

MODUL PANDUAN

PRAKTIKUM DASAR SISTEM KENDALI



Oleh:

SWADEXI ISTIQPHARA, S.T., M.T

ANISA ULYA DARAJAT, S.T., M.T

**LABORATORIUM SISTEM KENDALI DAN ELEKTRONIKA LANJUT
PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JAMBI
2017**

TATA TERTIB PRAKTIKUM

1. Mahasiswa yang diizinkan mengikuti praktikum adalah yang telah terdaftar dan memenuhi syarat yang ditentukan.
2. Praktikum dilaksanakan sesuai dengan jadwal dan praktikan harus hadir 5 menit sebelum praktikum dimulai. Bagi praktikan yang tidak hadir pada waktu tersebut dianggap mengundurkan diri dari praktikum. Praktikan harus mengisi daftar hadir pada setiap pelaksanaan percobaan.
3. Praktikan harus mengikuti pretest yang dilaksanakan sebelum praktikum keseluruhan.
4. Praktikan harus mengikuti posttest yang dilaksanakan setelah praktikum.
5. Penilaian praktikum didasarkan atas:
 - a. Tugas Pendahuluan : 10 %
 - b. Pretest : 5 %
 - c. Posttest : 5 %
 - d. Keaktifan : 15 %
 - e. Laporan : 40 %
 - f. Asistensi : 15 %
 - g. Alat : 10 %
6. Praktikan dilarang merokok, makan dan minum selama berada di dalam laboratorium.
7. Praktikan harus berpakaian rapi, tidak diperkenankan memakai kaos oblong dan sandal.
8. Praktikan dilarang ribut selama berada di dalam laboratorium dan wajib menjaga kebersihan di dalam maupun di luar laboratorium.
9. Bagi yang melanggar akan mendapat sanksi dikeluarkan dari ruang laboratorium dan dianggap tidak mengikuti praktikum.

Jambi, 4 September 2017

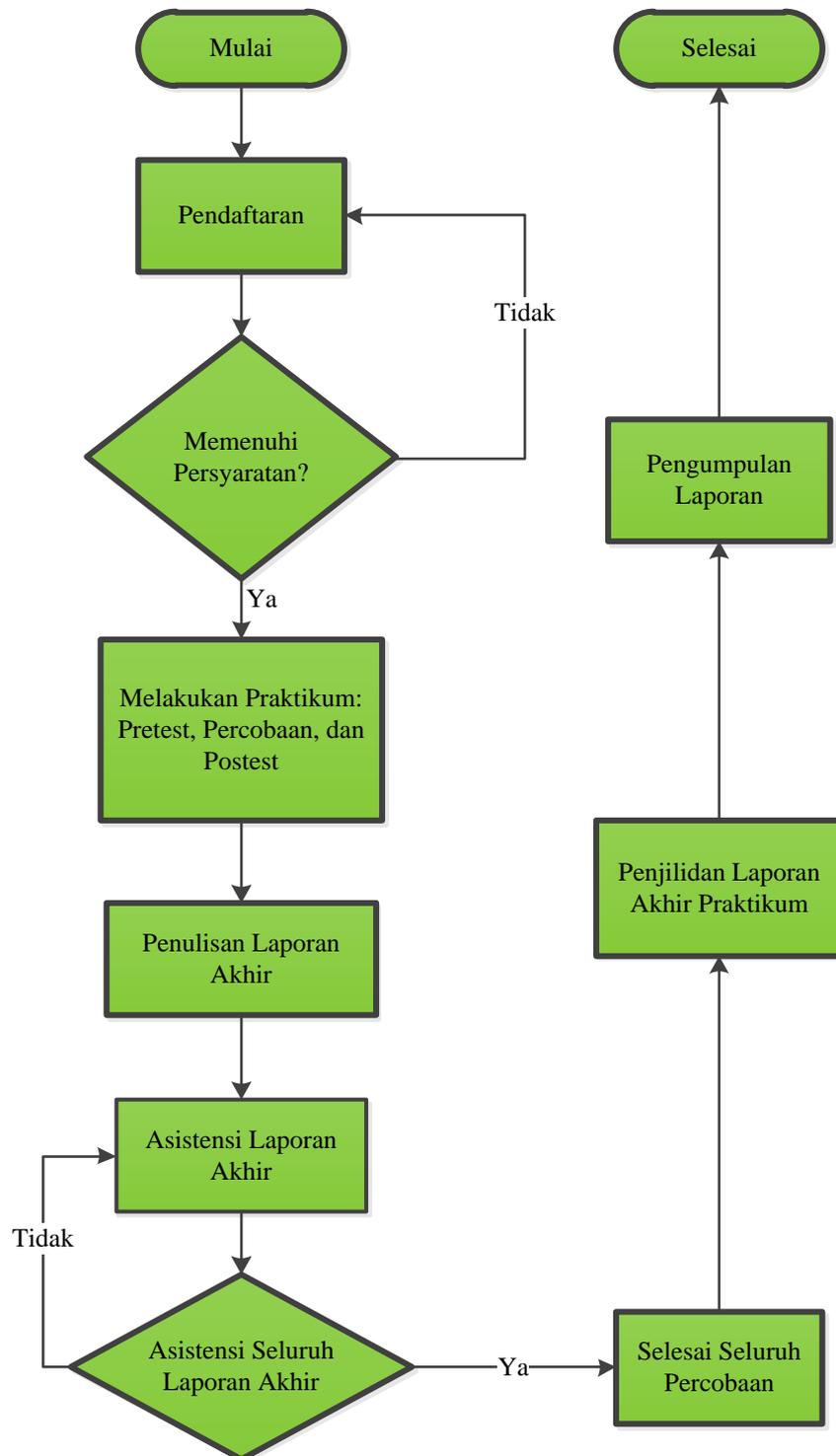
Dosen P.J. Praktikum Sistem Kendali

Dosen P.J. Praktikum Sistem Kendali

Swadexi Istiqphara, S.T, M.T.
NIDK. 201511111010

Anisa Ulya Darajat, S.T, M.T.
NIDK. 201704112003

DIAGRAM ALIR PELAKSANAAN PRAKTIKUM



PROSES PELAKSANAAN PRAKTIKUM SISTEM KENDALI

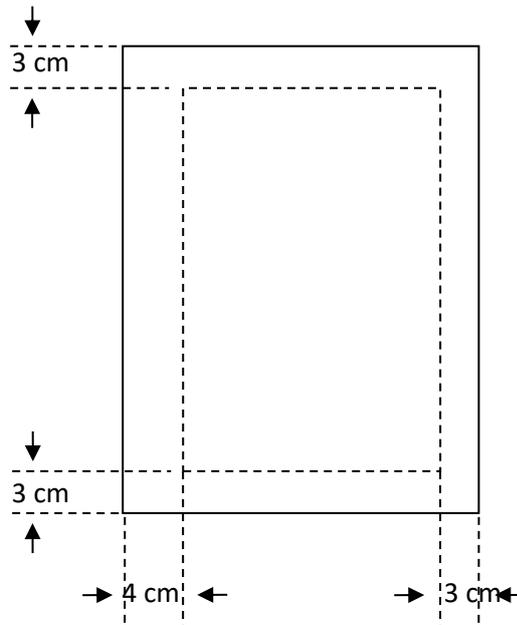
1. Mahasiswa wajib mendaftarkan diri dengan memenuhi persyaratan yang telah ditentukan.
2. Mengikuti pretest Praktikum Sistem Kendali dengan materi yang mencakup keseluruhan percobaan. Jadwal pelaksanaan pretest sebelum pelaksanaan praktikum.
3. Hasil dari pretest harus lebih dari atau sama dengan 65. Mahasiswa yang mendapatkan hasil pretest kurang dari 65, maka diwajibkan mengerjakan ulang seluruh soal-soal pretest dengan hasil tidak kurang dari 70.
4. Mahasiswa diwajibkan membaca dan memahami laporan pendahuluan yang berisi:
 - Judul Percobaan
 - Tujuan Percobaan
 - Teori Dasar
 - Alat dan Bahan
 - Rangkaian dan Percobaan
5. Mahasiswa yang telah melakukan percobaan diwajibkan untuk membuat laporan analisa praktikum serta melakukan asistensi dengan dosen yang bersangkutan, sampai laporan tersebut di Acc oleh dosen yang bersangkutan. Jika belum di Acc maka tidak dapat melakukan penjiilidan laporan secara keseluruhan.
6. Mahasiswa yang telah melakukan seluruh percobaan dan laporannya telah di Acc oleh dosen maka diwajibkan untuk menjilid seluruh Laporan tersebut dengan sampul warna yang ditentukan kemudian.
7. Batas waktu pengumpulan laporan keseluruhan akan diumumkan kemudian.

Kegiatan di atas seperti ditunjukkan pada diagram alir yang dilampirkan pada lembaran berikutnya.

Catatan: Bagi yang tidak melakukan asistensi untuk setiap percobaan tidak dapat mengumpulkan laporan akhir.

FORMAT LAPORAN PRAKTIKUM

1. Laporan ditulis tangan pada kertas putih ukuran A4.
2. Penulisan menggunakan pena warna **biru**.
3. Margin untuk penulisan laporan adalah:
Batas Kiri 4 cm, Batas Kanan 3 cm, Batas Atas 3 cm, dan Batas Bawah 3 cm.



4. Bila ada grafik dari data-data percobaan, penggambaran dilakukan pada kertas grafik (millimeter block).
5. Sampul untuk penjiilidan keseluruhan diberitahu kemudian.
6. Pada Halaman muka masing-masing percobaan diberikan sampul berwarna biru muda yang berisi : Nama, NIM, Kelompok, Logo UNJA, Tahun, dan tulisan lainnya yang dianggap perlu.
7. Pada sampul muka dituliskan kata:
 - Laboratorium Sistem Kendali dan Elektronika Lanjut
 - Program Studi Teknik Elektro
 - Universitas Jambi
 - Tahun
 - Nama
 - NIM
 - Kelompok
 - Logo UNJA



LABORATORIUM SISTEM KENDALI DAN ELEKTRONIKA LANJUT
PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JAMBI

2017

LEMBAR ASISTENSI PRAKTIKUM DASAR SISTEM KENDALI

Judul Percobaan :
Nama Praktikan (NIM) :
Nama Dosen (NIDK) :
Kelompok :

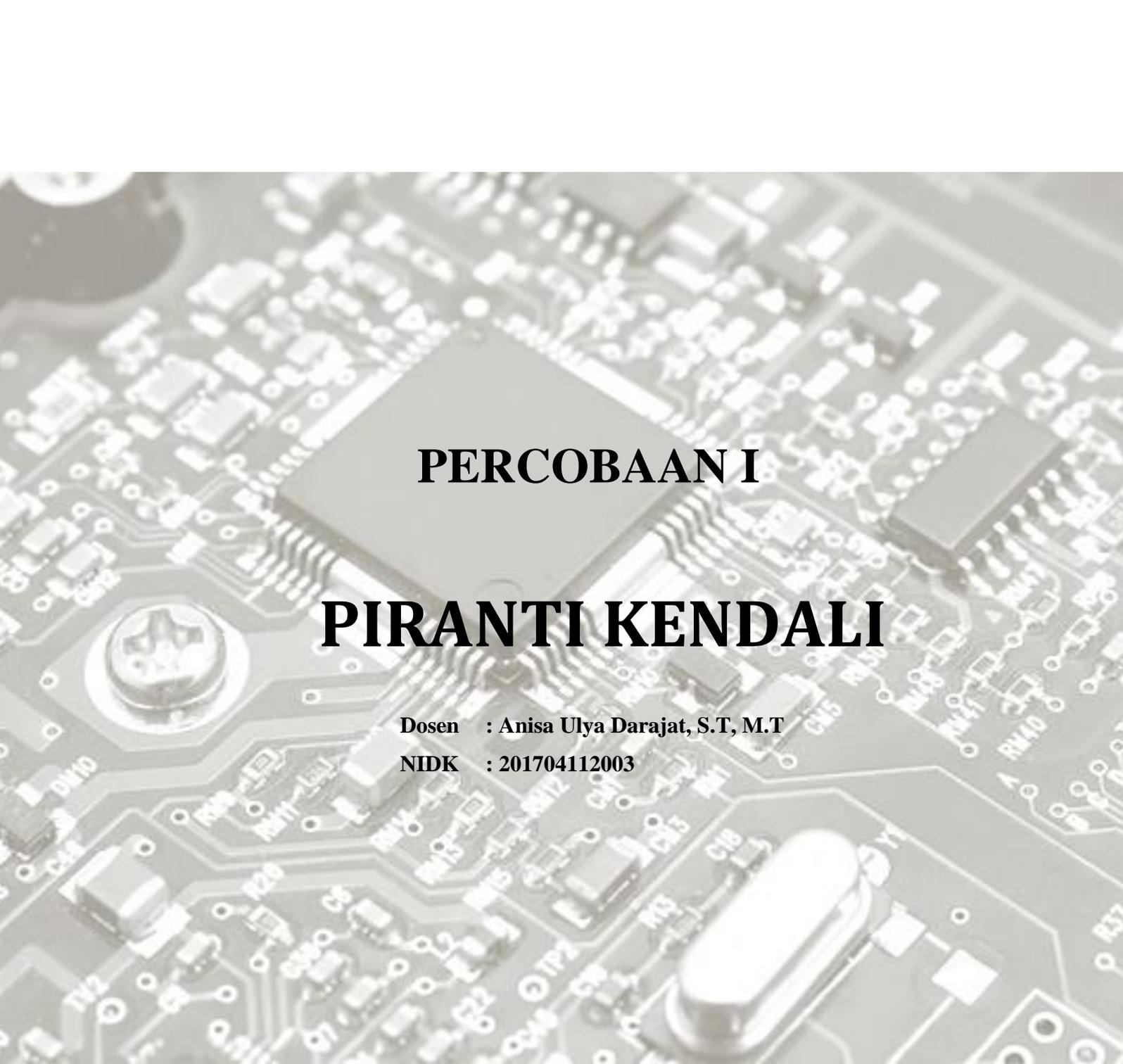
No	Catatan	Tanggal	TTD

Jambi, September 2017
Dosen Penanggung Jawab,

NIDK.

DAFTAR ISI

TATA TERTIB PRAKTIKUM	i
DIAGRAM ALIR PELAKSANAAN PRAKTIKUM	ii
PROSES PELAKSANAAN PRAKTIKUM SISTEM KENDALI.....	iii
FORMAT LAPORAN PRAKTIKUM	iv
LEMBAR ASISTENSI PRAKTIKUM	v
DAFTAR ISI.....	vi
PERCOBAAN I	
PIRANTI KENDALI	1
PERCOBAAN II	
SISTEM KENDALI <i>PROPORTIONAL (P)</i>	9
PERCOBAAN III	
SISTEM KENDALI <i>PROPORTIONAL DERIVATIVE (PD)</i>	17
PERCOBAAN IV	
SISTEM KENDALI <i>PROPORTIONAL INTEGRAL (PI) CONTROL</i>	25
PERCOBAAN V	
SISTEM KENDALI <i>PROPORTIONAL INTEGRAL DERIVATIVE (PID) CONTROL</i>	34



PERCOBAAN I

PIRANTI KENDALI

Dosen : Anisa Ulya Darajat, S.T, M.T

NIDK : 201704112003

I. JUDUL PERCOBAAN

Piranti Kendali

II. TUJUAN

1. Memahami pemrograman piranti Mikrokontroler.
2. Memahami proses konversi analog ke digital dengan Mikrokontroler.
3. Memahami input output pada sistem kendali.
4. Mampu Mengatur tegangan listrik untuk mengendalikan putaran roda dengan menggunakan PWM.

III. TEORI DASAR

3.1 Mikrokontroler / Arduino UNO

Arduino Uno adalah board mikrokontroler berbasis ATmega328. Uno memiliki 14 pin digital input / output (dimana 6 dapat digunakan sebagai output PWM), 6 input analog, resonator keramik 16 MHz, koneksi USB, jack listrik, header ICSP, dan tombol reset. Uno dibangun berdasarkan apa yang diperlukan untuk mendukung mikrokontroler, sumber daya bisa menggunakan power USB (jika terhubung ke komputer dengan kabel USB) dan juga dengan adaptor atau baterai.

Arduino Uno berbeda dari semua board sebelumnya yang dalam hal ini tidak menggunakan FTDI chip driver USB-to-serial. Sebaliknya, fitur Atmega16U2 (Atmega8U2 sampai versi R2) diprogram sebagai konverter USB-to-serial. Revisi 2 dari Uno memiliki resistor pulling 8U2 HWB yang terhubung ke ground, sehingga lebih mudah untuk menggunakan mode DFU.



Gambar 1.1 Arduino UNO

Tapi tidak semua pin Arduino dapat digunakan untuk mengolah signal analog. Pada board Arduino Uno, terdapat enam pin analog, yakni mulai dari A0 hingga A5. Huruf A pada awal nama pin Arduino menandakan pin tersebut dapat digunakan untuk mengolah signal analog. Seberapa tepat nilai signal analog yang dipetakan secara digital, ditentukan oleh seberapa besar resolusi ADC. Semakin besar resolusi ADC, maka semakin mendekati nilai analog dari signal tersebut. Untuk resolusi ADC pada board Arduino Uno ialah 10 bit, yang berarti mampu memetakan hingga 1024 discrete analog level. Beberapa jenis microcontroller lain memiliki resolusi 8 bit, 256 discrete analog level, bahkan ada yang memiliki resolusi 16 bit, 65536 discrete analog level.

3.2 Analog-to-Digital Converter (ADC)

Analog-to-Digital Converter (ADC) adalah sebuah piranti yang dirancang untuk mengubah sinyal-sinyal besaran analog menjadi bentuk sinyal digital. Nilai ADC menunjukkan ratio perbandingan dengan tegangan yang terbaca. Berikut persamaannya ialah nilai ADC terukur ialah nilai ADC maximum dikalikan tegangan terbaca, kemudian dibagi dengan nilai tegangan sumber. Nilai ADC tergantung dengan tegangan yang menjadi catu daya sistem mikrokontroler. Untuk board Arduino biasa menggunakan sumber tegangan 5 volt. Cara mencari nilai ADC, dengan menerapkan persamaan yang ada dan tegangan terbaca sebesar 2,12 volt pada board Arduino Uno.

- Board Arduino Uno memiliki resolusi 10 bit, dengan nilai terbesar 1023.
- Tegangan sumber 5 volt dan tegangan terbaca ialah 2,12 volt.

- Nilai ADC terukur ialah nilai ADC maximum dikalikan tegangan terbaca, kemudian dibagi dengan nilai tegangan sumber Sehingga diperoleh nilai ADC sebesar 434.

Dengan demikian diperoleh nilai ADC sebesar 434 dari tegangan terukur 2,12 volt. Untuk setiap unit ADC tersebut memiliki perbandingan tegangan sebesar 4,9 mV. Selain dipengaruhi oleh besarnya nilai resolusi ADC, tepat tidaknya Pengukuran nilai ADC juga dipengaruhi oleh *clock speed* ADC tersebut. Untuk board Arduino Uno sendiri *clock speed* ADC maximum yang disarankan ialah 200 kHz. Nilai *clock speed* 200 kHz tersebut berdasarkan spesifikasi internal DAC (Digital to Analog Converter) pada rangkaian pengubahnya. Meski demikian, penggunaan *clock speed* pada 1 MHz misalnya, tidak mengurangi kualitas resolusi ADC tersebut.

3.3 Motor Arus Searah (DC)



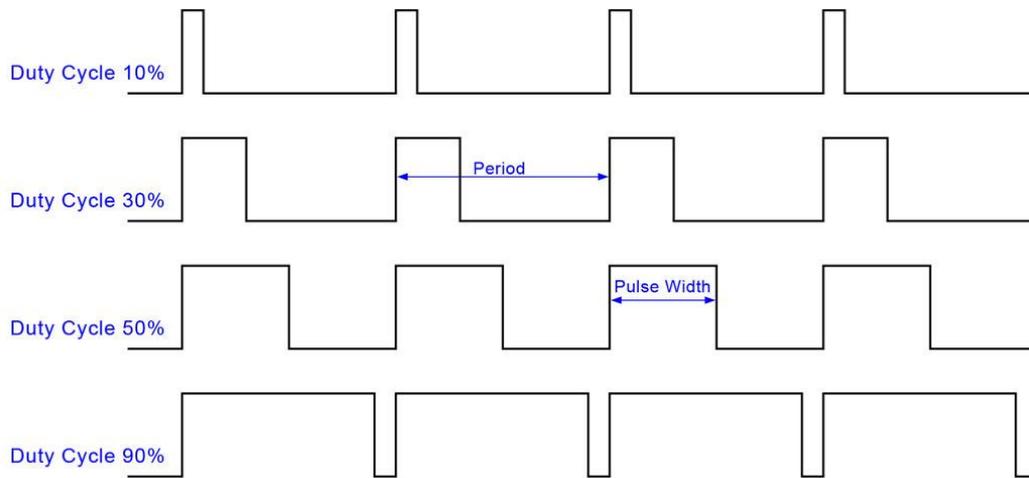
Gambar 1.2 Motor Arus Searah (DC)

Motor DC memiliki dua bagian penting. Bagian pertama adalah stator, yaitu bagian yang tidak berputar dan bagian kedua adalah rotor, yaitu bagian yang berputar. Di rotor inilah, poros diletakan. Diujung poros dapat dipasang objek (misalnya puli) yang ingin diputar.

3.4 PWM

PWM singkatan dari *Pulse Width Modulation*. Pada mikrokontroler, sinyal PWM beroperasi pada frekuensi rendah 500Hz – 2KHz. Pada board arduino, pin yang bisa dimanfaatkan untuk PWM adalah pin yang diberi tanda tilde (~), yaitu pin 3, 5, 6,

9, 10, dan pin 11. Pin-pin tersebut merupakan pin yang bisa difungsikan untuk menghasilkan sinyal PWM.



Gambar 1.3. Sinyal PWM

IV. TUGAS PENDAHULUAN

1. Cari Teori dasar tentang ADC dan PWM pada mikrokontroler minimal 1 halaman diketik dengan bahasa sendiri.

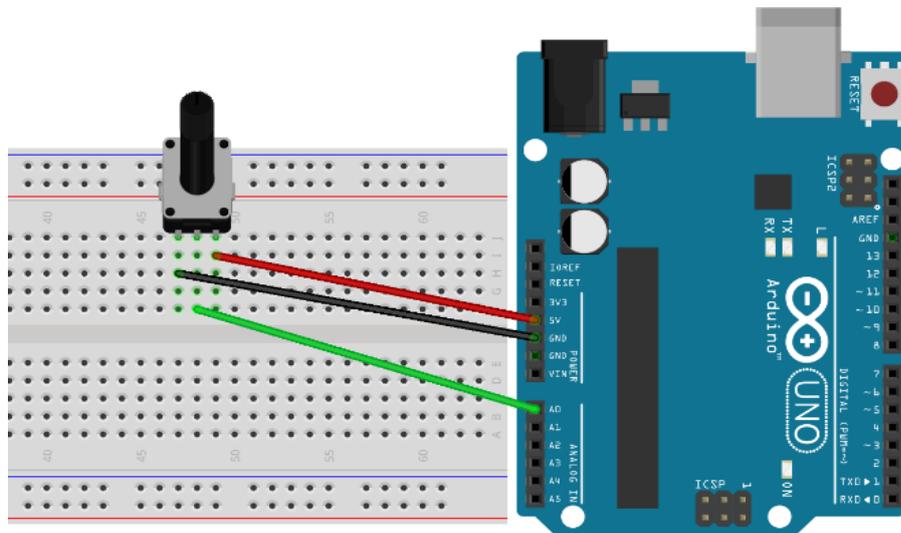
V. ALAT DAN BAHAN

Peralatan yang digunakan pada percobaan ini sebagai berikut:

1. Mikrokontroler Arduino
2. Multimeter digital
3. Kabel USB
4. Potensiometer
5. Osiloskop
6. Motor DC
7. Motor Driver L298N
8. Perangkat Lunak Arduino IDE

VI. RANGKAIAN DAN PROSEDUR PERCOBAAN

A. Konversi sinyal analog ke sinyal Digital (ADC)



Gambar 1.4 Rangkaian Kerja ADC

1. Hubungkan potensio ke Arduino seperti pada Gambar 1.4.
2. Pasang USB ke Arduino dan hubungkan ke computer.
3. Buka perangkat lunak Arduino IDE.
4. Ketikkan kode program berikut ini:

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  int hasil_adc = analogRead(A0);  
  Serial.print("NILAI ADC = ");  
  Serial.println(hasil_adc);  
  double adc_ke_v = hasil_adc*5.0/1024;  
  Serial.print("NILAI Vadc = ");  
  Serial.print(adc_ke_v);  
  Serial.println("Volt");  
  delay(200);  
}
```

5. Pilih “**Arduino Genuino Uno**” pada menu “**Tools => Board**”
6. Pilih “**Com (Arduino/Uno)**” pada menu “**Tools => Port**”
7. Compile dan upload kode program ke Arduino.

8. Atur tegangan keluaran Potensiometer seperti pada Tabel 1.1.
9. Catat Keluaran ADC yang dihasilkan pada layar terminal arduino IDE seperti pada Tabel 1.1.
10. Hitung *error* hasil pengukuran antara nilai tegangan pada potensiometer dengan hasil pengukuran dengan ADC dengan rumus:

$$\% \text{ error} = \frac{V_{pot} - V_{adc}}{V_{pot}} \times 100 \%$$

Tabel 1.1

No	Nilai Tegangan (V_{pot})	Nilai ADC	Nilai Konversi ADC ke Tegangan (V_{adc})	<i>Error Hasil Pengukuran (%)</i>
1.	0.2 Volt			
2.	1 Volt			
3.	2 Volt			
4.	2.5 Volt			
5.	3.5 Volt			
6.	4 Volt			
7.	5 Volt			

B. Kendali Kecepatan Putaran Motor dengan PWM

1. Hubungkan Arduino, motor driver dan Motor DC seperti pada Gambar 1.5.
2. Ketik program berikut ini kemudian upload ke Arduino.

```

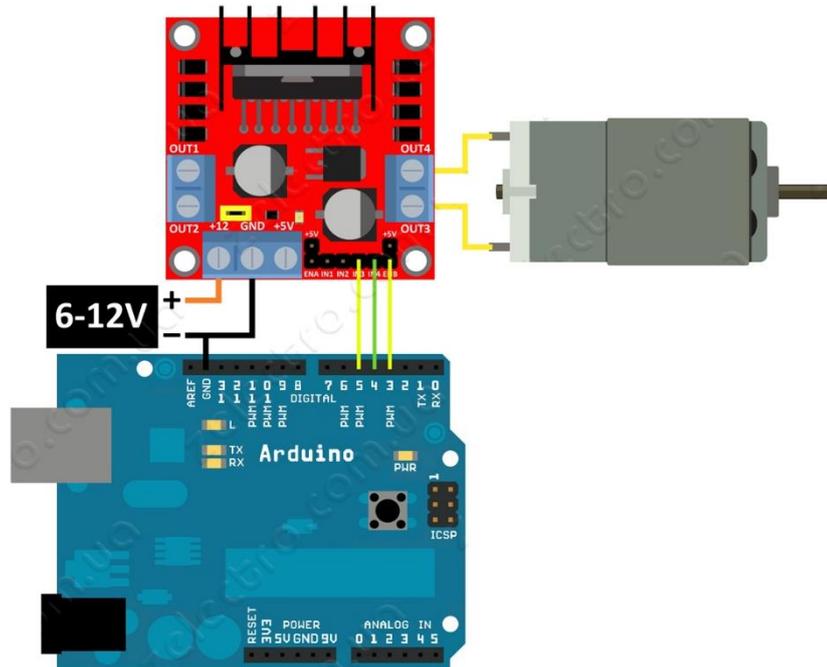
void setup() {
  pinMode(8,OUTPUT);
  pinMode(9,OUTPUT);
  pinMode(10,OUTPUT);
}

void loop() {
  digitalWrite(5, HIGH); // mengatur arah putaran motor
  digitalWrite(4, LOW); // mengatur arah putaran motor
  analogWrite(3, nilai_pwm); //ubah nilai pwm sesuai tabel
  delay(2); //delay dalam ms
}

```

3. Hubungkan Channel 1 Osiloskop dan multimeter dengan keluaran Pin #3 Arduino.

- Hubungkan Channel 2 Osiloskop dan multimeter dengan keluaran Output 4 atau 5 motor driver.



Gambar 1.5 Rangkaian Pengendali Kecepatan Motor DC

- Catat dan ambil gambar nilai yang terbaca pada multimeter dan osiloskop pada Tabel 1.2.

Tabel 1.2

No	Nilai PWM	V_{ideal}	$V_{multimeter}$	Lebar Pulsa Perhitungan	Lebar Pulsa Pengukuran	$Error V$ (%)	$Error LP$ (%)
1.	10						
2.	50						
3.	100						
4.	150						
5.	200						
6.	255						

PERCOBAAN II

SISTEM KENDALI *PROPORTIONAL* (P)

Dosen : Anisa Ulya Darajat, S.T, M.T

NIDK : 201704112003

I. JUDUL PERCOBAAN

SISTEM KENDALI PROPORSIONAL

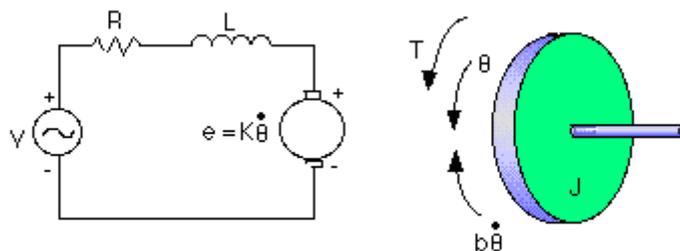
II. TUJUAN

1. Agar mahasiswa akan lebih mengenal tentang prinsip kerja kontrol proporsional.
2. Mampu menerangkan pengaruh kontroler P pada kinerja sistem pengaturan posisi motor dc.
3. Mampu melakukan analisis kinerja sistem pengaturan posisi motor arus searah.
4. Mampu menerangkan pengaruh kecepatan pada kinerja suatu sistem pengaturan posisi.

III. TEORI DASAR

3.1 Motor DC

Motor DC bekerja berdasarkan prinsip gaya elektromagnetik sehingga apabila motor tersebut diberi catu daya, arus akan mengalir ke dalam motor kemudian menghasilkan torsi putar yang sebanding dengan arus tersebut. Pemodelan Rangkaian internal Motor DC secara sederhana dan analisisnya adalah sebagai berikut:



Gambar 2.1. Motor DC

Persamaan torsi yang dibangkitkan oleh Motor DC dapat didekati secara linear menurut persamaan berikut ini:

$$T = K_a i \dots\dots\dots (1)$$

dimana K_a adalah konstanta jangkar motor yang bergantung pada banyaknya lilitan pada jangkar, jumlah kutub medan, tipe belitan dan penampang jangkarnya. Adapun besarnya tegangan ggl induksi lawan yang dibangkitkan motor ketika berputar adalah sebanding dengan konstanta motor K_b dan kecepatan sudut putaran motor $\dot{\theta}$ atau turunan pertama dari posisi sudut motor (θ):

$$e = K_b \dot{\theta} \dots\dots\dots (2)$$

Dengan menggunakan hukum newton, bahwa persamaan torsi yang terkait dengan momen inersia dan rasio redaman dari motor adalah:

$$T = J\ddot{\theta} + b\dot{\theta} \dots\dots\dots (3)$$

dari persamaan (1) dan (3) diperoleh:

$$i = \frac{J\ddot{\theta} + b\dot{\theta}}{K_a} \dots\dots\dots (4)$$

Sedangkan besarnya tegangan V menurut hukum kirchoff adalah:

$$V = iR + L \frac{di}{dt} + e \dots\dots\dots (5)$$

$$V = iR + L \frac{di}{dt} + K_b \dot{\theta} \dots\dots\dots (6)$$

Dengan me-laplace-kan persamaan (4) dan (6) dan mensubstitusikannya, maka diperoleh fungsi transfer antara posisi sudut motor q terhadap tegangan armature V dimana $K_a = K_b$

$$\frac{\theta(s)}{V(s)} = \frac{K}{s((J_s+b)(L_s+R)+K^2)} \dots\dots\dots (7)$$

Persamaan (7) diatas memiliki 5 konstanta yang belum diketahui. Nilai-nilai tersebut pada dasarnya bisa diperoleh dari percobaan identifikasi plant. Pada praktikum saat ini tidak melakukan proses identifikasi tersebut. Tetapi langsung menggunakan konstanta yang biasa dipakai dalam analisis motor DC. Adapun konstanta tersebut adalah sebagai berikut:

1. moment inertia rotor (J) = $0.1 \text{kg} \cdot \frac{\text{m}^2}{\text{s}^2}$
2. damping ratio of the mechanical system (b) = 0.01Ns/m

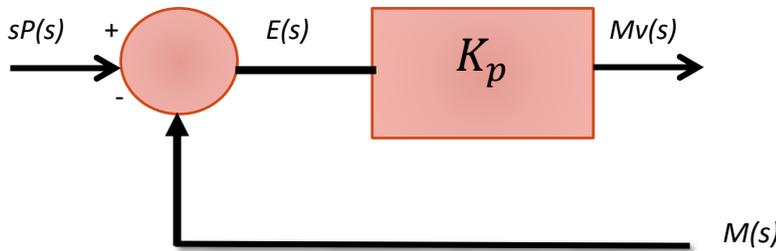
3. electromotive force constant ($K=Kb=Ka$) = 0.3 Nm/Amp
4. electric resistance (R) = 2 ohm
5. electric inductance (L) = 0.1 H

sehingga diperoleh fungsi transfer sebagai berikut:

$$\frac{\theta(s)}{V(s)} = \frac{30}{s^3+2.1s^2+11s} \dots\dots\dots (8)$$

3.2 Proportional Control

Proportional control adalah gain yang digunakan untuk mengatur perilaku dari sinyal error terhadap input dari sistem. 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:



Gambar 2.2 Blok Pengendali *Proportional*

Pada kontrol proporsional, besar sinyal output untuk elemen kontrol final sebanding secara proporsional dengan besar error *set point* yang muncul. Hubungan ini dapat dirumuskan sebagai berikut:

$$P = (K_p)E$$

dengan:

P = Output

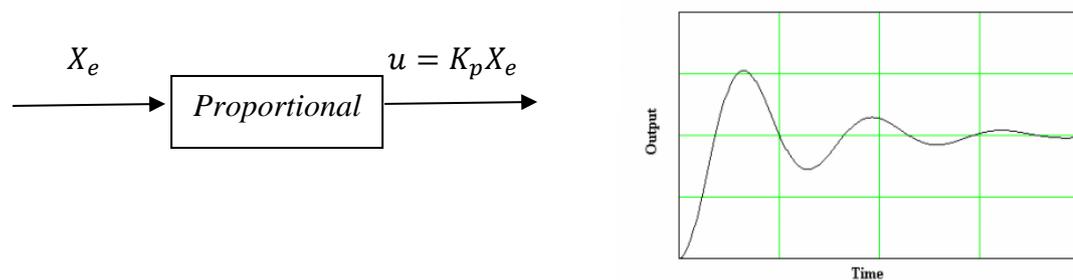
$E = Error$ (deviasi *set point*)

$K_p =$ Konstanta perkalian proporsional

Sistem kontrol proporsional ini menghasilkan hubungan yang linier antara error *set point* dengan sinyal output yang akan menjadi aksi pada elemen kontrol akhir. Konstanta perkalian proporsional dapat diubah nilainya sesuai dengan kebutuhan. Proses kontrol ini menghasilkan aksi yang cepat pada setiap perubahan *error set point* yang terjadi.

Proporsional digunakan untuk mengatur *speed* dari sistem. Nilai gain yang semakin besar dapat menghasilkan overshoot dari sistem. Karakteristik Proporsional controller:

- a. mereduksi risetime
- b. tidak pernah mengeliminasi *steady-state error*



Gambar 2.3. *Proportional Control*

Penggunaan mode kontrol proporsional harus memperhatikan hal – hal berikut:

- a. Jika nilai K_p kecil, mode kontrol proporsional hanya mampu melakukan koreksi kesalahan yang kecil, sehingga akan menghasilkan respon sistem yang lambat.
- b. Jika nilai K_p dinaikkan, respon sistem menunjukkan semakin cepat mencapai keadaan stabilnya.
- c. 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 $U = Kc \cdot e$ dimana ; $Kc =$ gain kontrol proporsional

IV. TUGAS PENDAHULUAN

- Apa yang dimaksud *rise time*, *overshoot*, *settling time*, *steady state* dan *error*?
- Cari Literatur tentang Kontroler *Proportional* pada kendali posisi motor dc dan kemudian simulasikan pada matlab simulink.

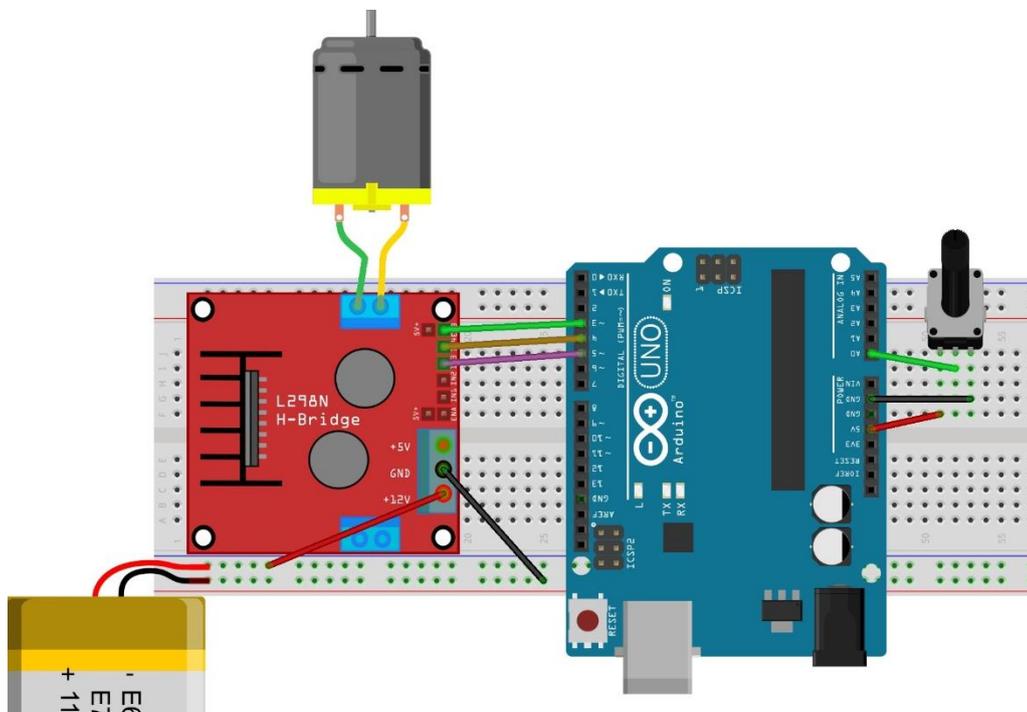
V. ALAT DAN BAHAN

Adapun alat dan bahan yang digunakan adalah sebagai berikut:

1. Motor DC
2. Motor Driver L298N
3. Mikrokontroler Arduino Uno
4. Osiloscope

VI. RANGKAIAN DAN PROSEDUR PERCOBAAN

1. Hubungkan Motor Driver L298N seperti pada Gambar 2.4.



Gambar 2.4 Rangkaian Kerja

2. Ketik kode program pada Tabel 2.1.
3. Upload program ke Arduino uno.
4. Buka jendela “**Serial Ploter**” pada menu “**Tools**”

Tabel 2.1

```

int error, sudut_ref,motor_pwm;
double sudut_aktual, P,nilai_PID;
const int Toleransi = 5;
const double Kp=0.5;

void setup() {
  Serial.begin(115200);
  pinMode(8,OUTPUT);
  pinMode(9,OUTPUT);
  pinMode(10,OUTPUT);
  sudut_ref = 90;
  double Kp=0.5;
}

void loop() {
  int ADCx = analogRead(A0);
  sudut_aktual = map(ADCx, 240, 860, 0, 180);
  error = sudut_ref - sudut_aktual;
  if(abs(error)<=Toleransi) error = 0;
  P = Kp * error;
  nilai_PID = P;

  if(nilai_PID < 0)
  {
    motor_pwm = constrain(abs(nilai_PID), 1, 30);
    motor_pwm = motor_pwm+35;
    digitalWrite(9, HIGH);
    digitalWrite(8, LOW);
    analogWrite(10, motor_pwm);
  }
  else if(nilai_PID > 0)
  {
    motor_pwm = constrain(abs(nilai_PID), 1, 30);
    motor_pwm = motor_pwm+35;
    digitalWrite(8, HIGH);
    digitalWrite(9, LOW);
    analogWrite(10, motor_pwm);
  }
  else // jika nilai_PID = 0
  {
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
    analogWrite(10, 0); // motor berhenti
  }

  Serial.print(sudut_aktual);
  Serial.print(" ");
  Serial.println(sudut_ref);

```

}

5. Ganti nilai K_p sesuai Tabel 2.1 dan kemudian ambil gambar grafik respon Sistem pada **serial Plotter**.
6. Catat *Overshoot*, *Rise time* dan Waktu *steady state* sistem pada table.

Tabel 2.1

No	Nilai Gain K_p	<i>Maximum Overshoot</i> (%)	<i>Rise Time</i> (s)	<i>Steady State Time</i> (s)
1.	0.1			
2.	0.3			
3.	0.5			
4.	0.75			
5.	1			
6.	1.25			
7.	1.5			

PERCOBAAN III

SISTEM KENDALI *PROPORTIONAL DERIVATIVE* (PD)

Dosen : Swadexi Istiqphara, S.T, M.T

NIDK : 201511111010

I. JUDUL PERCOBAAN

SISTEM KENDALI *PROPORTIONAL DERIVATIVE* (PD)

II. TUJUAN PERCOBAAN

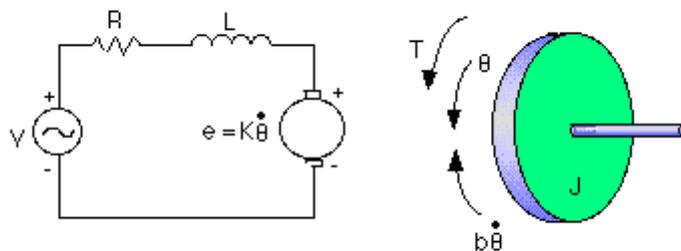
Praktikum ini memiliki tujuan, yaitu:

1. Agar mahasiswa akan lebih mengenal tentang prinsip kerja kontrol proporsional *derivative*.
2. Mampu menerangkan pengaruh kontroler PD pada kinerja sistem pengaturan posisi.
3. Mampu melakukan analisis kinerja sistem pengaturan posisi motor arus searah.
4. Mampu menerangkan pengaruh kecepatan pada kinerja suatu sistem pengaturan posisi.

III. TEORI DASAR

3.1 Motor DC

Motor DC bekerja berdasarkan prinsip gaya elektromagnetik sehingga apabila motor tersebut diberi catu daya, arus akan mengalir ke dalam motor kemudian menghasilkan torsi putar yang sebanding dengan arus tersebut. Pemodelan Rangkaian internal Motor DC secara sederhana dan analisisnya adalah sebagai berikut:



Gambar 3.1. Motor DC

Persamaan torsi yang dibangkitkan oleh Motor DC dapat didekati secara linear menurut persamaan berikut ini:

$$T = K_a i \dots\dots\dots (1)$$

dimana K_a adalah konstanta jangkar motor yang bergantung pada banyaknya lilitan pada jangkar, jumlah kutub medan, tipe belitan dan penampang jangkarnya. Adapun besarnya tegangan ggl induksi lawan yang dibangkitkan motor ketika berputar adalah sebanding dengan konstanta motor K_b dan kecepatan sudut putaran motor $\dot{\theta}$ atau turunan pertama dari posisi sudut motor (θ):

$$e = K_b \dot{\theta} \dots\dots\dots (2)$$

Dengan menggunakan hukum newton, bahwa persamaan torsi yang terkait dengan momen inersia dan rasio redaman dari motor adalah:

$$T = J\ddot{\theta} + b\dot{\theta} \dots\dots\dots (3)$$

dari persamaan (1) dan (3) diperoleh:

$$i = \frac{J\ddot{\theta} + b\dot{\theta}}{K_a} \dots\dots\dots (4)$$

Sedangkan besarnya tegangan V menurut hukum kirchoff adalah:

$$V = iR + L \frac{di}{dt} + e \dots\dots\dots (5)$$

$$V = iR + L \frac{di}{dt} + K_b \dot{\theta} \dots\dots\dots (6)$$

Dengan me-laplace-kan persamaan (4) dan (6) dan mensubstitusikannya, maka diperoleh fungsi transfer antara posisi sudut motor q terhadap tegangan armature V dimana $K_a = K_b$

$$\frac{\theta(s)}{V(s)} = \frac{K}{s((J_s+b)(L_s+R)+K^2)} \dots\dots\dots (7)$$

Persamaan (7) diatas memiliki 5 konstanta yang belum diketahui. Nilai-nilai tersebut pada dasarnya bisa diperoleh dari percobaan identifikasi plant. Pada praktikum saat ini tidak melakukan proses identifikasi tersebut. Tetapi langsung menggunakan konstanta yang biasa dipakai dalam analisis motor DC. Adapun konstanta tersebut adalah sebagai berikut:

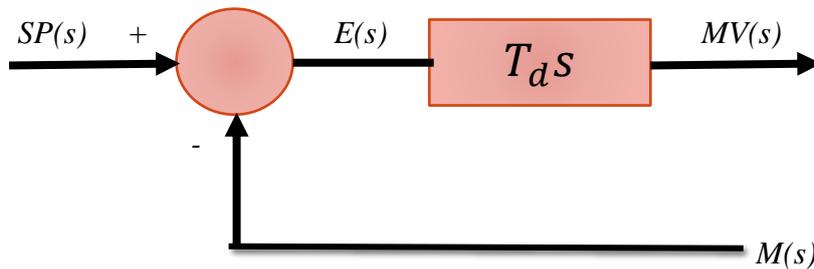
1. Moment of inertia of the rotor (J) = $0.1 \text{ kg} \cdot \frac{\text{m}^2}{\text{s}^2}$.
2. Damping ratio of the mechanical system (b) = 0.01 Ns/m .
3. Electromotive force constant ($K=Kb=Ka$) = 0.3 Nm/Amp .
4. Electric resistance (R) = 2 ohm .
5. Electric inductance (L) = 0.1 H .

sehingga diperoleh fungsi transfer sebagai berikut:

$$\frac{\theta(s)}{V(s)} = \frac{30}{s^3 + 2.1s^2 + 11s} \dots\dots\dots (8)$$

3.2 Sistem kendali *Proportional Derivative*

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 diferensial ditunjukkan oleh gambar berikut:



Gambar 3.2 Blok Pengendali Diferensial

Kontrol derivatif juga dikenal dengan sebutan *rate control*. Sistem kontrol menghasilkan output pada saat terjadi error pada *set point*, dan nilai output tersebut proporsional dengan laju (*rate*) perubahan dari error:

$$P = (T_d) \frac{dE}{dT}$$

dimana T_d adalah konstanta waktu laju (derivatif).

Pada kontrol derivatif, semakin cepat perubahan error *set point* yang terjadi, semakin besar nilai output kontrol yang keluar. Nilai output akan langsung menjadi nol apabila nilai error sama dengan nol. Nilai dari konstanta T_d juga dapat diubah

untuk menambah atau mengurangi sensitifitas sistem kontrol. Kontrol ini digunakan untuk menghasilkan respon yang cepat terhadap setiap perubahan error yang terjadi.

Karakteristik dari controller *differensial* adalah sebagai berikut:

- a. Kontroler ini tidak dapat menghasilkan keluaran bila tidak ada perubahan atau *error* sebagai sinyal kesalahan untuk masukannya.
- b. Jika sinyal error berubah terhadap waktu, maka keluaran yang dihasilkan controller tergantung pada nilai T_d dan laju perubahan sinyal kesalahan.
- c. 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:

$$U = K_c \cdot T_D \frac{de}{dt} \dots\dots\dots (9)$$

dimana:

$K_c = \text{gain}$

$e = \text{error}$

$T_D = \text{derivative time}$

Aksi dari derivatif control berguna untuk meningkatkan damping sistem, memungkinkan respon yang cepat tanpa adanya overshoot. Persamaan derivatif control adalah:

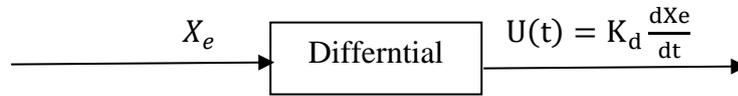
$$U(t) = K_d \frac{dx_e}{dt} = (K_p T_d) \frac{dx_e}{dt} \text{ bentuk Laplace Transformnya } U(s) \\ = (K_p T_d) s X_e \dots\dots\dots (10)$$

dimana K_p adalah konstanta proporsional dan T_d adalah differential time constant.

Karakteristik dari derivatif control adalah sebagai berikut:

- Meningkatkan Kestabilan dari sistem

- Mereduksi overshoot
- Meningkatkan respons transient



Gambar 3.3 Derivatif Control

IV. TUGAS PENDAHULUAN

- Cari Literatur tentang Kontroler *Proportional Derivative* pada kendali posisi motor dc dan kemudian simulasikan pada matlab simulink.

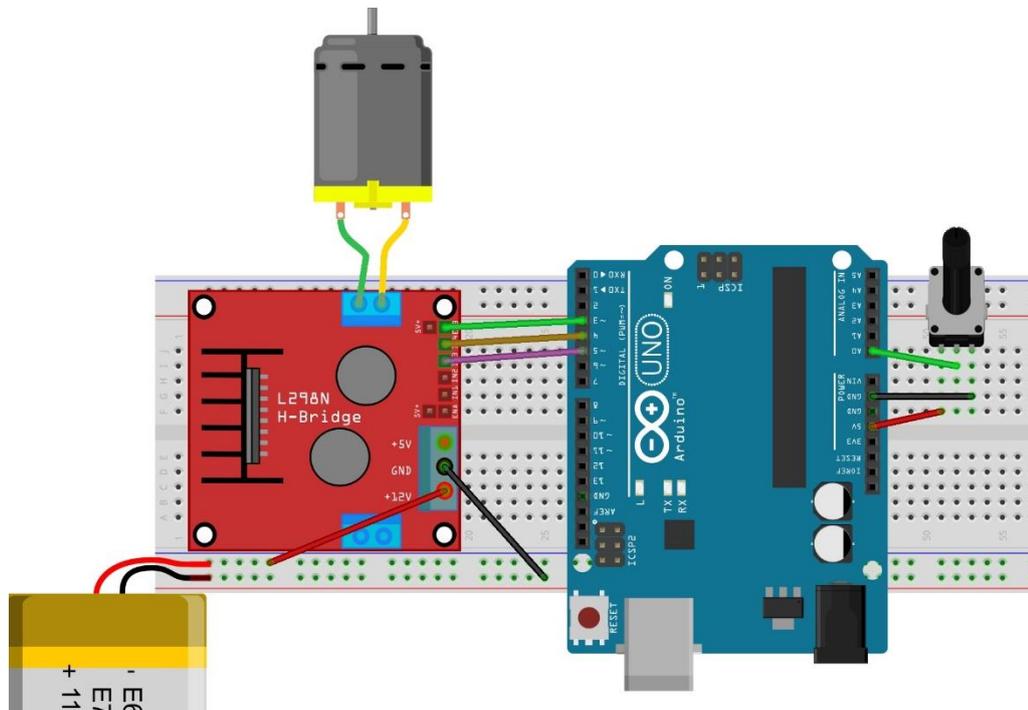
V. ALAT DAN BAHAN

Adapun alat dan bahan yang digunakan adalah sebagai berikut:

1. Motor DC
2. Arduino UNO
3. Osiloscope

VI. RANGKAIAN DAN PROSEDUR PERCOBAAN

1. Hubungkan Motor Driver L298N seperti pada Gambar 3.4.



Gambar 3.4 Rangkaian Kerja

2. Ketik kode program pada Tabel 3.1.
3. Upload program ke Arduino uno.
4. Buka jendela “**Serial Ploter**” pada menu “**Tools**”

Tabel 3.1

```

int error, sudut_ref, motor_pwm, error_sebelumnya;
double sudut_aktual, P, D, nilai_PID, delta_error;
const int Toleransi = 5;
const double Kp=0.5, Kd=0.5;
void setup() {
  Serial.begin(115200);
  pinMode(8,OUTPUT);
  pinMode(9,OUTPUT);
  pinMode(10,OUTPUT);
  sudut_ref = 90;
}

void loop() {
  int ADCx = analogRead(A0);
  sudut_aktual = map(ADCx, 240, 860, 0, 180); // Konversi nilai ADC ke sudut putar
  error = sudut_ref - sudut_aktual;
  if(abs(error)<=Toleransi) error = 0; // toleransi error
  delta_error = error - error_sebelumnya;
  P = Kp * error;
  D = Kd*delta_error;
  nilai_PID = P+D;
  if(nilai_PID < 0)
  {
    motor_pwm = constrain(abs(nilai_PID), 1, 30); // batasi nilai PID tidak lebih dari 30
    motor_pwm = motor_pwm+35;
    digitalWrite(9, HIGH);
    digitalWrite(8, LOW);
    analogWrite(10, motor_pwm);
  }
  else if(nilai_PID > 0)
  {
    motor_pwm = constrain(abs(nilai_PID), 1, 30);
  }
}

```

```

motor_pwm = motor_pwm+35;
digitalWrite(8, HIGH);
digitalWrite(9, LOW);
analogWrite(10, motor_pwm);
}
else // jika nilai_PID = 0
{
digitalWrite(8, LOW);
digitalWrite(9, LOW);
analogWrite(10, 0); // motor berhenti
}
Serial.print(sudut_aktual);
Serial.print(" ");
Serial.println(sudut_ref);

error_sebelumnya = error;
}

```

5. Ganti nilai K_p dan K_d sesuai Tabel 3.2 dan kemudian ambil gambar grafik respon Sistem pada **serial Plotter**.
6. Catat *Overshoot*, *Rise time* dan Waktu *steady state* sistem pada tabel.

Tabel 3.2

No	K_p	K_d	<i>Max Overshoot</i>	<i>Rise Time (s)</i>	<i>Steady State(s)</i>
1.	0.1	0.1			
2.	0.3	0.5			
3.	0.5	1			
4.	0.75	2			
5.	1	1			
6.	1.25	0.5			
7.	1.5	0.1			

VII. MATERI PEMBAHASAN

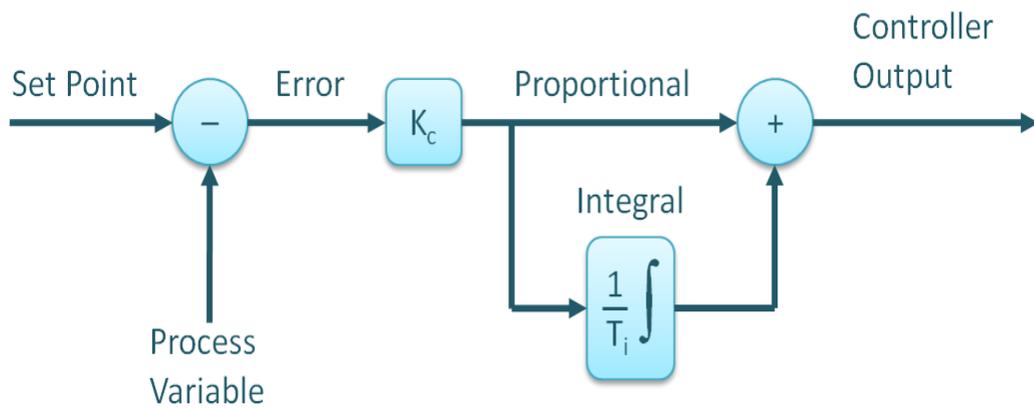
1. Analisa Grafik.
2. Apa pengaruh K_p , K_d pada *rise time*, *overshoot*, *settling time*, *steady state*?

PERCOBAAN IV

SISTEM KENDALI PROPORTIONAL INTEGRAL (PI)

Dosen : Swadexi Istiqphara, S.T, M.T

NIDK : 201511111010



I. JUDUL PERCOBAAN

SISTEM KENDALI *PROPORTIONAL INTEGRAL* (PI)

II. TUJUAN PERCOBAAN

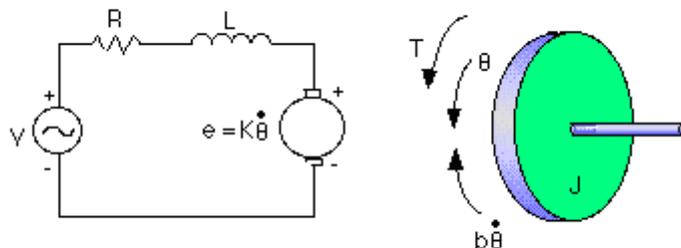
Adapun tujuan dari praktikum ini:

1. Agar mahasiswa akan lebih mengenal tentang prinsip kerja kontrol proporsional *integral*.
2. Mampu menerangkan pengaruh kontroler PI pada kinerja sistem pengaturan posisi.
3. Mampu melakukan analisis kinerja sistem pengaturan posisi motor arus searah.
4. Mampu menerangkan pengaruh kecepatan pada kinerja suatu sistem pengaturan posisi.

III. TEORI SINGKAT

3.1 Motor DC

Motor DC bekerja berdasarkan prinsip gaya elektromagnetik sehingga apabila motor tersebut diberi catu daya, arus akan mengalir ke dalam motor kemudian menghasilkan torsi putar yang sebanding dengan arus tersebut. Pemodelan Rangkaian internal Motor DC secara sederhana dan analisisnya adalah sebagai berikut:



Gambar 4.1. Motor DC

Persamaan torsi yang dibangkitkan oleh Motor DC dapat didekati secara linear menurut persamaan berikut ini:

$$T = K_a i \dots\dots\dots (1)$$

dimana K_a adalah konstanta jangkar motor yang bergantung pada banyaknya lilitan pada jangkar, jumlah kutub medan, tipe belitan dan penampang jangkarnya. Adapun besarnya tegangan ggl induksi lawan yang dibangkitkan motor ketika berputar adalah sebanding dengan konstanta motor K_b dan kecepatan sudut putaran motor $\dot{\theta}$ atau turunan pertama dari posisi sudut motor (θ):

$$e = K_b \dot{\theta} \dots\dots\dots (2)$$

Dengan menggunakan hukum newton, bahwa persamaan torsi yang terkait dengan momen inersia dan rasio redaman dari motor adalah:

$$T = J\ddot{\theta} + b\dot{\theta} \dots\dots\dots (3)$$

dari persamaan (1) dan (3) diperoleh:

$$i = \frac{J\ddot{\theta} + b\dot{\theta}}{K_a} \dots\dots\dots (4)$$

Sedangkan besarnya tegangan V menurut hukum kirchoff adalah:

$$V = iR + L \frac{di}{dt} + e \dots\dots\dots (5)$$

$$V = iR + L \frac{di}{dt} + K_b \dot{\theta} \dots\dots\dots (6)$$

Dengan me-laplace-kan persamaan (4) dan (6) dan mensubstitusikannya, maka diperoleh fungsi transfer antara posisi sudut motor q terhadap tegangan armature

V dimana $K_a = K_b$

$$\frac{\theta(s)}{V(s)} = \frac{K}{s((J_s+b)(L_s+R)+K^2)} \dots\dots\dots (7)$$

Persamaan (7) diatas memiliki 5 konstanta yang belum diketahui. Nilai-nilai tersebut pada dasarnya bisa diperoleh dari percobaan identifikasi plant. Pada praktikum saat ini tidak melakukan proses identifikasi tersebut. Tetapi langsung

menggunakan konstanta yang biasa dipakai dalam analisis motor DC. Adapun konstanta tersebut adalah sebagai berikut:

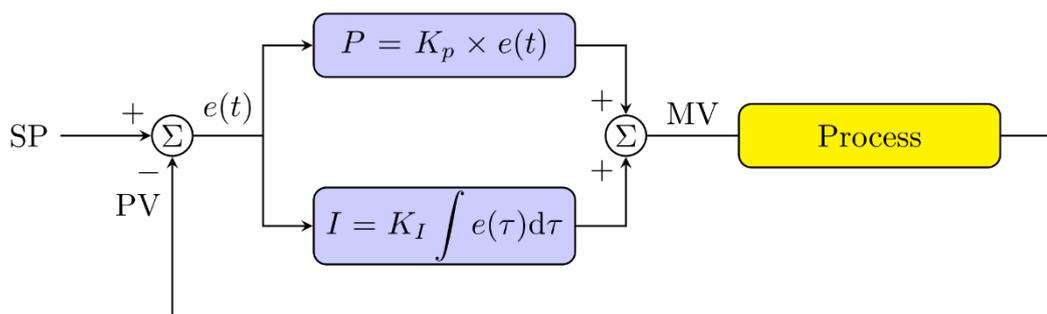
1. moment of inertia of the rotor (J) = $0.1 \text{ kg} \cdot \frac{\text{m}^2}{\text{s}^2}$
2. damping ratio of the mechanical system (b) = 0.01 Ns/m
3. electromotive force constant ($K=Kb=Ka$) = 0.3 Nm/Amp
4. electric resistance (R) = 2 ohm
5. electric inductance (L) = 0.1 H

sehingga diperoleh fungsi transfer sebagai berikut:

$$\frac{\theta(s)}{V(s)} = \frac{30}{s^3 + 2.1s^2 + 11s} \dots\dots\dots (8)$$

3.2 Proportional Integral Control

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:



Gambar 4.2 Blok Kontrol *Proportional Integral*

Proses kontrol integral juga dikenal dengan sistem kontrol reset. Dengan kontrol integral, output dari kontrol ini berubah setiap terjadi error *set point*. Perubahan output dengan *rate*/laju tertentu yang proporsional dengan perubahan nilai *error*:

$$P = \left(\frac{1}{T_i}\right) \int E \, dT$$

Dimana T_i adalah waktu integral (reset), yang nilainya dapat diubah-ubah untuk mengurangi deviasi error pada nilai output. Dengan menggunakan hubungan integral, maka output dari proses kontrol integral akan selalu berubah setiap muncul *error set point*. Dan baru benar-benar diam (tetap pada nilai output tertentu) pada saat nilai error sama dengan nol. Proses kontrol ini membutuhkan waktu yang lebih lama untuk mengejar nilai error sama dengan nol, sehubungan dengan sistem kontrol yang menggunakan rumusan integral.

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:

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

dimana:

T_i = *integral time*

e = *error (input dari unit control)*

K_c = *gain dari controller*

Integral control sangat berguna untuk mencegah *offset error* yang terjadi dan terkait dengan akurasi dari sebuah sistem control. Output akan berkembang sampai sistem merespon dan mereduksi error yang terjadi hingga nilainya 0.

Persamaannya sebagai berikut:

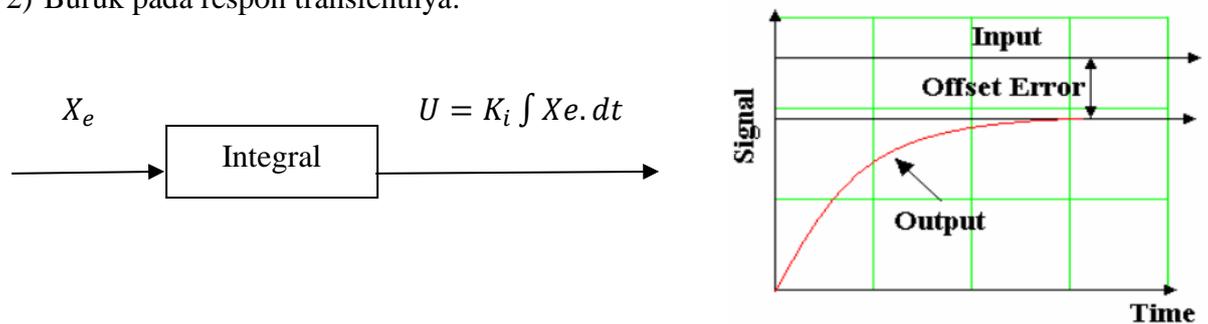
$$U(t) = K_i \int X_e. dt = \frac{K_p}{T_i} \int X_e. dt$$

Laplace Transformnya:

$$U(s) = \frac{K_p X_e}{T_i s}$$

dengan T_i adalah integral nilai konstantanya. Karakteristik integral control, sebagai berikut:

- 1) Mengeliminasi *steady state error*.
- 2) Buruk pada respon transienalnya.



Gambar 4.3 Integral Control

IV. TUGAS PENDAHULUAN

Cari Literatur tentang Kontroler *Proportional Integral* pada kendali posisi motor dc dan kemudian simulasikan pada matlab simulink.

V. ALAT DAN BAHAN

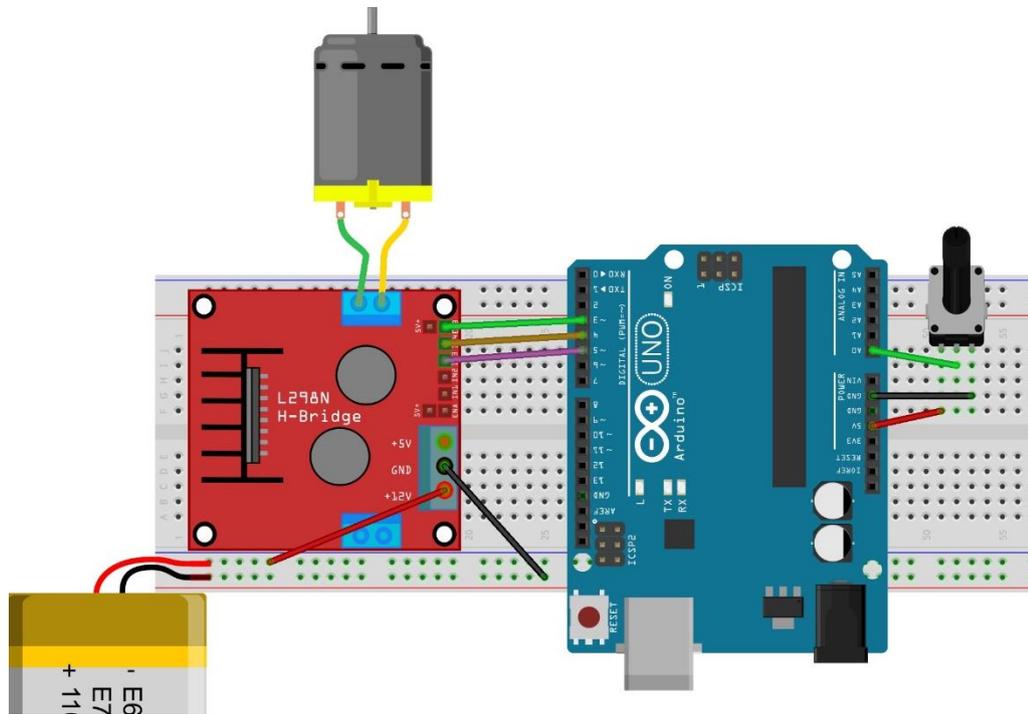
Adapun alat dan bahan yang digunakan adalah sebagai berikut:

1. Motor DC
2. Motor Driver L298N
3. Arduino UNO
4. Potensiometer

VI. RANGKAIAN DAN PROSEDUR PERCOBAAN

1. Hubungkan Motor Driver L298N dan Potensiometer ke arduino seperti pada Gambar 4.4.
2. Ketik kode program pada Tabel 4.1.

3. Upload program ke Arduino uno.
4. Buka jendela “Serial Ploter” pada menu “Tools”



Gambar 4.4 Rangkaian Kerja

Tabel 4.1

```

int error, sudut_ref, motor_pwm, error_sebelumnya;
double sudut_aktual, P, D, nilai_PID, delta_error, I;
const int Toleransi = 5;
const double Kp=0.5, Kd=1, Ki=0.05;

void setup() {

  Serial.begin(115200);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  sudut_ref = 90;
  double Kp=0.5;
}

void loop() {
  int ADCx = analogRead(A0);
  sudut_aktual = map(ADCx, 240, 860, 0, 180); // Konversi nilai
  ADC ke sudut putar

  error = sudut_ref - sudut_aktual;
  if(abs(error)<=Toleransi) error = 0; // toleransi
  error

  P = Kp * error;

```

```

I = I + Ki*error;

nilai_PID = P+I;

if(nilai_PID < 0)
{
    motor_pwm = constrain(abs(nilai_PID), 1, 30); // batasi
nilai PID tidak lebih dari 30
    motor_pwm = motor_pwm+35;
    digitalWrite(9, HIGH);
    digitalWrite(8, LOW);
    analogWrite(10, motor_pwm);
}
else if(nilai_PID > 0)
{
    motor_pwm = constrain(abs(nilai_PID), 1, 30);
    motor_pwm = motor_pwm+35;
    digitalWrite(8, HIGH);
    digitalWrite(9, LOW);
    analogWrite(10, motor_pwm);
}
else // jika nilai_PID = 0
{
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
    analogWrite(10, 0); // motor berhenti
}

Serial.print(sudut_aktual);
Serial.print(" ");
Serial.println(sudut_ref);

error_sebelumnya = error;
}

```

5. Ganti nilai K_p dan K_i sesuai Tabel 4.2 dan kemudian ambil gambar grafik respon Sistem pada **serial Plotter**.
6. Catat *Overshoot*, *Rise time* dan Waktu *steady state* sistem pada tabel.

Tabel 4.2

No	K_p	K_i	<i>Max Overshoot</i>	<i>Rise Time (s)</i>	<i>Steady State(s)</i>
1.	0.1	0.01			
2.	0.3	0.5			
3.	0.5	1			
4.	0.75	2			
5.	1	1			
6.	1.25	0.5			
7.	1.5	0.1			

VII. MATERI PEMBAHASAN

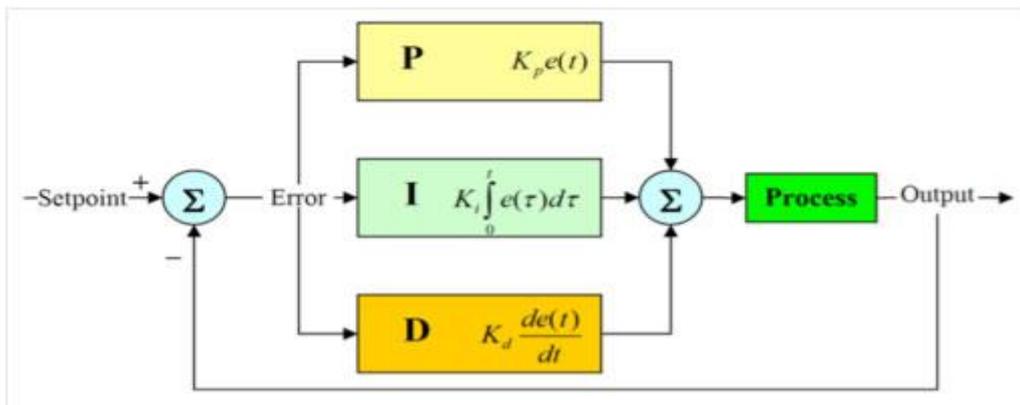
1. Analisa data hasil penelitian
2. Apa pengaruh K_p , K_i pada *respon sistem*?

PERCOBAAN V

SISTEM KENDALI *PROPORTIONAL INTEGRAL DERIVATIVE* (PID)

Dosen : Swadexi Istiqphara, S.T, M.T

NIDK : 20151111010



I. JUDUL PERCOBAAN

SISTEM KENDALI *PROPORTIONAL INTEGRAL DERIVATIVE* (PID)

II. TUJUAN PERCOBAAN

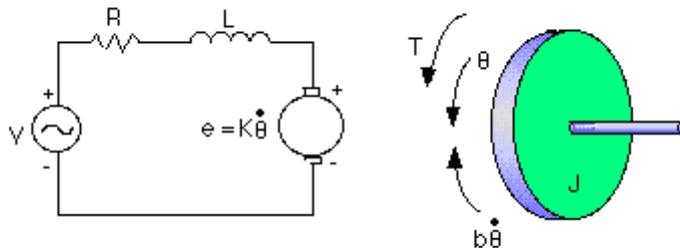
Adapun tujuan dari praktikum ini:

1. Agar mahasiswa akan lebih mengenal tentang prinsip kerja kontrol proporsional *integral derivative*.
2. Mampu menerangkan pengaruh kontroler PID pada kinerja sistem pengaturan posisi.
3. Mampu melakukan analisis kinerja sistem pengaturan posisi motor arus searah.
4. Mampu menerangkan pengaruh kecepatan pada kinerja suatu sistem pengaturan posisi.

III. TEORI SINGKAT

3.1 Motor DC

Motor DC bekerja berdasarkan prinsip gaya elektromagnetik sehingga apabila motor tersebut diberi catu daya, arus akan mengalir ke dalam motor kemudian menghasilkan torsi putar yang sebanding dengan arus tersebut. Pemodelan Rangkaian internal Motor DC secara sederhana dan analisisnya adalah sebagai berikut:



Gambar 5.1. Motor DC

Persamaan torsi yang dibangkitkan oleh Motor DC dapat didekati secara linear menurut persamaan berikut ini:

$$T = K_a i \dots\dots\dots (1)$$

dimana K_a adalah konstanta jangkar motor yang bergantung pada banyaknya lilitan pada jangkar, jumlah kutub medan, tipe belitan dan penampang jangkarnya. Adapun besarnya tegangan ggl induksi lawan yang dibangkitkan motor ketika berputar adalah sebanding dengan konstanta motor K_b dan kecepatan sudut putaran motor $\dot{\theta}$ atau turunan pertama dari posisi sudut motor (θ):

$$e = K_b \dot{\theta} \dots\dots\dots (2)$$

Dengan menggunakan hukum newton, bahwa persamaan torsi yang terkait dengan momen inersia dan rasio redaman dari motor adalah:

$$T = J\ddot{\theta} + b\dot{\theta} \dots\dots\dots (3)$$

dari persamaan (1) dan (3) diperoleh:

$$i = \frac{J\ddot{\theta} + b\dot{\theta}}{K_a} \dots\dots\dots (4)$$

Sedangkan besarnya tegangan V menurut hukum kirchoff adalah:

$$V = iR + L \frac{di}{dt} + e \dots\dots\dots (5)$$

$$V = iR + L \frac{di}{dt} + K_b \dot{\theta} \dots\dots\dots (6)$$

Dengan me-laplace-kan persamaan (4) dan (6) dan mensubstitusikannya, maka diperoleh fungsi transfer antara posisi sudut motor q terhadap tegangan armature V dimana $K_a = K_b$

$$\frac{\theta(s)}{V(s)} = \frac{K}{s((J_s+b)(L_s+R)+K^2)} \dots\dots\dots (7)$$

Persamaan (7) diatas memiliki 5 konstanta yang belum diketahui. Nilai-nilai tersebut pada dasarnya bisa diperoleh dari percobaan identifikasi plant. Pada praktikum saat ini tidak melakukan proses identifikasi tersebut. Tetapi langsung menggunakan konstanta yang biasa dipakai dalam analisis motor DC. Adapun konstanta tersebut adalah sebagai berikut:

1. moment of inertia of the rotor (J) = $0.1 \text{kg} \cdot \frac{\text{m}^2}{\text{s}^2}$
2. damping ratio of the mechanical system (b) = 0.01Ns/m

3. electromotive force constant ($K=K_b=K_a$) = 0.3 Nm/Amp
4. electric resistance (R) = 2 ohm
5. electric inductance (L) = 0.1 H

sehingga diperoleh fungsi transfer sebagai berikut:

$$\frac{\theta(s)}{V(s)} = \frac{30}{s^3 + 2.1s^2 + 11s} \dots\dots\dots (8)$$

3.2 Proportional Integral Derivative Control

Pengontrol PID adalah jenis pengontrol yang banyak diaplikasikan dalam kontrol proses industri karena kesederhanaan strukturnya, lebih tahan terhadap gangguan luar serta dapat diterapkan dalam kondisi operasi yang bervariasi. Tetapi pengontrol PID perlu ditala secara benar yaitu menentukan harga konstanta pengontrol proporsional, integral dan derivatif yang mengoptimalkan kinerja sistem. Setelah tiga parameter tersebut ditala, maka nilai parameter pengontrol tersebut pada PID biasanya dipertahankan tetap selama proses pengontrolan.

Sebuah sistem kendali *close-loop* yang dasar, diperlihatkan pada Gambar 5.2 terdiri dari sebuah pengendali dan sebuah *plant*. PID digunakan sebagai pengendali. Pengendali PID ini terdiri dari tiga buah komponen: bagian *proportional*, bagian *integral* dan bagian *derivative*. Pengendali PID menggunakan persamaan kendali sebagai berikut:

$$C(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) = \frac{K_d s^2 + K_p s + K_i}{s}$$

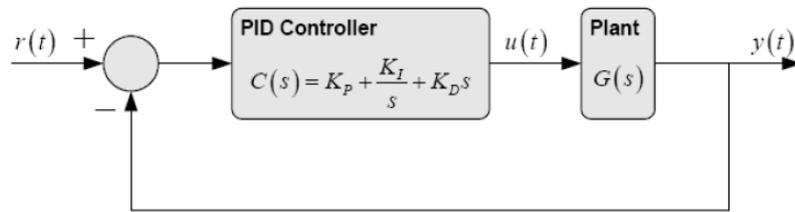
dimana:

K_p adalah parameter proporsional,

$K_i = K_p/T_i$ adalah parameter integral dan

$K_d = K_p \times T_d$ adalah parameter derivatif.

Dalam perancangan pengendali PID, ketiga konstanta tersebut harus dipilih agar sistem *close-loop* memberikan respon yang diinginkan. Respon yang diinginkan haruslah memiliki *settling time* yang minimal dengan *overshoot* yang kecil atau tanpa *overshoot* dari respon step sistem *close-loop*.



Gambar 5.2 Sistem *close-loop*

PID Controller sebenarnya terdiri dari 3 jenis cara pengaturan yang saling dikombinasikan, yaitu P (*Proportional*) Controller, D (*Derivative*) Controller, dan I (*Integral*) Controller. Masing-masing memiliki parameter tertentu yang harus diset untuk dapat beroperasi dengan baik, yang disebut sebagai *konstanta*. Setiap jenis, memiliki kelebihan dan kekurangan masing-masing, hal ini dapat dilihat pada tabel di bawah ini:

Tabel 5.1

Respon PID Controller Terhadap Perubahan Konstanta[1]

<i>Type</i>	<i>Rise Time</i>	<i>Overshoot</i>	<i>Settling Time</i>	<i>Steady State Error</i>
<i>Kp</i>	<i>Decrease</i>	<i>Increase</i>	<i>Small change</i>	<i>Decrease</i>
<i>Ki</i>	<i>Decrease</i>	<i>Increase</i>	<i>Increase</i>	<i>Eliminate</i>
<i>Kd</i>	<i>Small change</i>	<i>Decrease</i>	<i>Decrease</i>	<i>Small change</i>

Sebuah kontrol otomatis dapat menggunakan sistem kontrol yang mengkombinasikan antara ketiga elemen sistem kontrol yang ada. Yang paling banyak digunakan adalah sistem kontrol proporsional, proporsional ditambah integral (PI), dan proporsional ditambah integral ditambah derivatif (PID).

Karakteristik aksi pengontrolan Proporsional adalah mengurangi waktu naik, menambah *overshoot*, dan mengurangi kesalahan keadaan tunak. Fungsi alih sistem dengan menambahkan aksi pengontrolan P menjadi:

$$\frac{P(s)}{q(s)} = \frac{Kp}{s^2 + 5s + (8 + Kp)}$$

Misal, diambil konstanta $Kp = 80$, maka:

$K_p = 80;$

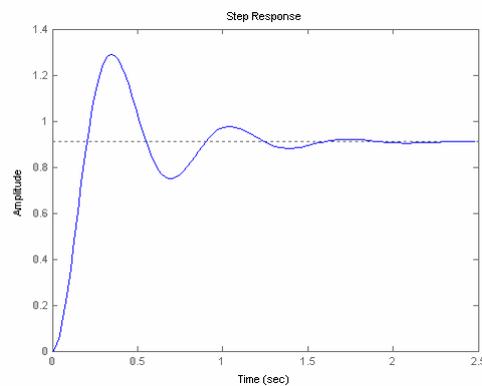
$P_s = [K_p];$

$Q_s = [1 \ 5 \ 8 + K_p];$

$t = 0 : 0.01 : 2;$

`step(Ps, Qs)`

`title('Tanggapan Sistem Loop Tertutup Proporsional')`



Gambar 5.3: Tanggapan sistem terhadap aksi kontrol proporsional

Penambahan aksi kontrol P mempunyai pengaruh mengurangi waktu naik dan kesalahan keadaan tunak, tetapi konsekuensinya *overshoot* naik cukup besar. Kenaikan *overshoot* ini sebanding dengan kenaikan nilai parameter K_p . Waktu turun juga menunjukkan kecenderungan yang membesar.

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

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

dimana:

$u(t)$: sinyal kontrol

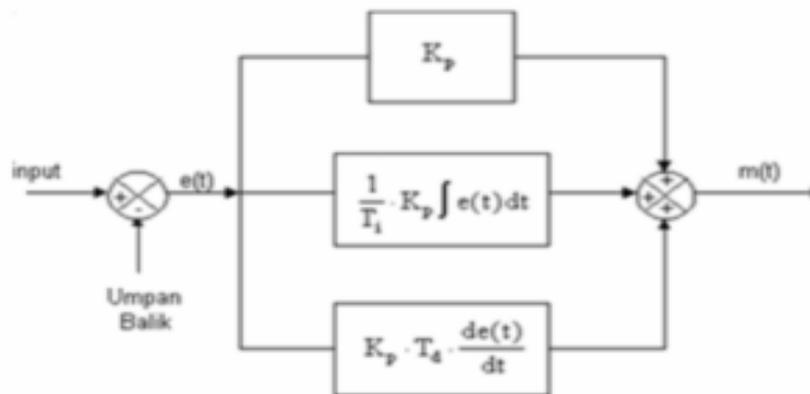
$e(t)$: error

K_c : gain controller

T_i : integral time

T_d : derivative time

Keluaran controller PID merupakan penjumlahan dari keluaran controller proporsional, controller integral dan controller differensial. Karakteristik controller PID sangat dipengaruhi oleh kontribusi besar dari ketiga parameter P, I dan D. Pengaturan 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 menghilangkan *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 *Derivative* menjadi lebih menonjol sehingga respon cenderung cepat, sedangkan T_d yang kecil kurang memberi nilai ekstra pada saat – saat awal.

IV. TUGAS PENDAHULUAN

Cari literatur tentang PID, cari contohnya dan simulasikan pada matlab.

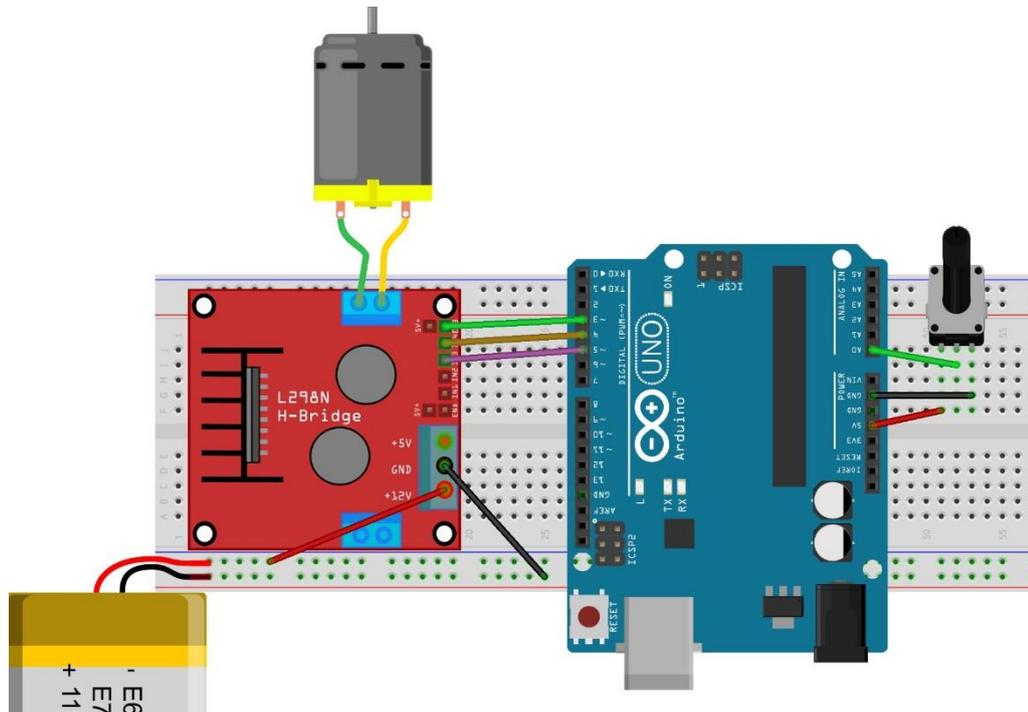
V. ALAT DAN BAHAN

Adapun alat dan bahan yang digunakan adalah sebagai berikut:

1. Motor DC
2. Arduino UNO
3. Motor Driver L298N
4. Potensiometer

VI. RANGKAIAN DAN PROSEDUR PERCOBAAN

1. Hubungkan Motor Driver L298N seperti pada Gambar 5.4.



Gambar 5.4 Rangkaian Kerja

2. Ketik kode program pada Tabel 5.1.
3. Upload program ke Arduino uno.
4. Buka jendela “Serial Ploter” pada menu “Tools”

Tabel 5.1 Kode Program PID

```
int error, sudut_ref, motor_pwm, error_sebelumnya;
double sudut_aktual, P, D, nilai_PID, delta_error, I;
const int Toleransi = 5;
const double Kp=0.5, Kd=1, Ki=0.05;

void setup() {
  Serial.begin(115200);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  sudut_ref = 90;
  double Kp=0.5;
}

void loop() {
  int ADCx = analogRead(A0);
  sudut_aktual = map(ADCx, 240, 860, 0, 180);
  error = sudut_ref - sudut_aktual;
  if(abs(error)<=Toleransi) error = 0; // toleransi error
  delta_error = error - error_sebelumnya;

  P = Kp * error;
  I = I + Ki*error;
  D = Kd*delta_error;

  nilai_PID = P+I+D;

  if(nilai_PID < 0)
  {
    motor_pwm = constrain(abs(nilai_PID), 1, 30); //
    batasi nilai PID tidak lebih dari 30
    motor_pwm = motor_pwm+35;
    digitalWrite(9, HIGH);
    digitalWrite(8, LOW);
    analogWrite(10, motor_pwm);
  }
  else if(nilai_PID > 0)
  {
    motor_pwm = constrain(abs(nilai_PID), 1, 30);
    motor_pwm = motor_pwm+35;
    digitalWrite(8, HIGH);
    digitalWrite(9, LOW);
    analogWrite(10, motor_pwm);
  }
  else // jika nilai_PID = 0
  {
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
    analogWrite(10, 0); // motor berhenti
  }
}
```

```

Serial.print(sudut_aktual);
Serial.print(" ");
Serial.println(sudut_ref);

error_sebelumnya = error;
}}

```

5. Ganti nilai K_p dan K_d sesuai Tabel 5.2 dan kemudian ambil gambar grafik respon Sistem pada **serial Plotter**.
6. Catat *Overshoot*, *Rise time* dan Waktu *steady state* sistem pada tabel.

Tabel 5.2

No	K_p	K_d	K_i	<i>Max Overshoot</i> (%)	<i>Rise Time</i> (s)	<i>Steady State(s)</i>
1.	0.1	0.1	0.01			
2.	0.3	0.5	0.05			
3.	0.5	1	0.1			
4.	0.75	2	0.5			
5.	1	1	1			
6.	1.25	0.5	0.5			
7.	1.5	0.1	0.1			