



**PENGEREMAN OTOMATIS PADA SEPEDA MOTOR MENGGUNAKAN  
METODE PID (*PROPORTIONAL INTEGRAL DERIVATIVE*)**



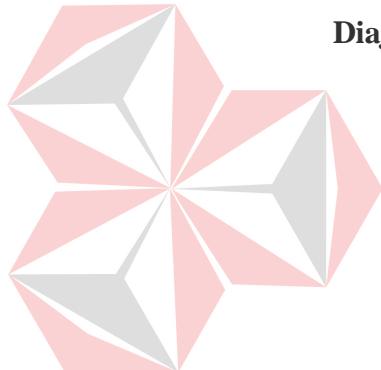
Oleh:  
**Dimas Attala Naoval**  
**18410200045**

---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**  
**UNIVERSITAS DINAMIKA**  
**2022**

**PENGEREMAN OTOMATIS PADA SEPEDA MOTOR  
MENGGUNAKAN METODE PID (*PROPORTIONAL INTEGRAL  
DERIVATIVE*)**

**TUGAS AKHIR**



**Diajukan sebagai salah satu syarat untuk menyelesaikan  
Program Sarjana Teknik**

**UNIVERSITAS  
Dinamika**

**Disusun Oleh:**

<b>Nama</b>	<b>: Dimas Attala Naoval</b>
<b>NIM</b>	<b>: 18410200045</b>
<b>Program Studi</b>	<b>: S1 Teknik Komputer</b>

**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
UNIVERSITAS DINAMIKA  
2022**

## TUGAS AKHIR

### PENEREMAN OTOMATIS PADA SEPEDA MOTOR MENGGUNAKAN METODE PID (*PROPORTIONAL INTEGRAL DERIVATIVE*)

Dipersiapkan dan disusun oleh:

**Dimas Attala Naoval**

**NIM : 18410200045**

Telah diperiksa, dibahas dan disetujui oleh Dewan Pembahas

Pada: 25 Juli 2022

#### Susunan Dewan Pembahas

##### Pembimbing:

- I. Harianto, S.Kom., M.Eng.

NIDN: 0722087701

- II. Musayyanah, S.ST., M.T

NIDN: 0730069102

##### Pembahas:

- I. Pauladie Susanto, S.Kom., M.T.

NIDN: 0729047501



DN: cn=Harianto, c=ID, o=Universitas  
dinamika, ou=Fakultas Teknologi dan  
Informatika, emalh@dinamika.ac.id  
Date: 2022.07.25 15:32:52 +07'00'

Digitally signed by Musayyanah  
DN: cn=Musayyanah, ou=Fakultas  
Teknologi dan Informatika, emalh@dinamika.ac.id, c=ID  
Date: 2022.07.25 16:14:54 +07'00'  
Adobe Acrobat Reader/versi6.0  
2022.07.25 16:14:54 +07'00'

Universitas  
Dinamika  
2022.07.26  
08:22:31 +07'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan

untuk memperoleh gelar Sarjana

Digitally signed by  
Universitas Dinamika  
Date: 2022.07.27  
11:08:01 +07'00'

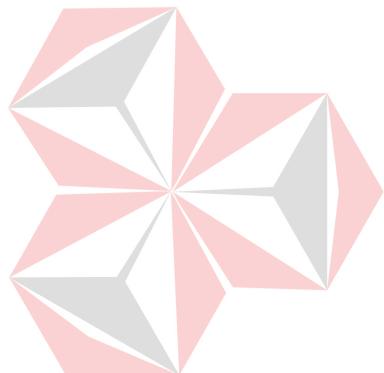
  
**Tri Sagirani, S.Kom., M.MT.**

NIDN: 0731017601

Dekan Fakultas Teknologi dan Informatika  
UNIVERSITAS DINAMIKA

*"Dijalani aja dulu, siapa tau ada solusinya"*

- *Dimas Attala*



UNIVERSITAS  
**Dinamika**

*Dipersembahkan kepada Bapak, Ibu, Keluarga saya, Teman – teman TK18, serta Teman – teman DINAMIKA ROBOTIC atas dukungan, motivasi, dan doa terbaik yang diberikan kepada saya. Beserta semua orang yang selalu membantu, mendukung, memberi masukan, dan memberi motivasi agar tetap berusaha dan belajar agar menjadi lebih baik.*



UNIVERSITAS  
**Dinamika**

**SURAT PERNYATAAN**  
**PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH**

Sebagai mahasiswa Universitas Dinamika, saya:

Nama : Dimas Attala Naoval  
NIM : 18410200045  
Program Studi : S1 Teknik Komputer  
Fakultas : Fakultas Teknologi dan Informatika  
Jenis Karya : Laporan Tugas Akhir  
Judul Karya : **PENEREMAN OTOMATIS PADA SEPEDA MOTOR MENGGUNAKAN METODE PID (*PROPORTIONAL INTEGRAL DERIVATIVE*)**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmedikian dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau sebagai pemilik pencipta dan Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 15 Juni 2022

Yang menyatakan  
  
**Dimas Attala Naoval**  
NIM: 18410200045

## ABSTRAK

Sepeda motor yang nyaman dan memiliki kecepatan yang tinggi adalah impian dari semua pengendara sepeda motor. Namun seiring dengan tingginya kecepatan suatu sepeda motor atau kendaraan, maka resiko terjadinya kecelakaan akan meningkat juga. Berdasarkan permasalahan diatas perlu adanya alat yang dapat mendeteksi jarak aman kendaraan lain yang ada disekitar dan mendeteksi berapa kecepatan saat mengendarai kendaraan bermotor. Dengan dibuatnya alat pengereman otomatis ini, diharapkan bisa menyempurnakan penelitian sebelumnya dengan mengganti sensor ultrasonik HC-SR05 dengan sensor ultrasonik JSN-SR04T yang memiliki batas bacaan maksimal sejauh 2 meter dan menambahkan sensor kecepatan sebagai variable input untuk mengimbangi sensor ultrasonik sebagai nilai yang akan diproses menggunakan metode PID dan mengatur setpoint PID secara otomatis berdasarkan hasil perhitungan yang didapat dari sensor ultrasonic dan sensor kecepatan. Metode PID digunakan untuk menstabilkan gerak output yang berupa servo motor yang digunakan di bagian rem belakang motor. Pada sistem pengendalian dengan PID dirancang lalu parameter kontrol ( $K_p$ ,  $K_i$ ,  $K_d$ ) didapat menggunakan metode Ziegler-Nichols. Dari hasil penelitian yang dilakukan, ditetapkan nilai *setpoint* servo sebesar 30 derajat. Dari hasil pengujian, didapatkan nilai  $K_p$  8,27  $K_i$  45,95 dan  $K_d$  0,37 dengan menggunakan metode Ziegles-Nichols tipe 1. Dari hasil pengujian kontroler PID didapatkan nilai pengereman otomatis *overshoot* 0,6% *rise time* 0,6 detik, *settling time* 0,6 detik, serta *error steady state* 2,21%.

**Kata Kunci:** *Pengereman, Sensor Jarak, PID Ziegler Nichols, Jarak Aman.*

## KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa atas segala rahmat yang telah diberikan-Nya, sehingga penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul “Pengereman Otomatis Menggunakan Metode PID (*Proportional Integral Derivative*)”.

Dalam usaha menyelesaikan penulisan Laporan Tugas Akhir ini penulis banyak mendapat bantuan dari berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih kepada:

1. Allah SWT, karena dengan rahmat-Nya dan hidayah-Nya penulis dapat menyelesaikan Laporan Tugas Akhir ini.
2. Orang Tua dan Seluruh Keluarga penulis tercinta yang telah memberikan dorongan dan bantuan baik moral maupun materi sehingga penulis dapat menempuh dan menyelesaikan laporan tugas akhir ini.
3. Ibu Tri Sagirani, S.Kom., M.MT. selaku Dekan Fakultas Teknologi dan Informatika (FTI) Universitas Dinamika.
4. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer sekaligus sebagai Dosen Penguji. Penulis ucapkan terima kasih atas bimbingan yang diberikan dan kesempatannya serta tuntunan baik itu materi secara tertulis maupun lisan sehingga penulis dapat menyelesaikan Tugas Akhir.
5. Bapak Harianto, S.Kom., M.Eng., selaku dosen pembimbing yang telah memberikan dukungan penuh berupa motivasi, saran, dan wawasan bagi penulis selama pelaksanaan tugas akhir dan pembuatan laporan Tugas Akhir.
6. Ibu Musayyanah, S.ST., M.T., selaku dosen pembimbing yang banyak memberikan masukan dan solusi agar tugas akhir ini menjadi lebih baik dan penulis dapat menyelesaikan Tugas Akhir ini.
7. Teman - teman seperjuangan S1 Teknik Komputer angkatan 2018 Universitas Dinamika, yang telah menemani penulis dalam menempuh jenjang program sarjana.
8. Alief Amelia Rosanti, yang selalu memberikan semangat dan dukungan kepada penulis untuk dapat segera menyelesaikan laporan tugas akhir ini.

Penulis berharap semoga laporan ini dapat berguna dan bermanfaat untuk menambah wawasan bagi pembacanya. Penulis juga menyadari dalam penulisan laporan ini banyak terdapat kekurangan. Oleh karena itu penulis sangat mengharapkan saran dan kritik untuk memperbaiki kekurangan dan berusaha untuk lebih baik lagi.

Surabaya, 8 Juni 2022

Penulis



## DAFTAR ISI

	Halaman
ABSTRAK .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI .....	x
DAFTAR GAMBAR .....	xiii
DAFTAR TABEL .....	xiv
DAFTAR LAMPIRAN .....	xv
BAB I PENDAHULUAN .....	16
1.1    Latar Belakang .....	16
1.2    Rumusan Masalah .....	17
1.3    Batasan Masalah .....	17
1.4    Tujuan .....	17
1.5    Manfaat .....	18
BAB II LANDASAN TEORI .....	19
2.1    Arduino IDE .....	19
2.2    Arduino UNO .....	19
2.3    Sensor <i>Infrared</i> .....	20
2.4    Sensor Jarak JSN – SR04T .....	22
2.5    LCD ( <i>Liquid Crystal Device</i> ) .....	24
2.6    Servo Motor .....	25
2.7    Metode PID ( <i>Proportional Integral Derivative</i> ) .....	26
2.8    Ziegler-Nichols Tipe 1 ( <i>Open Loop</i> ) .....	27
2.9    Ziegler-Nichols Tipe 2 ( <i>Close Loop</i> ) .....	28
BAB III METODOLOGI PENELITIAN .....	30
3.1    Metode Penelitian .....	30

3.1.1	Rancangan Sistem .....	30
3.1.2	Rangkaian Skematik.....	31
3.1.3	Flowchart Sistem.....	32
3.1.4	Normalisasi Data.....	36
3.1.5	Penentuan nilai Kp, Ki, dan Kd .....	37
3.1.6	Model Perancangan Alat .....	39
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>		<b>40</b>
4.1	Pengujian Sensor Jarak JSN-SR04T .....	40
4.1.1	Tujuan Pengujian Sensor Jarak JSN-SR04T.....	40
4.1.2	Alat yang digunakan .....	40
4.1.3	Prosedur Pengujian Sensor Jarak JSN-SR04T .....	40
4.1.4	Hasil Pengujian Sensor Jarak JSN-SR04T.....	41
4.1.5	Analisis Data .....	42
4.2	Pengujian LCD 16x2 .....	42
4.2.1	Tujuan Pengujian LCD 16x2 .....	42
4.2.2	Alat Yang Digunakan.....	42
4.2.3	Prosedur Pengujian LCD 16x2.....	42
4.2.4	Hasil Pengujian LCD 16x2 .....	43
4.2.5	Analisis Data .....	43
4.3	Pengujian Sensor <i>Infrared</i> .....	43
4.3.1	Tujuan Pengujian Sensor <i>Infrared</i> .....	43
4.3.2	Alat Yang Digunakan.....	43
4.3.3	Prosedur Pengujian Sensor <i>Infrared</i> .....	43
4.3.4	Hasil Pengujian Sensor <i>Infrared</i> .....	44
4.3.5	Analisis Data .....	45
4.4	Pengujian Sistem PID .....	45

4.4.1	Tujuan Pengujian .....	45
4.4.2	Alat Yang Digunakan.....	45
4.4.3	Prosedur Pengujian.....	46
4.4.4	Hasil Pengujian Sistem PID.....	46
4.4.5	Analisis Data .....	46
<b>BAB V PENUTUP</b>	.....	<b>48</b>
5.1	Kesimpulan.....	48
5.2	Saran .....	48
<b>DAFTAR PUSTAKA</b>	.....	<b>49</b>
<b>LAMPIRAN</b>	.....	<b>50</b>
<b>BIODATA PENULIS</b>	.....	<b>66</b>



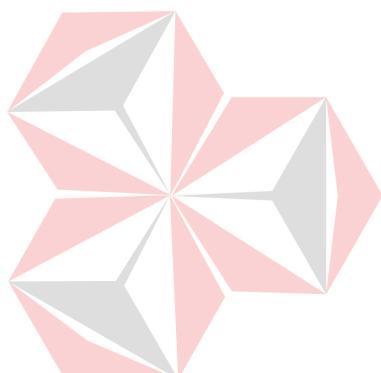
UNIVERSITAS  
**Dinamika**

## DAFTAR GAMBAR

	Halaman
Gambar 2. 1 Arduino Uno.....	19
Gambar 2. 2 Sensor Infrared.....	21
Gambar 2. 3 Sensor Jarak Jsn-Sr04t .....	22
Gambar 2. 4 Lcd 16x2.....	24
Gambar 2. 5 Servo Motor.....	25
Gambar 2. 6 Diagram Blok Pid.....	26
Gambar 2. 7 Tanggapan Unit-Step.....	27
Gambar 2. 8 Kurva Respon Berbentuk S .....	27
Gambar 2. 9 Blok Diagram Ziegler Nichols Tipe 2.....	28
Gambar 2. 10 Proses Osilasi Parameter Pcr.....	28
Gambar 3. 1 Blok Diagram Sistem .....	30
Gambar 3. 2 Rangkaian Keseluruhan Sistem.....	31
Gambar 3. 3 Flowchart Sistem Penggereman Otomatis .....	32
Gambar 3. 4 Flowchart Sistem Pid .....	34
Gambar 3. 5 Blok Diagram Sistem Pid .....	35
Gambar 3. 6 Grafik Uji Open Loop Metode Pid.....	38
Gambar 3. 7 Kotak Alat Dan Sensor.....	39
Gambar 4. 1 Pengujian Sensor Jarak.....	41
Gambar 4. 2 Hasil Pengujian Lcd 16x2 .....	43
Gambar 4. 3 Pengujian Sensor Kecepatan .....	44
Gambar 4. 4 Grafik Derajat Pid .....	46

## DAFTAR TABEL

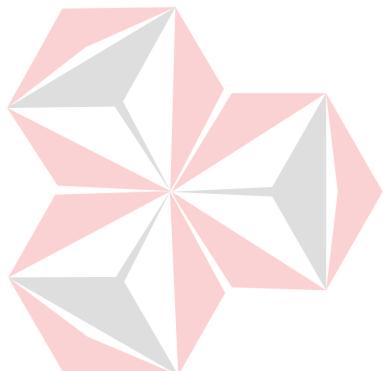
	Halaman
Tabel 2. 1 Spesifikasi Arduino Uno .....	20
Tabel 2. 2 Spesifikasi Sensor Infrared .....	21
Tabel 2. 3 Spesifikasi Sensor Jarak Jsn-Sr04t.....	23
Tabel 2. 4 Spesifikasi Lcd 16x2 .....	24
Tabel 2. 5 Spesifikasi Servo Motor.....	25
Tabel 2. 6 Aturan Tuning Ziegler Nichols Open Loop .....	28
Tabel 2. 7 Aturan Pengaturan Ziegler-Nichols Closed Loop.....	29
Tabel 3. 1 Pin Arduino Uno .....	31
Tabel 4. 1 Hasil Pengujian Sensor Jarak Jsn-Sr04t.....	41
Tabel 4. 2 Hasil Pengujian Sensor Kecepatan .....	44



UNIVERSITAS  
**Dinamika**

## DAFTAR LAMPIRAN

Lampiran 1 Source Code Program Pengujian Sensor Jarak Jsn-Sr04t .....	50
Lampiran 2 Source Code Program Pengujian Lcd 16x2.....	50
Lampiran 3 Source Code Program Pengujian Sensor Infrared (Kecepatan).....	51
Lampiran 4 Source Code Program Keseluruhan.....	53
Lampiran 5 Hasil Pengujian Sistem Pengereman Otomatis .....	59
Lampiran 6 Rangkaian Skematik Setiap Input Dan Output.....	60
Lampiran 7 Hasil Turnitin.....	62



UNIVERSITAS  
**Dinamika**

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Perkembangan industri otomotif saat ini semakin meningkat dan canggih yang membuat banyaknya produksi sepeda motor khususnya. Sepeda motor yang aman serta mempunyai kecepatan yang tinggi merupakan keinginan bagi semua pengendara motor roda dua. Tetapi bersamaan dengan meningkatnya kecepatan suatu sepeda motor, maka efek terjadinya musibah akan meningkat juga. (Alam & Maulana, 2020)

Menurut Korlantas Polri, jumlah kecelakaan di Jawa Timur telah terjadi sebanyak 80 kasus pada tahun 2022. Saat ini, total kerusakan properti diperkirakan mencapai Rp 40 juta. Menurut Kementerian Perhubungan, kecelakaan lalu lintas disebabkan oleh banyak faktor. Dengan kata lain, 61% kecelakaan disebabkan oleh faktor manusia (berkaitan dengan keterampilan dan kepribadian pengemudi) dan 9% disebabkan oleh faktor kendaraan (berkaitan dengan terwujudnya teknologi keselamatan lalu lintas). 30% berdasarkan faktor infrastruktur dan lingkungan. (Moniva & Mutia, 2022)

Berdasarkan informasi diatas, persentase terbesar disebabkan oleh faktor manusia. Faktor manusia sendiri bisa terjadi karena lalainya pengguna kendaraan bermotor, salah satunya adalah tidak memperkirakan kondisi jarak dan kecepatan yang dibutuhkan, untuk menggunakan penggereman supaya penggereman bekerja secara optimal dan tidak dapat menyebabkan kecelakaan. Dalam penelitian ini, peneliti menggunakan metode PID (*Proporsional, Integral, Derivative*) untuk merancang sistem kontrol otomatis. Kontroler PID menciptakan kontrol yang lebih stabil dengan membandingkan nilai kesalahan dari *feedback* sistem untuk menentukan nilai kontrol selanjutnya. (Ira Puspasari, 2018).

Berdasarkan permasalahan diatas perlu adanya alat yang dapat mendeteksi jarak aman kendaraan lain yang ada disekitar dan mendeteksi berapa kecepatan saat mengendarai kendaraan bermotor. Pada penelitian sebelumnya (Alam & Maulana, 2020) hanya menggunakan variable jarak untuk menentukan penggereman secara otomatis dan tidak menggunakan metode apapun yang menyebabkan pengaturan

output menjadi kurang maksimal. Kemudian dari penelitian sebelumnya (Agprianta, 2018) dapat mengatur setpoint secara manual dengan memberi nilai sesuai keinginan pengguna. Dengan dibuatnya alat pengereman otomatis ini, diharapkan bisa menyempurnakan penelitian sebelumnya dengan mengganti sensor jarak HC-SR05 dengan sensor jarak JSN-SR04T yang memiliki batas bacaan maksimal sejauh 2 meter dan menambahkan sensor kecepatan sebagai variable input untuk mengimbangi sensor ultrasonik sebagai nilai yang akan diproses menggunakan metode PID dan mengatur setpoint PID secara otomatis berdasarkan hasil perhitungan yang didapat dari sensor ultrasonic dan sensor kecepatan. Metode PID digunakan untuk menstabilkan gerak output yang berupa servo motor yang digunakan di bagian rem belakang motor. Dengan adanya penambahan alat dan metode yang lebih baik dari metode sebelumnya, diharapkan alat ini dapat mengurangi tingkat terjadinya kecelakaan kendaraan bermotor.



### **1.2 Rumusan Masalah**

Dari latar belakang, dapat disimpulkan bahwa rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana membangun alat pengereman otomatis dengan input sensor jarak dan kecepatan pada sepeda motor?
2. Bagaimana menerapkan sistem kendali PID untuk menjaga jarak aman dan kecepatan sepeda motor?

### **1.3 Batasan Masalah**

Dalam merancang dan membuat alat ini, uraian masalah dibatasi pada:

1. Sensor JSN-SR04T hanya mampu mengukur jarak minimal 25 cm dan maksimal 200 cm.
2. Servo AX8601 hanya mampu menggerakkan beban maksimal hingga 25 kg.
3. Penggunaan output servo hanya digunakan di rem belakang.

### **1.4 Tujuan**

Dari latar belakang dan permasalahan tersebut, maka tujuan dari Tugas Akhir ini adalah:

1. Membuat alat pengereman otomatis pada sepeda motor dengan menggunakan sensor jarak dan sensor infra merah.
2. Menerapkan sistem kendali PID untuk menjaga jarak aman dan kecepatan sepeda motor.

### **1.5 Manfaat**

Kelebihan atau manfaat dari Tugas Akhir ini adalah:

1. Mengurangi terjadinya kecelakaan yang disebabkan oleh kelalaian pengguna kendaraan bermotor.
2. Mengembangkan industri khususnya dibidang otomotif.
3. Dapat menjadi referensi bagi mahasiswa yang membutuhkannya.



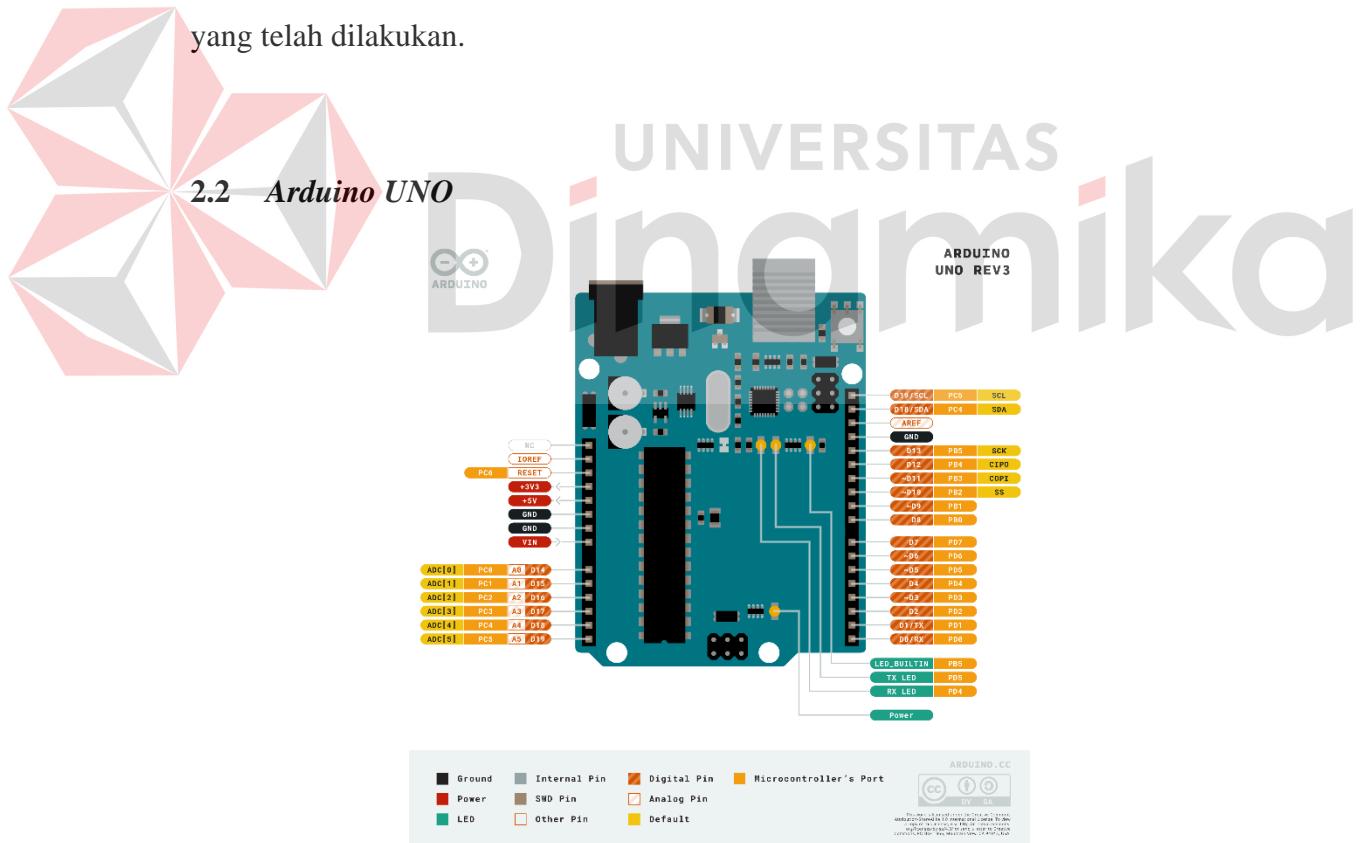
## BAB II

### LANDASAN TEORI

#### 2.1 *Arduino IDE*

IDE adalah singkatan untuk *Integrated Development Environment*. Arduino disebut lingkungan karena diprogram oleh perangkat lunak ini untuk melakukan fungsi yang dibangun ke dalam *syntax* pemrograman. Arduino dapat dioperasikan dengan bahasa pemrograman berpemilik yang mirip dengan C (Arduino, 2017). Bahasa pemrograman Arduino telah diubah untuk memudahkan pemula memprogram dari bahasa aslinya.

Dalam Tugas Akhir ini Arduino IDE difungsikan untuk membuat program yang akan berfungsi untuk membaca nilai dari sensor jarak dan sensor *infrared*, membuat perhitungan normalisasi dan perhitungan metode PID, serta mengendalikan derajat motor servo berdasar dari hasil perhitungan metode PID yang telah dilakukan.



Gambar 2. 1 Arduino Uno  
(sumber: <https://store.arduino.cc/products/arduino-uno-rev3>)

Pada Gambar 2.1 merupakan *Board* Arduino Uno yang menggunakan ATmega328P. Arduino Uno nantinya digunakan untuk memproses data input,

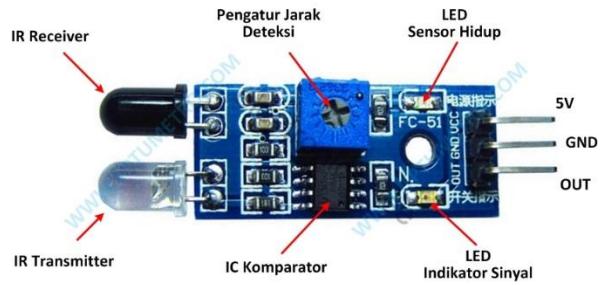
output, dan metode yang akan digunakan di Tugas Akhir ini. Input yang digunakan adalah sensor jarak JSON – SR04T dan sensor *infrared* yang berfungsi untuk mengukur kecepatan sepeda motor. Metode yang digunakan yaitu metode normalisasi dan metode PID, sedangkan untuk output yang digunakan adalah LCD 16x2 untuk melihat hasil pembacaan dari sensor dan motor servo untuk mengontrol pegas rem belakang. Spesifikasi dari Arduino Uno ada di Tabel 2. 1

Tabel 2. 1 Spesifikasi Arduino Uno

Mikrokontroler	Atmega328
Tegangan Operasi	5volt
Tegangan Rekomendasi	7-12 volt
Rentang Tegangan	6-20 volt
Kaki - kaki Input/Output Digital	14
Kaki - kaki Input Analog	6
Ampere Pada Pin Digital	40 mA
Ampere Pada Pin 3,3	50 mA
<i>Memory Flash</i>	32 KB (0,5 KB untuk <i>boot loader</i> )
SRAM	2 KB
EEPROM	1 KB
Kecepatan <i>Clock</i>	16 MHz

### 2.3 Sensor *Infrared*

Modul sensor infra merah adalah modul komponen yang difungsikan untuk menangkap sinar infra merah di daerah deteksi. Dalam modul *infrared* ada dua komponen yaitu pemancar (*infrared transmitter*) dan penerima (*infrared receiver*). *Infrared receiver* merupakan sebuah *photodiode* yang dapat memancarkan sinar *infrared*, sedangkan *infrared receiver* adalah diode khusu yang berfungsi untuk menerima sinar *infrared*. Pada Tugas Akhir ini sensor *infrared* akan digunakan sebagai pendekripsi kecepatan dengan menggunakan rumus tertentu. Gambar sensor inframerah ditunjukkan pada Gambar 2.8.

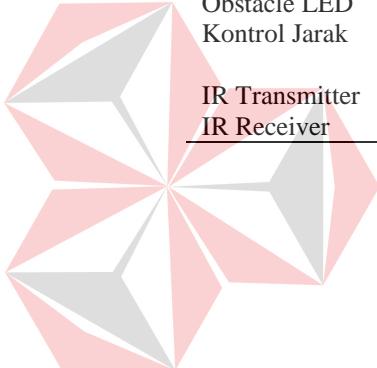


Gambar 2. 2 Sensor *Infrared*

Sumber: (<https://www.tptumetro.com/2021/01/cara-kerja-modul-infrared-fc-51-sensor.html>)

Tabel 2. 2 Spesifikasi Sensor Infrared

Nama	Keterangan
Pin VCC	3.3 to 5 VDC Tegangan Masukan
Pin GND	Ground Input
Output	Output akan LOW jika halangan terdeteksi
Power LED	Menyala jika power terpasang
Obstacle LED	Menyala jika halangan terdeteksi
Kontrol Jarak	Search jarum jam untuk mengurangi jarak, dan berlawanan arah jarum jam untuk menambah jarak
IR Transmitter	Untuk memancarkan gelombang sinyal
IR Receiver	Untuk menerima gelombang sinyal



UNIVERSITAS  
Dinamika

## 2.4 Sensor Jarak JSN – SR04T

JSN-SR04T adalah sensor jarak yang dikembangkan dari sensor ultrasonik HC-SR04. Ini memiliki kelebihan seperti menambahkan fungsi tahan air, dan dapat digunakan bahkan di tempat yang lembab.



Gambar 2. 3 Sensor Jarak JSN-SR04T

(sumber: <https://www.nyebarilmu.com/tutorial-mengakses-sensor-ultrasonic-jsn-sr04t/>)

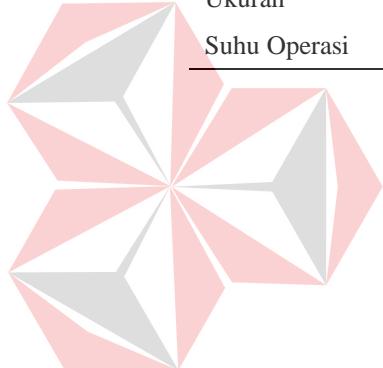
Pada Gambar 2.3 ini merupakan sensor jarak JSN-SR04T dengan pembacaan data yang stabil. Cara kerja sensor jarak yaitu dengan cara memancarkan gelombang ultrasonik. Gelombang ini dikembalikan oleh suatu objek dan dideteksi oleh sensor. Dapatkan jarak antara sensor dan objek dengan menghitung waktu antara pengiriman dan penerimaan gelombang suara.

Perhitungan rumus jarak didapatkan dari kecepatan gelombang suara sekitar 340 m/s, jadi untuk 1 sentimeter dapat dihitung dengan rumus  $\frac{1}{340 \times 100}$  kecepatan dari suara adalah 0,0000294 detik atau 29,4  $\mu$ s, karena gelombang ultrasonic mengirim dan menerima sinyal, sehingga ada 2 kali waktu yaitu 29,4 dikali 2 menjadi 58,8  $\mu$ s yang dibulatkan menjadi 58  $\mu$ s. Yang signifikan antara sensor ini dan HC-SR04 kecuali kedap air adalah sensor ini hanya menggunakan satu

perangkat ultrasonic yang bertindak sebagai pemancar dan penerima ultrasonik. Adapun spesifikasi dari sensor JSN-SR04T ada di Tabel 2.3.

Tabel 2. 3 Spesifikasi Sensor Jarak JSN-SR04T

Nama	Keterangan
Tegangan Operasi	DC 3.0-5.5V
Konsumsi Arus	Kurang dari 8mA
Frekuensi	40KHz
Jarak Terjauh	600cm
Jarak Terdekat	20cm
Akurasi Jarak	$\pm 1\text{cm}$
Resolusi Pembacaan	1mm
Sudut Pengukuran	$75^\circ$
Wiring	3-5.5V (Power +), Trig RX, Echo TX, GND
Ukuran	L42 * W29 * H12 mm
Suhu Operasi	-7°C sampai +40°C



## 2.5 LCD (*Liquid Crystal Device*)

LCD (*Liquid Crystal Display*) adalah sejenis media tampilan yang tampilan utamanya adalah layar kristal cair. LCD digunakan dalam berbagai bidang perangkat elektronik.

LCD yang digunakan pada Tugas Akhir ini adalah LCD 16x2 yang berarti LCD terdiri dari 16 kolom dan 2 baris karakter. LCD ini nantinya akan berfungsi sebagai penampil dari nilai nilai yang telah diproses di mikrokontroler Arduino. Gambar LCD 16x2 ditunjukkan pada Gambar 2.4.



Gambar 2. 4 LCD 16x2

(sumber: <https://www.nyebarilmu.com/cara-mengakses-modul-display-lcd-16x2/>)

Adapun spesifikasi dari LCD 16x2 ada di Tabel 2.4.

Tabel 2. 4 Spesifikasi LCD 16x2

PIN	SIMBOL	FUNGSI
1	VSS	Power Ground
2	VDD	Power untuk Logic
3	V0	Pengaturan Kontras
4	RS	Data Command
5	RW	Read and write
6	EN	Sinyal Enable
7-14	D0-D7	Data Bus
15	BLA	5V
16	BLK	0V

## 2.6 Servo Motor

Servo motor adalah komponen yang digunakan dalam alat industri cerdas yang dirancang untuk menggerakkan objek menggunakan kontrol yang memerlukan presisi tinggi dalam hal posisi sudut, akselerasi, dan kecepatan.

Pada Tugas Akhir ini servo motor digunakan untuk menggerakkan pegas rem belakang untuk mengatur berapa tekanan rem yang diperlukan oleh system yang telah dibuat. Gambar servo ditunjukan pada Gambar 2.5



Gambar 2. 5 Servo Motor

(sumber: <https://www.jsumo.com/mg996r-servo-motor-digital>)

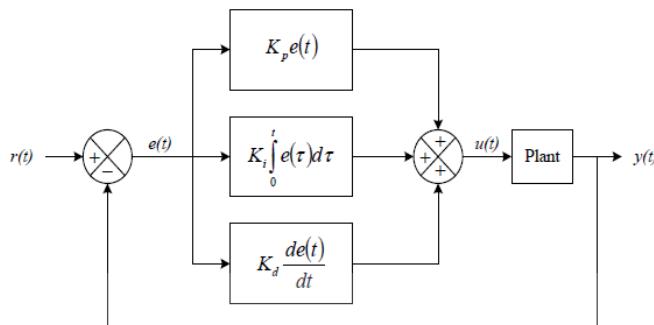
Adapun spesifikasi dari Servo Motor ditunjukkan pada Tabel 2.4

Tabel 2. 5 Spesifikasi Servo Motor

Nama	Keterangan
Dead Band	4 $\mu$ s
Posisi Netral	1500 $\mu$ s / 330hz
Motor	Core Motor
Angle	130 - 160° $\pm$ 2°
Kecepatan Operasi (6V)	0.15sec/60°
Beban Torsi (6V)	25kg-cm

## 2.7 Metode PID (*Proportional Integral Derivative*)

Fungsi dari kontroler PID adalah untuk menjaga keakuratan sistem instrumentasi dengan umpan balik yang bertindak sebagai kontrol kesalahan agar keluaran sesuai atau mendekati pengaturan titik masukan yang diinginkan oleh pengguna. Cara mendapatkan nilai PID dengan cara coba-coba (trial and error) dipicu oleh parameter  $K_p$ ,  $K_i$ , dan  $K_d$ . (Ira Puspasari, 2018). Gambar 2.6 menunjukkan diagram skema dari kesalahan blok PID.



Gambar 2. 6 Diagram Blok PID

Berikut adalah pengertian nilai-nilai yang terdapat pada Gambar 2.3:

- Nilai  $r(t)$ : Nilai *setpoint*.
- Nilai  $y(t)$ : Nilai keluaran metode PID yang sudah memenuhi nilai yang ditetapkan (*setpoint*).
- Nilai  $e(t)$ : Nilai kesalahan.
- Nilai  $u(t)$ : Jumlah perhitungan nilai  $K_p$ ,  $K_i$ , dan  $K_d$ .

Konstanta PID terlebih dahulu diubah menjadi persamaan diskrit sebagai berikut:

- a. Bentuk Diskrit

$$u(t_k) = K_p e(t_k) + K_i \Delta t \sum_{i=1}^k e(t_i) + \frac{K_d}{\Delta t} (e(t_k) - e(t_{k-1})) \quad (2)$$

Keterangan rumus PID bentuk diskrit:

$e(t_k)$ : *error* sekarang.

$\Delta t \sum_{i=1}^k e(t_i)$ : jumlah *error*.

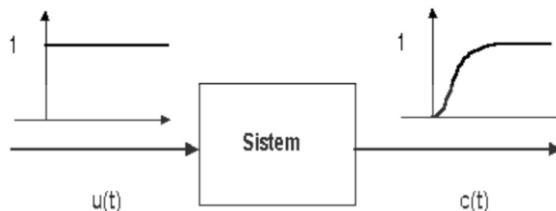
$e(t_{k-1})$ : *error* sebelumnya.

Karena mikrokontroler memiliki sinyal digital yang menyampaikan informasi biner dan mewakili nilai diskrit, ekspresi PID yang digunakan untuk mikrokontroler

menggunakan format diskrit (Feni Isdaryani, 2021). Oleh karena itu, untuk menggunakan metode atau rumus PID, harus terlebih dahulu untuk mengonversinya ke bentuk diskrit untuk memudahkan penulisan ke dalam bentuk program Bahasa C yang ada di mikrokontroler.

## 2.8 Ziegler-Nichols Tipe 1 (*Open Loop*)

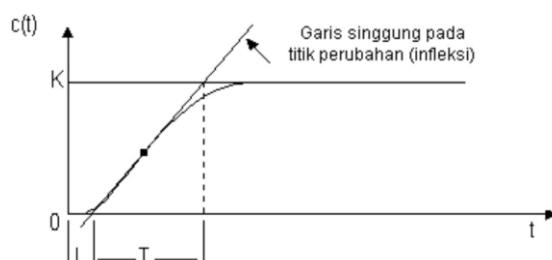
Untuk mendapatkan parameter  $K_p$ ,  $K_i$ , dan  $K_d$  pada metode loop terbuka, dilakukan percobaan respon plant terhadap input unit step dan mengamati output dari kondisi awal hingga kondisi stabil. (P.B, 2018). Hasil outputnya membentuk kurva berbentuk S seperti ditunjukkan Gambar 2.7.



Gambar 2. 7 Tanggapan unit-step

Sumber: (Fitria Suryatini, 2018)

Fungsi kurva S adalah untuk menentukan nilai jeda waktu ( $L$ ) dan konstanta waktu ( $T$ ). Proses penentuan kurva  $L$  dan  $T$  ditunjukkan pada Gambar 2.8, dari nilai  $L$  dan  $T$  tersebut parameter  $K_p$ ,  $K_i$ , dan  $K_d$  dapat diperoleh.



Gambar 2. 8 Kurva respon berbentuk S

Sumber: (Anita Puput Widiastuti, 2016)

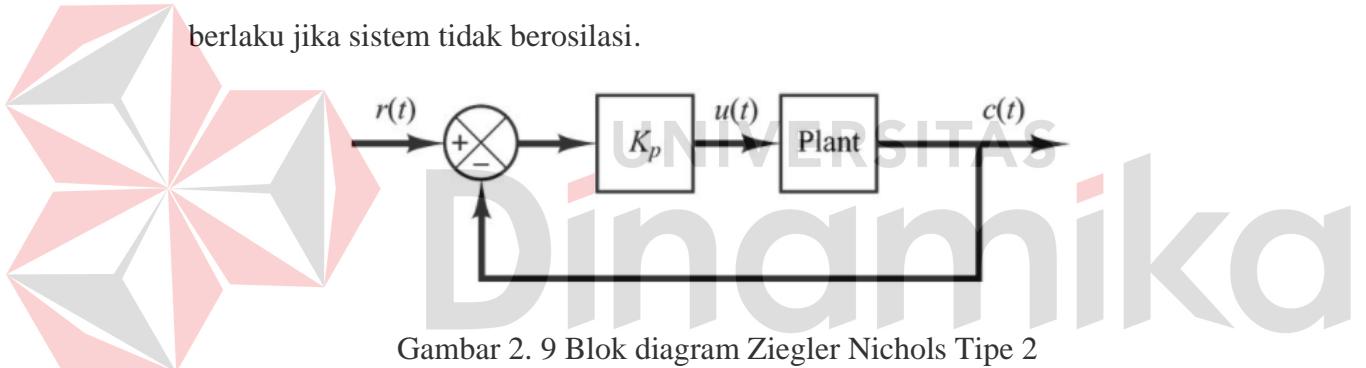
Setelah mendapatkan nilai parameter  $L$  dan  $T$ , akan dilanjutkan ke perhitungan tuning (Anita Puput Widiastuti, 2016) pada Tabel 2.6.

Tabel 2. 6 Aturan tuning Ziegler Nichols Open Loop

Tipe Kontroler	Kp	Ti	Td	Ki	Kd
P	$T_L$	$\infty$	0		
PI	$0,9 T_L$	$L_{0,3}$	0		
PID	$1,2 T_L$	2L	0,5L	$K_p/T_i$	$K_p \times T_d$

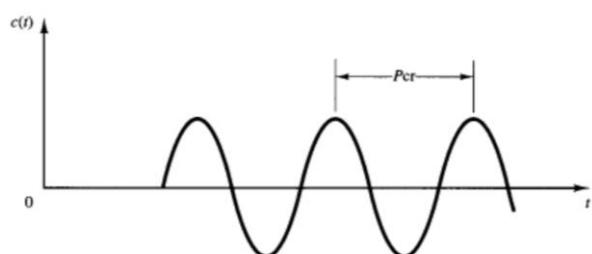
## 2.9 Ziegler-Nichols Tipe 2 (*Close Loop*)

Sistem Kontrol Tertutup adalah sistem kontrol umpan balik. Dengan metode ini, pertama-tama atur nilai  $T_i = \infty$  dan  $T_d = 0$ . Gunakan kendali  $K_p$  untuk menggetarkan sistem dengan mengubah nilai  $K_p$  dari 0 hingga nilai kritis  $K_{cr}$ . Sebuah plot getaran secara berulang ditunjukkan pada Gambar 2.9. Metode ini tidak berlaku jika sistem tidak berosilasi.



Gambar 2. 9 Blok diagram Ziegler Nichols Tipe 2

Sumber: (Nasrul Hamid, 2021)



Gambar 2. 10 Proses osilasi parameter pcr

Sumber: (Anita Puput Widiastuti, 2016)

Perhitungan PID Ziegler-Nichols Close Loop pada Table 2.7.

Tabel 2. 7 Aturan pengaturan Ziegler-Nichols Closed Loop

Tipe Kontroler	Kp	Ti	Td	Ki	Kd
P	$0.5 K_{cr}$	$\infty$	0		
PI	$0.45 K_{cr}$	$\frac{1}{1,2} P_{cr}$	0		
PID	$0.6 K_{cr}$	$0,5 P_{cr}$	$0,125 P_{cr}$	$2 \times \frac{K_p}{T_i}$	$K_p \times Td$



## BAB III

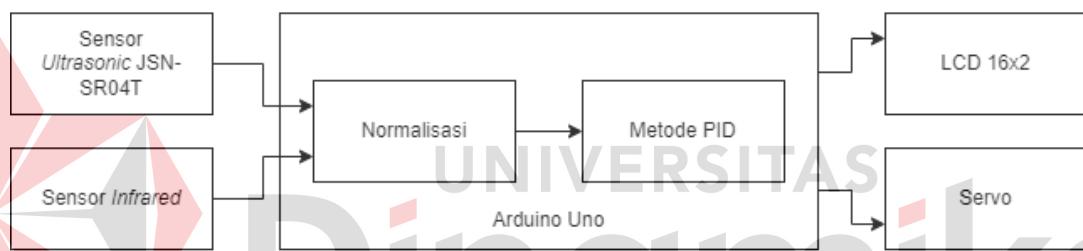
### METODOLOGI PENELITIAN

#### 3.1 Metode Penelitian

Metode penelitian yang terdapat dalam Tugas Akhir ini terdiri dari perancangan sistem, skema diagram, diagram alir sistem, dan model perancangan. Penelitian ini diawali dengan pengontrolan jarak dan kecepatan secara otomatis dengan mikrokontroler Arduino. Kontrol ini dilakukan sesuai kebutuhan jarak aman untuk pengereman. Kondisi yang telah diproses mikrokontroler juga dapat dimonitoring dari LCD.

##### 3.1.1 Rancangan Sistem

Diagram blok dari sistem yang dibuat terletak pada Gambar 3.1.



Gambar 3. 1 Blok diagram sistem

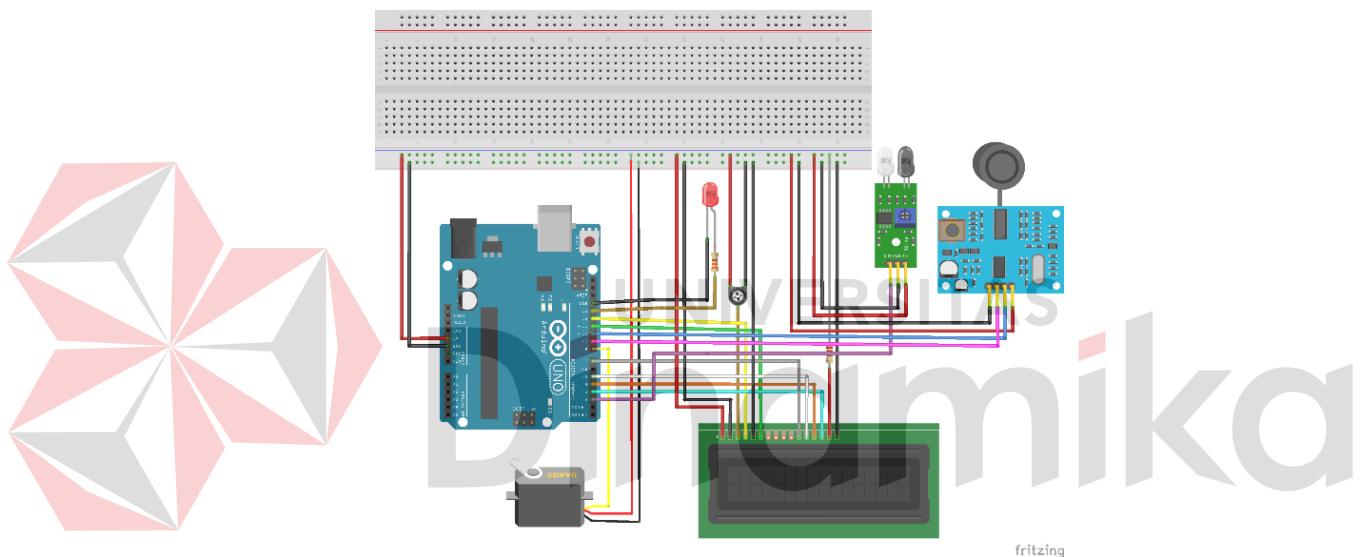
Pada Gambar 3.1 diagram blok sistem di atas, setiap bagian terhubung ke Arduino Uno memiliki fungsi:

1. *Input*
  - a. Sensor *Ultrasonic* JSON – SR04T: Sensor ini digunakan untuk pengukuran jarak antar kendaraan.
  - b. Sensor *Infrared*: Komponen ini berfungsi untuk mendeteksi kondisi yang akan digunakan untuk perhitungan kecepatan kendaraan.
2. *Proses*
  - a. Arduino Uno: Alat ini digunakan untuk mengontrol masukan dan keluaran dari sistem pengereman otomatis.
  - b. Normalisasi: Normalisasi digunakan untuk menghitung data dari sensor jarak dan kecepatan menjadi satu nilai normalisasi, hasil normalisasi nanti yang akan digunakan untuk setpoint dari PID.

- c. Metode PID: Kontroler PID digunakan untuk memproses perbandingan nilai masukan yang diterima dengan menggunakan parameter Kp, Ki, dan Kd yang diberikan dan menghasilkan output yang diinginkan.
3. *Output*
- a. Servo Motor: Alat ini bertindak sebagai pengatur tekanan rem.
  - b. LCD 16x2: berfungsi untuk menampilkan kondisi jarak dan kecepatan yang telah diproses di arduino.

### 3.1.2 Rangkaian Skematik

Gambar 3.2 menunjukkan rangkaian perangkat yang digunakan.



Gambar 3. 2 Rangkaian keseluruhan sistem

Keterangan kaki – kaki komponen pada Gambar 3.2 terdapat pada tabel:

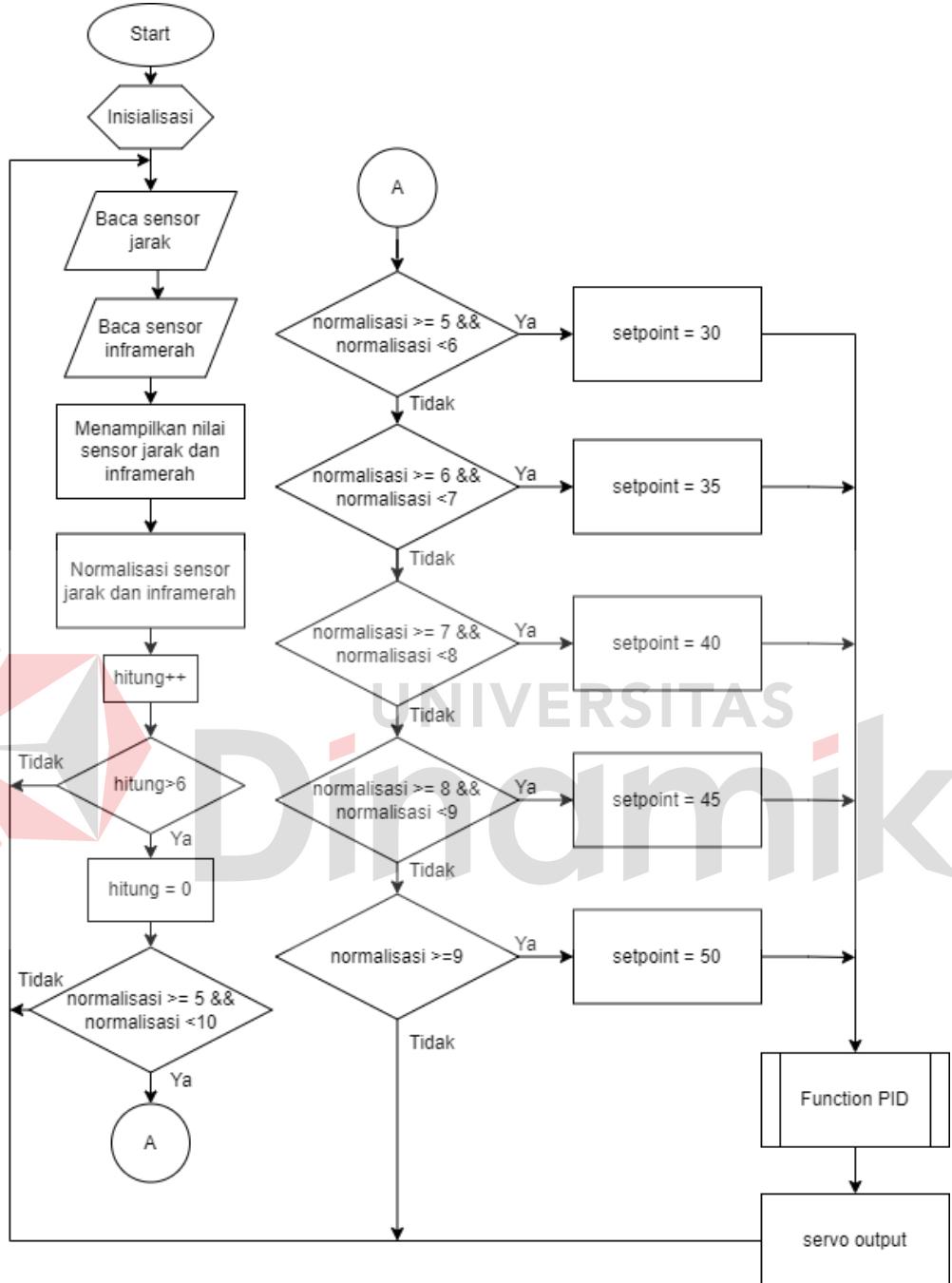
Tabel 3. 1 Pin Arduino Uno

No	Nama rangkaian	Pin Out
1	Sensor <i>Infrared</i>	D2
2	Sensor <i>Ultrasonic</i> JST-SR04T	TRIG(D10) & ECHO(D9)
3	LED	D13
4	Servo	D6
5	LCD 16x2	RS(12), EN(11), D7(3), D6(4), D5(5), D6(7)

Untuk rangkaian skematik setiap input dan output dapat dilihat di Lampiran 6

### 3.1.3 Flowchart Sistem

Diagram alir keseluruhan untuk sistem ini dijelaskan sebagai berikut:



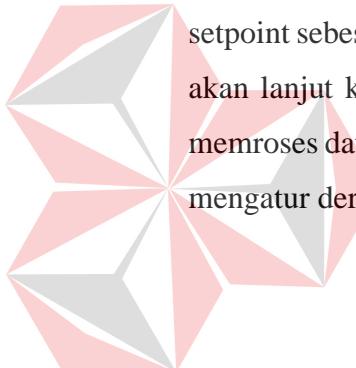
Gambar 3. 3 Flowchart sistem pengerman otomatis

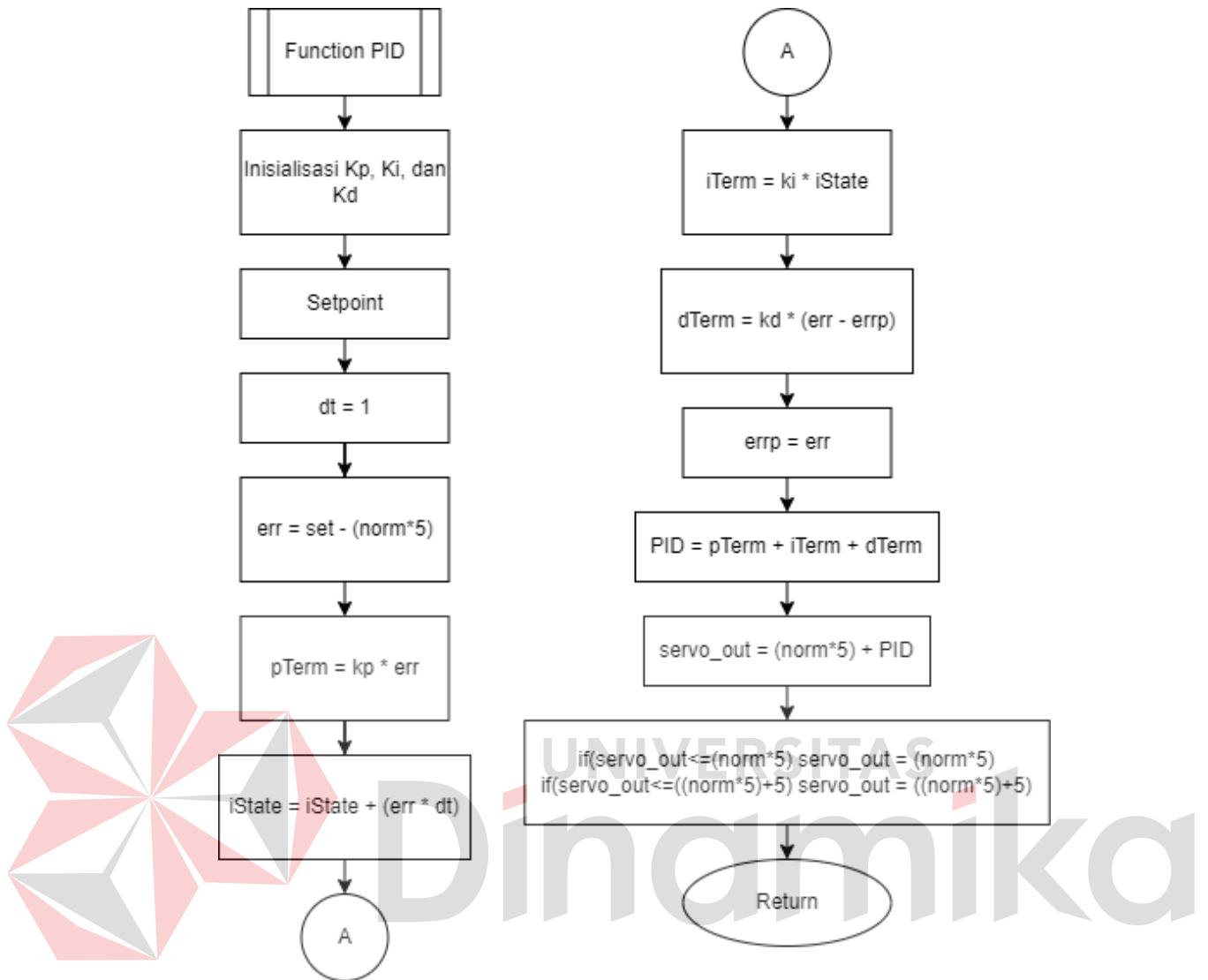
Flowchart Gambar 3.3 pada sistem, dimulai dengan deklarasi variabel pada sistem. Setelah itu sensor jarak mendapatkan hasil nilai jarak, sensor *infrared* akan melakukan perhitungan yang akan menghasilkan data kecepatan dan mengirimkan hasil pembacaan ke Arduino Uno. Nilai tersebut akan ditampilkan di LCD 16x2

yang terhubung dengan mikrokontroler. Kemudian data nilai sensor jarak dan data kecepatan akan dihitung kembali menggunakan normalisasi.

Setelah dihitung menggunakan metode normalisasi, maka aka nada variable hitung yang bernilai awal 0 ditambahkan secara berulang hingga bernilai lebih dari 6, jika sudah lebih dari 6 maka nilai variable hitung diubah menjadi 0 dan dilakukan perhitungan setpoint.

Untuk kondisi yang menentukan masuk atau tidaknya ke system PID adalah variabel normalisasi, rentang yang dibutuhkan adalah nilai normalisasi 5 sampai 10. Setiap nilai normalisasi nantinya dibagi menjadi 5 setpoint yang terdiri dari normalisasi 5 sampai dengan 6 mempunyai setpoint sebesar 30 derajat, normalisasi 6 sampai dengan 7 mempunyai setpoint sebesar 35 derajat, normalisasi 7 sampai dengan 8 mempunyai setpoint sebesar 40 derajat, normalisasi 8 sampai dengan 9 mempunyai setpoint sebesar 45 derajat, dan normalisasi lebih dari 9 mempunyai setpoint sebesar 50 derajat. Jika nilai normalisasi kurang dari 5, maka program tidak akan lanjut ke system PID. Setelah dinyatakan masuk ke system PID, PID akan memroses data yang kemudian akan menghasilkan nilai yang akan digunakan untuk mengatur derajat servo pada pengereman.





Gambar 3. 4 *Flowchart* sistem PID

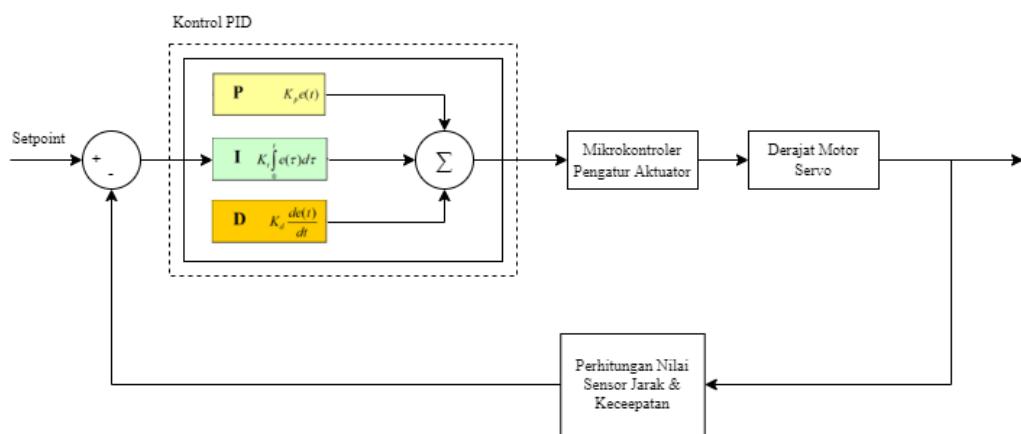
Pada Gambar 3.4 adalah *flowchart* dari sistem PID penggereman otomatis, inisialisasi variabel terlebih dahulu dilakukan di awal program. Perhitungan PID penggereman otomatis dimulai dengan mendapatkan nilai error dari penurunan nilai yang ditetapkan (variabel setpoint) dengan hasil perhitungan nilai jarak dan kecepatan yaitu normalisasi (variabel norm).

Untuk memasukkan rumus PID, harus mendapatkan integral dari nilai *error* terlebih dahulu. Setelah itu dikonversi ke persamaan PID bentuk diskrit menggunakan variabel iState. iState diperoleh dengan menambahkan nilai iState sebelumnya (nilai variabel iState pertama = 0) ke nilai kesalahan saat ini (variabel err) dikali dengan dt.

Proses penghitungan ekspresi PID dengan mengalikan nilai  $K_p$  dengan nilai  $err$  dan nilai  $K_i$  dengan nilai  $iState$ , kemudian mengalikan nilai  $K_d$  dengan selisih antara kesalahan saat ini (variabel  $err$ ) dan nilai kesalahan sebelumnya (variabel  $errp$ ). Nilai awal variabel  $errp$  diberi nilai 0 karena diasumsikan tidak ada nilai error pada saat *startup*. Setelah itu nilai  $errp$  diperbarui dengan nilai yang sama dengan  $err$ .

Setelah didapatkan nilai PID, maka nilai servo dapat diketahui dengan cara nilai normalisasi dikali dengan 5 yang terbaca sekarang di tambahkankan dengan nilai jumlah PID. Nilai tersebut dikali dengan 5 karena untuk mengimbangi nilai setpoint yang memiliki rentang 0 sampai 180 derajat dibandingkan dengan nilai normalisasi yang memiliki rentang 0 sampai 15.

Pada sistem, jika nilai servo out di bawah nilai normalisasi yang dikali 5 maka nilai servo out sama dengan nilai normalisasi dikali 5 karena sudut servo minimum dari beberapa setpoint adalah normalisasi dikali 5, jika nilai servo out diatas nilai normalisasi dikali 5 maka nilai servo out sama dengan nilai normalisasi dikali 5 karena sudut servo maksimum dari beberapa setpoint yang digunakan adalah normalisasi dikali 5. Setelah didapatkan nilai akhir dari servo out, mikrokontroler akan mengirimkan data tersebut untuk mengirimkan data yang diterima ke servo.



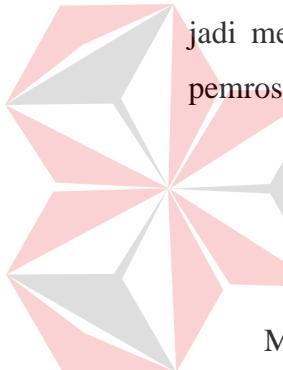
Gambar 3. 5 Blok diagram sistem PID

Pada blok diagram Gambar 3.5, sistem penggereman otomatis pada sepeda motor membutuhkan hubungan antar dua sensor atau dua nilai yang ada di Tugas Akhir ini. Nilai sensor jarak dan nilai kecepatan akan dihitung dengan normalisasi

untuk mencapai nilai yang dibutuhkan untuk menjadi setpoint sistem yang kemudian dikendalikan oleh sistem PID untuk menjaga stabilitas. Setelah mendapatkan setpoint dari perhitungan normalisasi harus menentukan perhitungan nilai Kp, Ki, dan Kd untuk mengatur keadaan normalisasi yang dihasilkan dari nilai sensor jarak dan kecepatan. Setelah proses *setting*, mikrokontroler menerima masukan berupa syarat nilai perhitungan normalisasi yang dihasilkan sensor jarak dan kecepatan yang teradapat dalam system. Di dalam system penggereman otomatis *greenhouse* jika nilai normalisasi kurang dari nilai normalisasi yang dibutuhkan system, maka servo akan bergerak sampai kondisi *setpoint* sensor tercapai.

### 3.1.4 Normalisasi Data

Normalisasi data adalah metode yang penting untuk pemrosesan data. Normalisasi membuat data yang telah didapat memiliki rentang nilai yang sama, jadi membuat data tidak ada yang terlalu besar atau kecil, sehingga membuat pemrosesan data menjadi lebih mudah.



Metode normalisasi yang dipakai di Tugas Akhir ini adalah metode normalisasi min – max, cara kerjanya nilai nilai yang terbaca dikurangi dengan nilai minimum yang telah ditentukan, kemudian dibagi dengan nilai maksimum yang telah ditentukan kemudian dikurangi dengan nilai minimum yang telah ditentukan. Metode ini dibuat hanya untuk memproses data jarak dan kecepatan yang telah terbaca oleh sensor *infrared*, setelah dilakukan normalisasi dari nilai sensor jarak dan kecepatan akan dilakukan normalisasi Kembali untuk menentukan rentang nilai yang digunakan untuk penentuan *setpoint* system PID.

$$norm = (-2.5 * (xJarak)) + (14 * xKecepatan) \quad [2]$$

Nilai nilai konstanta yang terdapat di rumus diatas didapat dari beberapa percobaan sehingga menghasilkan rentang nilai yang cocok untuk dijadikan kondisi penentuan *setpoint* system PID. Variabel normalisasi dari sensor jarak yaitu xJarak

dikali dengan *minus* 2,5 untuk mengurangi rentang nilai jarak yang memiliki nilai minimal 20 cm dan maksimal 300 cm. Rentang nilai minimal pada jarak didapatkan dari minimal pembacaan sensor jarak dan rentang nilai maksimal pada jarak didapatkan dari kondisi aman jarak antar kendaraan. Rentang nilai minimal pada kecepatan didapatkan dari kondisi berhenti pada sepeda motor dan rentang nilai maksimal pada kecepatan didapatkan dari kondisi rata – rata kecepatan pada saat kencang di keadaan jalan yang ramai. Variabel normalisasi dari kecepatan yaitu  $x$ Kecepatan dikali dengan 14 untuk menambah rentang nilai kecepatan yang memiliki rentang nilai kecepatan yang memiliki nilai minimal 0 km per jam dan maksimal 70 km per jam.

Tabel 3. 2 Rentang nilai normalisasi

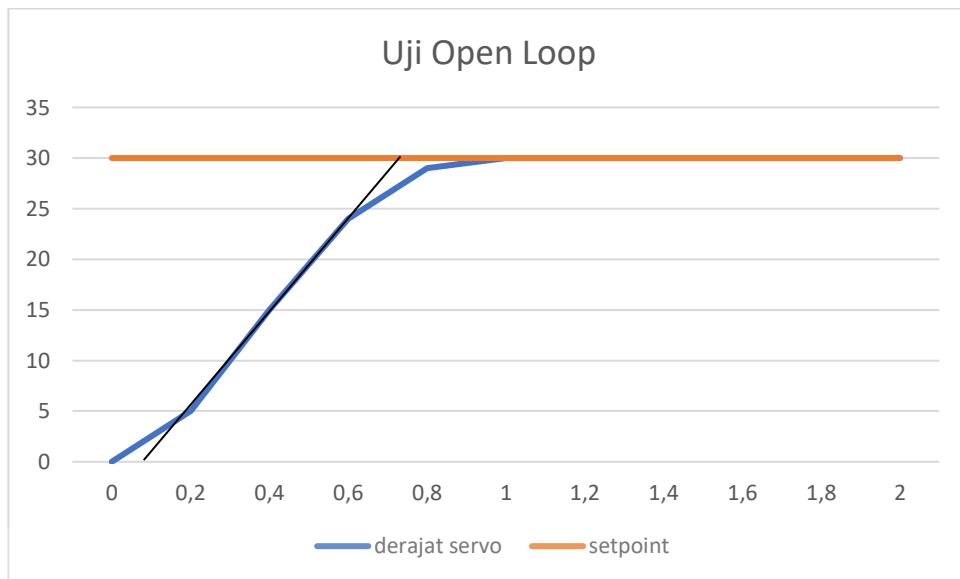
	Jarak		Kecepatan		Normalisasi penentuan PID
	Sensor	Normalisasi	Sensor	Normalisasi	
min	20	0	min	0	0
min	20	0	max	70	14
max	300	1	min	0	-2,5
max	300	1	max	70	11,5

Pada Tabel 3.2 didapatkan rentang nilai normalisasi untuk penentuan PID dari -2,5 sampai nilai 14, saat nilai normalisasi jarak dan kecepatan di nilai minimal didapatkan nilai normalisasi penentuan PID sebesar 0, saat nilai normalisasi jarak di nilai minimal dan normalisasi kecepatan di nilai maksimal didapatkan nilai normalisasi penentuan PID sebesar 14, saat nilai normalisasi jarak di nilai maksimal dan normalisasi kecepatan di nilai minimal didapatkan nilai normalisasi penentuan PID sebesar -2,5, dan saat nilai normalisasi jarak di nilai maksimal serta nilai normalisasi kecepatan di nilai maksimal didapatkan nilai normalisasi penentuan PID sebesar 11,5.

### 3.1.5 Penentuan nilai K<sub>p</sub>, K<sub>i</sub>, dan K<sub>d</sub>

Metode PID Ziegler-Nichols tipe 1 (loop terbuka) digunakan untuk menentukan nilai K<sub>p</sub>, K<sub>i</sub>, dan K<sub>d</sub> pada system ini. Untuk mendapatkan sistem kontrol yang stabil, awal mula dengan melakukan uji coba system dengan mengatur

nilai servo sebesar 30 derajat yang dimulai dari nilai servo sebesar 0 selama 2 detik, maka akan mendapatkan nilai penguatan.



Gambar 3. 6 Grafik uji *open loop* metode PID

Pada Gambar 3.6 dengan membuat garis pada nilai *setpoint* pada grafik, dari grafik tersebut didapatkan nilai parameter  $L = 0,09$  dan  $T = 0,58$  dengan setpoint sebesar 30 derajat. Setelah mendapatkan tersebut, maka dilanjutkan ke dalam rumus Ziegler- Nichols Tipe 1 dan kemudian diperoleh nilai  $K_p$ ,  $K_i$ , dan  $K_d$  sebagai berikut:

$$K_p = 1,2 \times \frac{T}{L} = 1,2 \times \frac{0,62}{0,09} = 8,27$$

$$T_i = 2 \times L = 2 \times 0,09 = 0,18$$

$$T_d = 0,5 \times L = 0,5 \times 0,09 = 0,045$$

$$K_i = K_p / T_i = 8,27 / 0,18 = 45,95$$

$$K_d = K_p \times T_d = 8,27 \times 0,045 = 0,37$$

Dari hasil hitungan rumus diatas, nilai – nilai Kp, Ki, dan Kd dimasukkan ke system penggereman otomatis.

### 3.1.6 Model Perancangan Alat

Model desain untuk alat ini ditunjukkan pada Gambar 3.7 dimana sistem penggereman otomatis memiliki komponen komponen yang digunakan, sebagai berikut:

1. Sensor Jarak JSN – SR04T
2. Sensor *Infrared*
3. Servo Motor
4. LCD 16x2



Gambar 3. 7 Kotak alat dan sensor

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

Peneliti melakukan bahasan pengujian "Pengereman Otomatis pada Sepeda Motor menggunakan Metode PID". Tes parametrik digunakan untuk melakukan analisis agar mencapai tujuan dan menjawab rumusan masalah. Pengujian parametrik dimulai dengan pengujian setiap komponen yang ditempatkan pada rangkaian. Hal ini dimaksudkan untuk memvalidasi setiap komponen agar berfungsi dengan lancar saat digunakan. Pengujian sensor meliputi pengujian sensor jarak, pengujian sensor kecepatan, dan pengujian LCD, serta pengujian komponen lainnya.

#### **4.1 Pengujian Sensor Jarak JSN-SR04T**

##### **4.1.1 Tujuan Pengujian Sensor Jarak JSN-SR04T**

Pengujian *Proximity Sensor* JSN-SR04T dilakukan untuk mengetahui keakuratan jarak yang terdeteksi oleh sensor dan jarak sebenarnya..

##### **4.1.2 Alat yang digunakan**

1. Laptop
2. Kabel USB
3. Arduino Uno
4. Sensor Jarak JSN-SR04T

##### **4.1.3 Prosedur Pengujian Sensor Jarak JSN-SR04T**

1. Menjalakan laptop dan aplikasi Arduino IDE.
2. Menghubungkan Arduino Uno ke Laptop dengan kabel USB.
3. Menghubungkan kaki - kaki sensor jarak JSN-SR04T dengan Arduino Uno.
4. Menjalankan *source code* pembacaan sensor jarak JSN-SR04T, *source code* dapat dilihat pada lampiran (1).
5. Mengumpulkan data jarak pada sensor jarak JSN-SR04T dan meteran untuk membandingkan jarak yang terbaca dari sensor dan yang terbaca secara aktual.

#### 4.1.4 Hasil Pengujian Sensor Jarak JSN-SR04T



Gambar 4. 1 Pengujian sensor jarak

Perhitungan *error* pada pengujian Sensor Jarak JSN – SR04T dengan meteran menggunakan rumus:

$$error = \left( \left| \frac{\text{Meteran} - \text{Sensor Jarak}}{\text{Meteran}} \right| \right) \times 100\% \quad [3]$$

Tabel 4. 1 Hasil Pengujian Sensor Jarak JSN-SR04T

No	Jarak		Error (%)
	Sensor (cm)	Meteran (cm)	
1	37	38	2,63
2	51	53	3,77
3	69	72	4,17
4	79	82	3,66
5	87	90	3,33
6	93	96	3,13
7	105	109	3,67
8	111	114	2,63
9	117	120	2,50
10	129	133	3,01
11	134	138	2,90
12	135	139	2,88
13	136	140	2,86
14	139	142	2,11
15	147	152	3,29
16	164	167	1,80
17	171	173	1,16
18	177	179	1,12

19	179	181	1,10
20	182	185	1,62
21	193	196	1,53
22	198	200	1,00
23	203	206	1,46
24	210	214	1,87
25	215	219	1,83
26	236	239	1,26
27	243	246	1,22
28	254	238	6,72
29	278	282	1,42
30	300	303	0,99
Rata - rata error		<b>2,42</b>	

#### 4.1.5 Analisis Data

Dari pengujian sensor jarak JSN-SR04T pada Tabel 4.1, terdapat perbedaan nilai terukur dari sensor jarak JSN-SR04T yang dilengkapi dengan alat ukur meteran. Hasil error rata-rata adalah 2,4206%.

### 4.2 Pengujian LCD 16x2

#### 4.2.1 Tujuan Pengujian LCD 16x2

Pengujian LCD 16x2 dijalankan untuk memperlihatkan tulisan pada layar LCD 16x2 yang terkoneksi ke Arduino Uno.

#### 4.2.2 Alat Yang Digunakan

1. Laptop
2. Kabel USB
3. Arduino Uno
4. LCD 16x2

#### 4.2.3 Prosedur Pengujian LCD 16x2

1. Menyalakan laptop dan membuka *software* Arduino IDE.
2. Menyambungkan Arduino Uno ke Laptop dengan kabel USB.
3. Menghubungkan kaki - kaki yang terdapat di LCD 16x2 ke Arduino Uno.
4. Menjalankan *source code* untuk memperlihatkan data di LCD 16x2, program dapat dilihat pada lampiran (3).

#### 4.2.4 Hasil Pengujian LCD 16x2



Gambar 4. 2 Hasil pengujian LCD 16x2

#### 4.2.5 Analisis Data

Dari hasil pengujian, Gambar 4.2 menunjukkan bahwa data yang dikirim oleh mikrokontroler Arduino Uno sesuai dengan apa yang ditampilkan di LCD 16x2.

### 4.3 Pengujian Sensor *Infrared*

#### 4.3.1 Tujuan Pengujian Sensor *Infrared*

Pengujian Sensor *Infrared* dilakukan untuk mendapatkan nilai kecepatan yang dibutuhkan di sistem pengerman otomatis.

#### 4.3.2 Alat Yang Digunakan

1. Laptop
2. Kabel USB
3. Arduino Uno
4. Sensor *Infrared*

#### 4.3.3 Prosedur Pengujian Sensor *Infrared*

1. Menyalakan laptop dan membuka Arduino IDE.
2. Menyambungkan Arduino Uno ke Laptop dengan kabel USB.
3. Menghubungkan kaki - kaki sensor *infrared* dengan Arduino Uno.
4. Menjalankan *source code* sensor *infrared*, *source code* terdapat pada lampiran (3).



#### 4.3.4 Hasil Pengujian Sensor *Infrared*



Gambar 4. 3 Pengujian sensor kecepatan

Perhitungan *error* pada pengujian Sensor Kecepatan yang dihasilkan dari sensor *infrared* dengan *speedometer* menggunakan rumus:

$$error = \left( \left| \frac{Speedometer - Sensor Kecepatan}{Speedometer} \right| \right) \times 100\% \quad [4]$$

Tabel 4. 2 Hasil Pengujian Sensor Kecepatan

No	Kecepatan		Error (%)
	Sensor (km/h)	Speedometer (km/h)	
1	0,18	1	82,00
2	2,15	2	7,50
3	5,67	6	5,50
4	6,36	7	9,14
5	7,04	7	0,57
6	7,34	7	4,86
7	10,07	10	0,70
8	11,25	11	2,27
9	13,01	13	0,08
10	13,69	14	2,21
11	17,31	17	1,82
12	22,69	23	1,35
13	23,57	24	1,79
14	27,68	28	1,14
15	31,49	31	1,58
16	35,11	35	0,31
17	36,87	37	0,35

18	37,85	38	0,39
19	40,1	41	2,20
20	41,56	41	1,37
21	43,72	44	0,64
22	44,79	45	0,47
23	45,77	46	0,50
24	49,68	50	0,64
25	52,81	53	0,36
26	54,2	54	0,37
27	55,84	56	0,29
28	56,53	57	0,82
29	58,68	59	0,54
30	60,44	61	0,92
Rata - rata error		<b>4,42</b>	

#### 4.3.5 Analisis Data

Dari percobaan sensor kecepatan yang dihasilkan oleh sensor infra merah pada Tabel 4.1, terdapat perbedaan nilai pembacaan sensor kecepatan yang dihasilkan oleh sensor infra merah yang dibandingkan dengan *speedometer*. Rata-rata nilai *error* adalah 4,4229%.

### 4.4 Pengujian Sistem PID

#### 4.4.1 Tujuan Pengujian

Pengujian sistem dirancang untuk menguji seluruh sistem. Pengujian ini dimaksudkan untuk memastikan bahwa semua alat yang digunakan dapat berjalan dengan sesuai. Uji seluruh sistem ini dan sesuaikan sistem penggereman otomatis.

#### 4.4.2 Alat Yang Digunakan

1. Laptop
2. Kabel USB
3. Arduino Uno
4. Sensor Jarak
5. Sensor *Infrared*
6. LCD 16x2
7. Servo Motor
8. AC
9. *Powerbank*

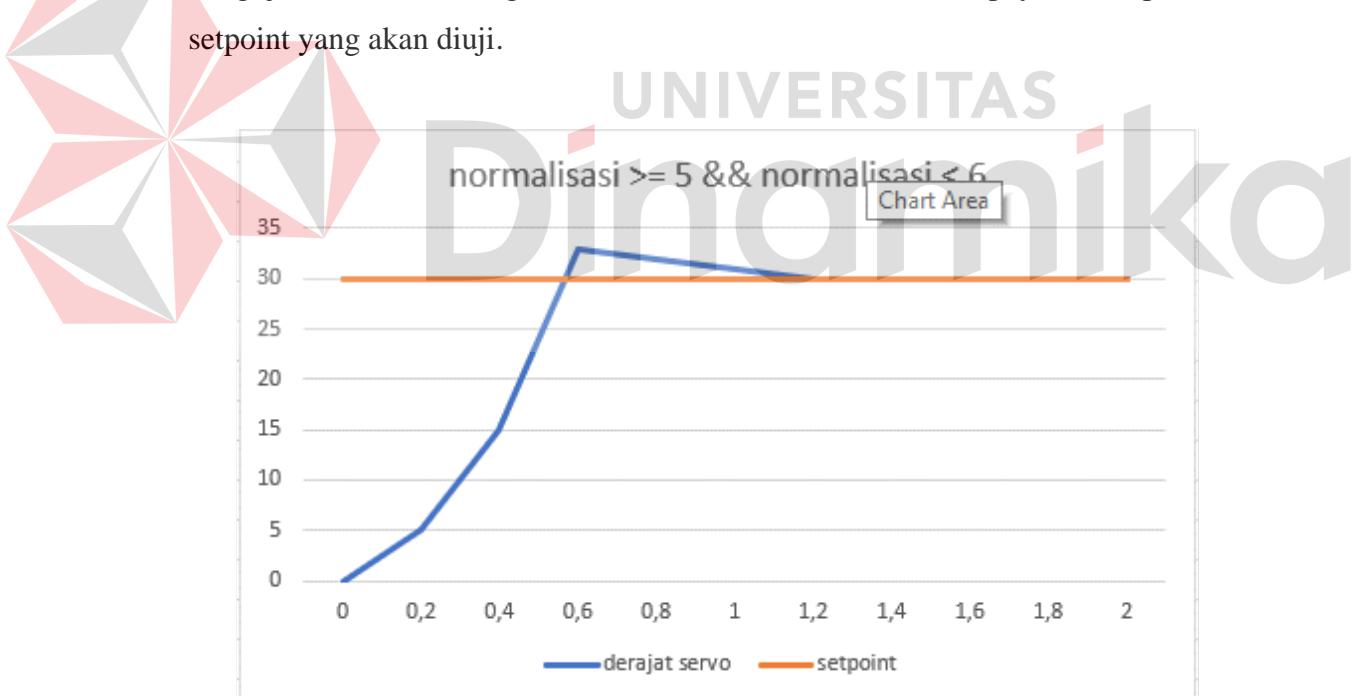
#### 4.4.3 Prosedur Pengujian

1. Menghubungkan semua komponen yang akan digunakan.
2. Menghubungkan USB dari Arduino Uno ke *Powerbank*.
3. Unggah seluruh program sistem ke Arduino Uno. Program terdapat pada lampiran (7).
4. Setelah itu, lihat data yang ditampilkan pada LCD 16x2 untuk melihat nilai sensor jarak dan kecepatan, serta *setpoint* dan derajat servo.

#### 4.4.4 Hasil Pengujian Sistem PID

Pada pengujian kestabilan dilakukan dengan menargetkan salah satu nilai normalisasi untuk mengambil nilai *setpoint* yang akan diuji. Pada pengujian ini data diuji selama 10 detik untuk melihat letak kestabilan sistem dengan menggunakan parameter yang telah di dapatkan dari metode Ziegler Nichols tipe 1.

Pada percobaan, dimasukkan nilai  $K_p = 8,27$   $K_i = 45,95$  dan  $K_d = 0,42$ . Pengujian dilakukan dengan menentukan nilai normalisasi supaya mendapat nilai *setpoint* yang akan diuji.



Gambar 4. 4 Grafik derajat PID

Untuk data dari setiap setpoint serta tabel data dapat di lihat di Lampiran 5

#### 4.4.5 Analisis Data

Percobaan system penggereman otomatis menggunakan metode PID dapat mengatur nilai servo sehingga penggereman dapat dilakukan secara baik. Pengujian

dengan menggunakan parameter  $K_p$  8,27  $K_i$  45,95  $K_d$  0,37. Sistem penggereman otomatis dengan metode PID memiliki *rise time* = 0,6 detik, *overshoot* = 3,33%, *settling time* = 0,6 detik dan *error steady state* = 2,21%.



## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Hasil pengujian kinerja alat yang telah dilaksanakan terdapat beberapa kesimpulan, yaitu:

1. Pada percobaan sensor jarak dan kecepatan memiliki rata-rata *error* sebesar 2,4% dan 4,4%.
2. Menerapkan sistem kontrol PID menggunakan metode Ziegler-Nichols tipe 1 menghasilkan nilai  $K_p$  sebesar 8,37,  $K_i$  sebesar 45,95, dan  $K_d$  sebesar 0,37. Dari grafik sistem kendali PID didapatkan *rise time* 0,6 detik, *overshoot* sebesar 3,33 %, *settling time* sebesar 0,6 detik, dan *error steady state* sebesar 2,21%.

#### **5.2 Saran**

Berikut adalah saran-saran yang dapat ditambahkan untuk melanjutkan penelitian ini.

1. Menggunakan output tambahan selain servo seperti pengatur gas secara otomatis agar dapat mendapatkan sistem penggereman yang lebih aman dan baik.
2. Menggunakan metode selain PID seperti Jaringan Syaraf Tiruan.

## DAFTAR PUSTAKA

- Arga. (2020). *Pengertian Arduino Uno dan Spesifikasinya*. Retrieved from <https://pintarelektronik.com/pengertian-arduino-uno/>
- Feni Isdaryani, M. F. (2021). Sintesis Kendali PID Digital dengan Diskritisasi . *ELKOMIKA*, 459-473.
- Fitria Suryatini, A. F. (2018). KENDALI P, PI, DAN PID ANALOG PADA PENGATURAN. *Journal of Electrical and Electronics*, 5.
- Kustanti, I. (2014). Pengendalian Kadar Keasaman (pH) Pada Sistem Hidroponik Stroberi Menggunakan Kontroler PID Berbasis Arduino Uno, 6.
- Nasrul Hamid, A. M. (2021). Penalaan Parameter PID dengan Metode Ziegler-Nichols untuk Optimasi Kontrol Kecepatan Motor pada Alat Spin Coater. *Prosiding Seminar Nasional Teknik Elektro dan Informatika*, 2-3.

manfaatnya-untuk-tubuh

P.B, P. A. (2018). Kendali Temperatur Menggunakan PID Untuk Sistem Pesteurasi Susu. Surabaya.

Reza Muhardian, K. (2020). Kendali Kecepatan Motor DC Dengan Kontroller PID dan Antarmuka. *JTEV (JURNAL TEKNIK ELEKTRO DAN VOKASIONAL)*, 3.



UNIVERSITAS  
**Dinamika**