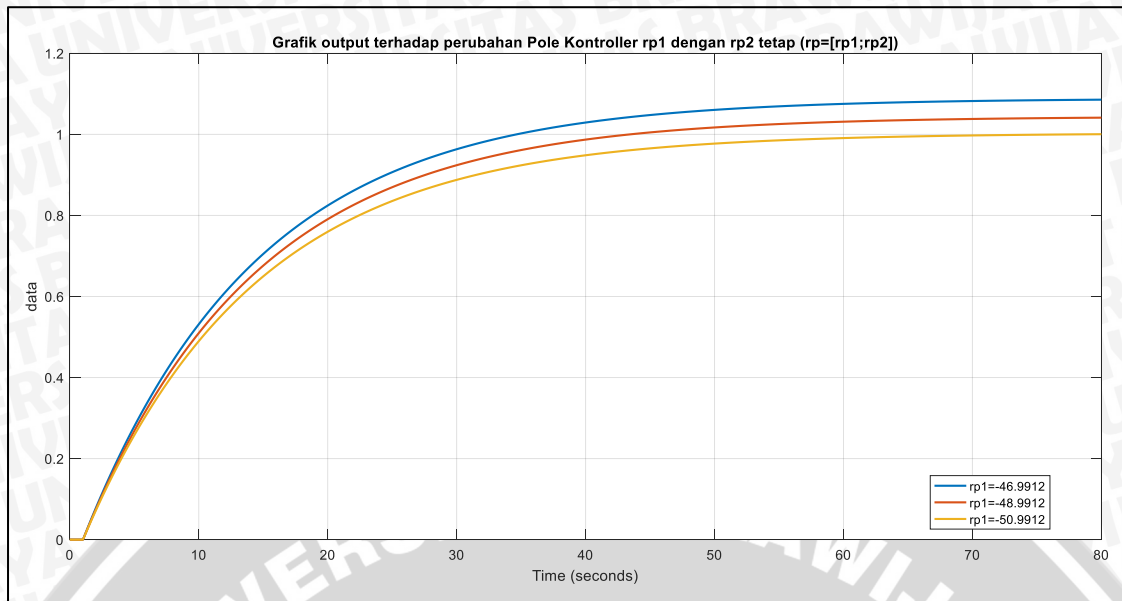
Tabel 4. 3 Nilai Matriks R dan L terhadap Perubahan Nilai Kontroler Pole rp1

| No | L pole observer    | R pole kontroler    | R               | L                |
|----|--------------------|---------------------|-----------------|------------------|
| 1  | [-2.6834; -0.0098] | [-46.9912; -0.0746] | [2.6047 0.3137] | [2.6625; 1.7039] |
| 2  | [-2.6834; -0.0098] | [-48.9912; -0.0746] | [2.7157 0.3271] | [2.6625; 1.7039] |
| 3  | [-2.6834; -0.0098] | [-50.9912; -0.0746] | [2.8266 0.3404] | [2.6625; 1.7039] |

Dari data tabel 4.3 simulasi dengan menggunakan simulink dapat dilakukan untuk melihat perubahan U dan *output* sistem terhadap perubahan kontroler pole rp1.



Gambar 4.6 Grafik U dengan nilai rp1 berbeda-beda rp2 tetap,  $rp=[rp1;rp2]$



**Gambar 4.7 Output Sistem dengan nilai  $rp1$  berbeda-beda  $rp2$  tetap,  $rp=[rp1;rp2]$**

Dari Gambar 4.6 dapat dilihat bahwa perubahan nilai kontroler *pole*  $rp1$  tidak mempengaruhi grafik  $U$  sedangkan pada gambar 4.7 dapat dilihat bahwa semakin kecil nilai kontroler *pole*  $rp1$  maka nilai *output* akan semakin kecil pula.

#### 4.3.2 Perubahan Nilai $rp2$

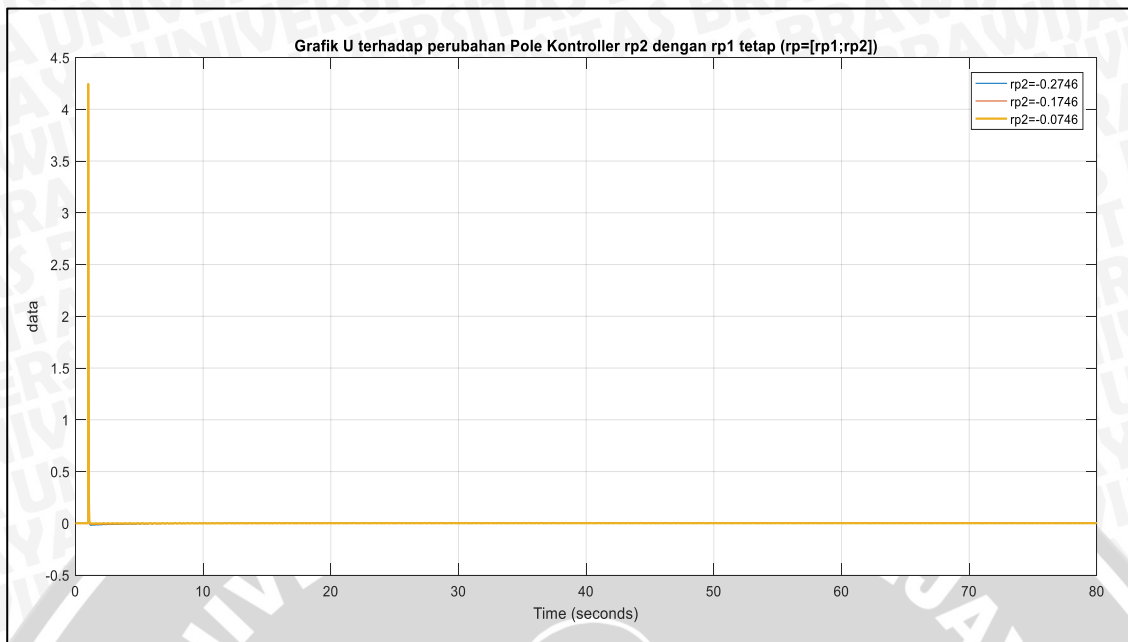
Simulasi perubahan nilai  $rp2$  menggunakan 3 data (lihat Tabel 4.4). Dari simulasi yang dilakukan akan dilihat pengaruh dari perubahan nilai  $rp2$  terhadap nilai  $U$  (lihat Gambar 4.8) dan terhadap *output* sistem (lihat Gambar 4.9).

**Tabel 4. 4 Nilai Matriks  $R$  dan  $L$  terhadap Perubahan Nilai Kontroler Pole  $rp2$**

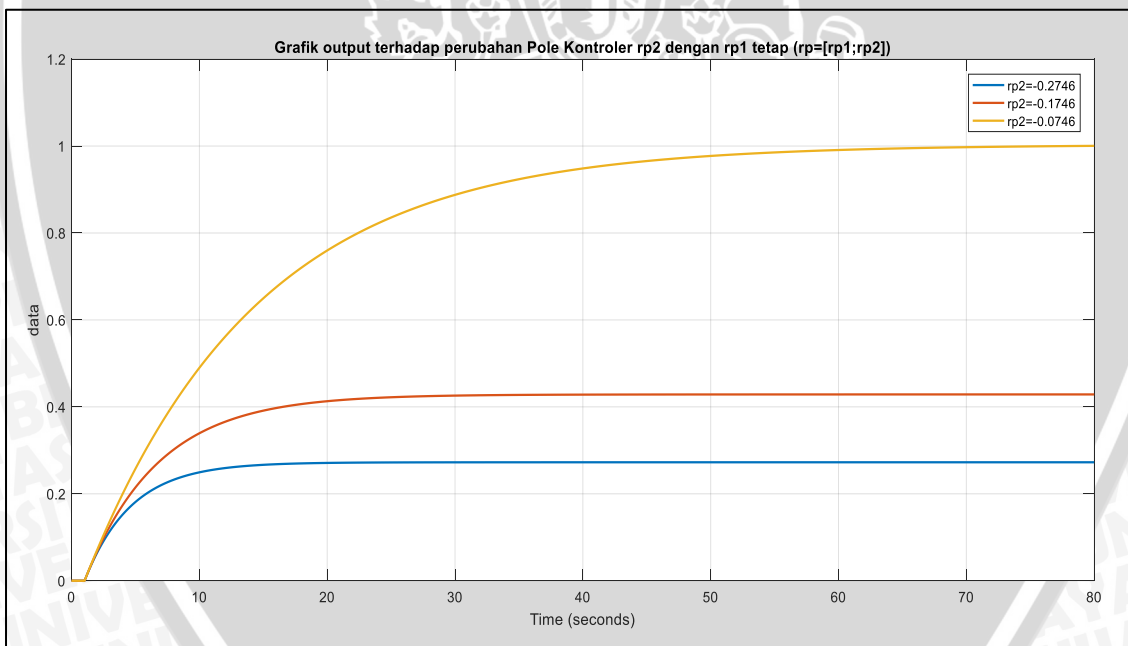
| No | L pole observer    | R pole kontroler    | R               | L                |
|----|--------------------|---------------------|-----------------|------------------|
| 1  | [-2.6834; -0.0098] | [-46.9912; -0.0746] | [2.6047 0.3137] | [2.6625; 1.7039] |
| 2  | [-2.6834; -0.0098] | [-48.9912; -0.0746] | [2.7157 0.3271] | [2.6625; 1.7039] |
| 3  | [-2.6834; -0.0098] | [-50.9912; -0.0746] | [2.8266 0.3404] | [2.6625; 1.7039] |

Dari data dalam Tabel 4.4 simulasi dengan menggunakan simulink dapat dilakukan untuk melihat perubahan  $U$  dan tegangan *output* sistem terhadap perubahan kontroler pole  $rp2$ .





Gambar 4.8 Grafik U dengan nilai  $rp_2$  berbeda-beda  $rp_1$  tetap,  $rp=[rp_1;rp_2]$

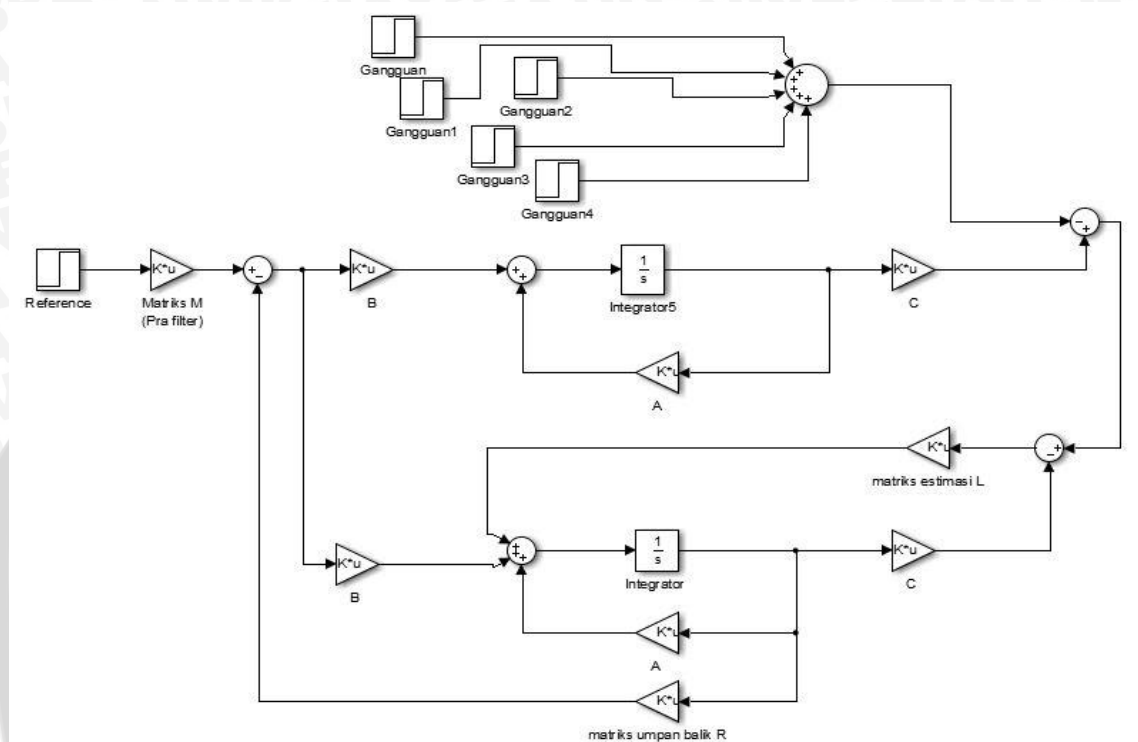


Gambar 4.9 Output Sistem dengan nilai  $rp_2$  berbeda-beda  $rp_1$  tetap,  $rp=[rp_1;rp_2]$

Dari Gambar 4.8 dapat dilihat bahwa perubahan nilai kontroler *pole*  $rp_1$  tidak mempengaruhi grafik  $U$  sedangkan pada gambar 4.9 dapat dilihat bahwa semakin kecil nilai kontroler *pole*  $rp_2$  maka *output* sistem untuk mencapai *steady state* akan semakin cepat.

#### 4.4 Simulasi dengan Gangguan

Setelah diketahui pengaruh perubahan kontroler pole dan *observer pole*, langkah selanjutnya adalah simulasi dengan memberikan gangguan pada tegangan *output* sistem. Simulasi dilakukan dengan menggunakan simulink pada matlab seperti dalam Gambar 4.10.



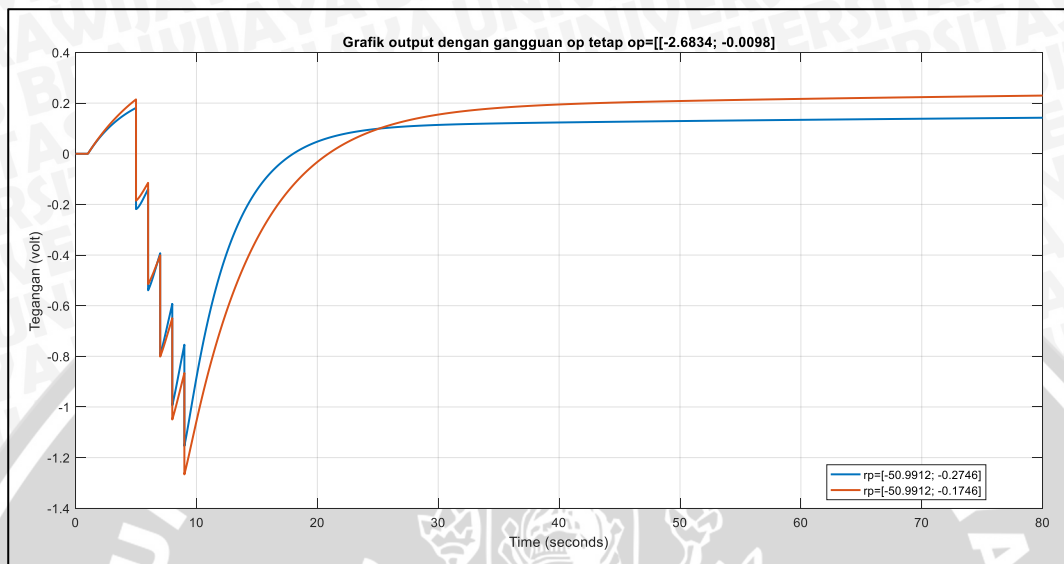
Gambar 4.10 Simulasi dengan gangguan

Dari simulasi perubahan nilai kontroler pole dan *observer pole*, maka diputuskan simulasi dengan gangguan menggunakan 2 data yang berbeda. (lihat Tabel 4.5). Data ini dipilih dengan melihat dari grafik  $U$  dan tegangan *output* sistem pada perubahan kontroler pole. Nilai matriks  $R$  dan  $L$  yang dipilih adalah nilai yang mempunyai grafik dengan nilai  $U$  yang tidak terlalu besar, waktu yang diperlukan sistem untuk mencapai *steady state* cukup cepat.

Tabel 4. 5 Nilai Matriks  $R$ ,  $L$ ,  $M$  untuk Simulasi dengan Gangguan

| No | $L$ pole observer  | $R$ pole kontroler  | $R$              | $L$              |
|----|--------------------|---------------------|------------------|------------------|
| 1  | [-2.6834; -0.0098] | [-50.9912; -0.2746] | [14.1638 0.3417] | [2.6625; 1.7039] |
| 2  | [-2.6834; -0.0098] | [-50.9912; -0.1746] | [8.4952 0.3411]  | [2.6625; 1.7039] |

Dari data dalam Tabel 4.5 simulasi dengan menggunakan simulink dapat dilakukan untuk melihat tegangan *output* sistem apabila diberi gangguan (lihat Gambar 4.11).



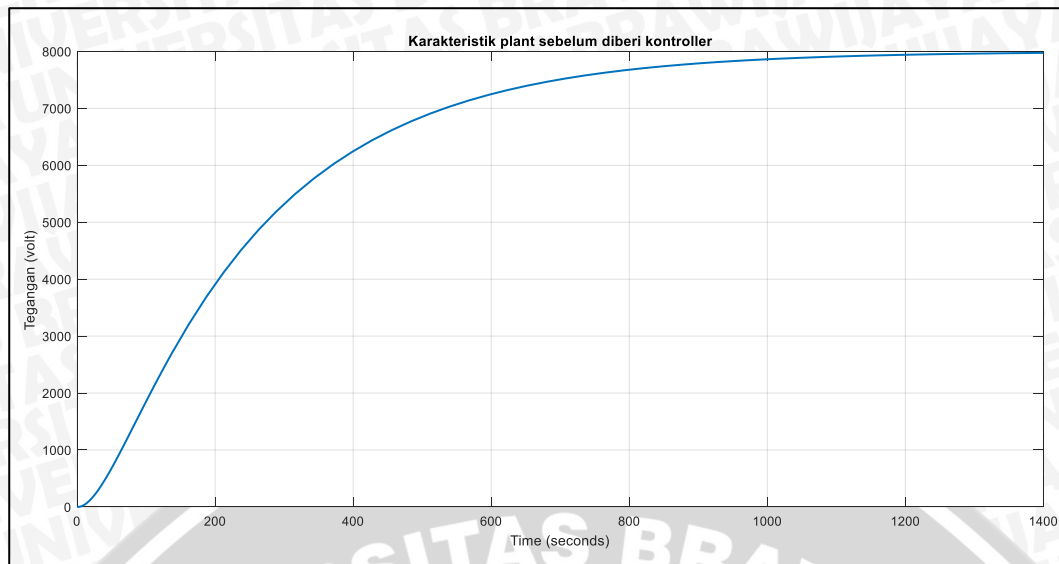
**Gambar 4.11 Tegangan *Output* sistem dengan gangguan**

Dari simulasi yang dilakukan, dapat dilihat bahwa *output* sistem yang bisa kembali ke keadaan semula adalah data dengan nilai  $rp=[-50.991; -0.1746]$  meskipun waktu yang diperlukan *output* untuk mencapai *steady state* lebih lambat dari data dengan nilai  $rp=[-50.991; -0.2746]$ .

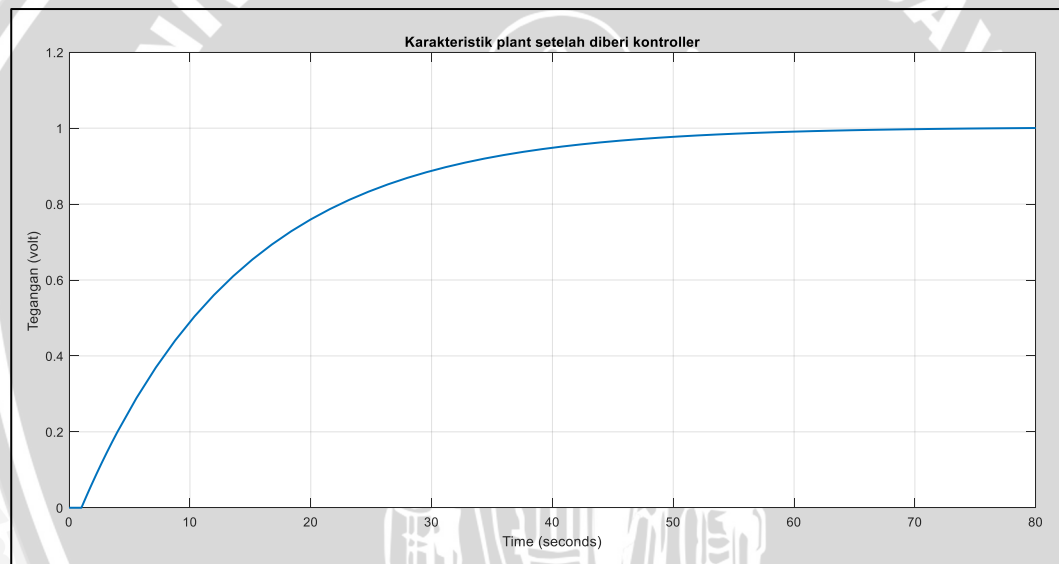
#### 4.5 Simulasi Karakteristik Plant Sebelum dan Sesudah diberi Kontroler

Simulasi karakteristik plant sebelum dan sesudah diberi kontroler ini berfungsi untuk mengetahui pengaruh kontroler terhadap tegangan *output* plant (lihat Gambar 4.12 dan Gambar 4.13).





Gambar 4.12 Karakteristik tegangan *output plant* sebelum diberi kontroler



Gambar 4.13 Karakteristik tegangan *output plant* sesudah diberi kontroler

Dari simulasi dapat diketahui bahwa *output plant* saat diberi *input unit step* sebelum diberi kontroler mempunyai *settling time* 1200 detik dengan *error steady state*  $>50\%$  tanpa *overshoot*. Sedangkan *output plant* saat diberi *input unit step* setelah diberi kontroler mempunyai *settling time* 60 detik dengan *error steady state*  $<1\%$  tanpa *overshoot*.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

1. Berdasarkan cara penurunan model matematik dari *plant*, kemudian didapatkan bentuk *state space* :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -0.024737 & 0.005997 \\ 0.005997 & -0.005997 \end{bmatrix} \begin{bmatrix} H_2 \\ H_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 149.92504 \end{bmatrix} Q_1$$

$$y = [1 \quad 0] \begin{bmatrix} H_2 \\ H_1 \end{bmatrix}$$

maka nilai matriks  $A = [-0.024737 \quad 0.005997; 0.005997 \quad -0.005997]$ ,  $B = [0; 149.925]$ ,  $C = [1 \quad 0]$

2. Berdasarkan hasil simulasi didapatkan nilai kontroler *pole* dan *observer pole* yang optimal dengan energi yang minimal dan settling time yang cepat yaitu  $op = [-2.6834; -0.0098]$ ,  $rp = [-50.9912; -0.0746]$  dan menghasilkan nilai matriks  $R = [2.8266 \quad 0.3404]$ ,  $L = [2.6625; 1.7039]$ , dan  $M = 4.2444$
3. Simulasi *output plant* saat diberi *input unit step* setelah diberi kontroler metode *Output Feedback Control* mempunyai *settling time* 60 detik dengan *error steady state* <1% dan tanpa *overshoot*.
4. Hasil dari simulasi pada simulink saat diberi gangguan, *output* sistem bisa kembali ke keadaan semula pada nilai  $rp = [-50.991; -0.1746]$  yang mempunyai *settling time* 40 detik dengan *error steady state* <1% dan tanpa *overshoot*.

#### 5.2 Saran

Saran yang diberikan untuk penelitian selanjutnya adalah dengan menggunakan metode *state feedback control* agar bisa didapatkan respon sistem dengan *settling time* yang kecil.

**DAFTAR PUSTAKA**

- Amalia, Z. 2016. *Desain Kontroler Struktur Output Feedback Control Dengan Pole Placement Pada Pengontrolan Tegangan Output Generator DC*. Skripsi. Teknik Elektro. Universitas Brawijaya. Malang.
- Gopal, M. 1987. *Modern Control System Theory*. John Wiley and Sons, Inc.
- Houpis, D. 1995. *Linear Control Systems Analysis and Design*. USA: McGraw-Hill, Inc.
- Kwakernaak, H., Sivan, R. 2011. *Linear Optimal Control Systems*. United States: Wiley Interscience.
- Lewis, S. 1995. *Optimal Control*. USA: John Wiley & Sons, Inc
- Ogata, K. 1997. *Modern Control Engineering*. Prentice Hall. New Jersey.
- Ogata, K. 1997. *Teknik Kontrol Automatik*. terjemahan: Edi Laksono Ir. Jakarta : Penerbit Erlangga.
- Ogata, K. 2010. *Modern Control Engineering*. New Jersey: Pearson Education, Inc.
- Ramadhan, K. 2014. *Sistem Kontrol Multivariabel Temperatur dan Level dengan Yokogawa DCS Centum VP*. Malang: Teknik Elektro Universitas Brawijaya.
- Rusli, M. 2015. *Dasar Teknik Kontrol*. Malang: Penerbit UB Press