

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

Dimas Attala Naoval¹⁾ Harianto Harianto²⁾ Musayyanah³⁾ Pauladie Susanto⁴⁾

Program Studi/Jurusan Teknik Komputer
Universitas Dinamika

Jl. Raya Kedung Baruk 98 Surabaya, 60298

Email: 1)18410200045@dinamika.ac.id, 2) hari@dinamika.ac.id, 3) musayyanah@dinamika.ac.id, 4) pauladie@dinamika.ac.id

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 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 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.

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 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 pengereman supaya pengereman 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 pengereman 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 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.

Mikrokontroler	Atmega328
Tegangan Operasi	5volt
Tegangan Rekomendasi	7-12 volt
Batasan Tegangan	6-20 volt
Pin Input/Output Digital	14
Pin Input Analog	6
Arus Pada Pin Digital	40 mA
Arus Pada Pin 3,3	50 mA
Flash Memori	32 KB (0,5 KB untuk <i>bootloader</i>)
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

LANDASAN TEORI

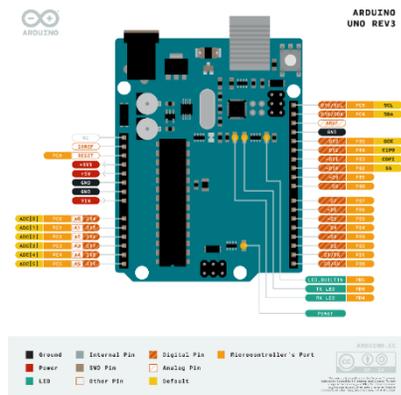
Arduino IDE

IDE adalah singkatan untuk *Interated Development Environment*. Arduino disebut lingkungan karena diprogram oleh perangkat lunak ini untuk melakukan fungsi yang dibangun ke dalam sintaks pemrograman. Arduino menggunakan bahasa pemrograman berpemilik yang mirip dengan C (Arduino, 2017). Bahasa pembrograman Arduino sudah dilakukan

perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya.

Dalam penelitian ini Arduino IDE digunakan untuk membuat program yang 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.

Arduino UNO



Gambar 1. Arduino Uno

Pada gambar 1 merupakan *Board* Arduino Uno yang menggunakan ATmega328P. Arduino Uno nantinya digunakan untuk memproses data input, output, dan metode yang digunakan di Penelitian ini. Input yang digunakan adalah sensor jarak JSN – SR04T dan sensor *infrared* yang digunakan untuk mengukur kecepatan sepeda motor. Metode yang digunakan adalah normalisasi dan metode PID, sedangkan untuk output yang digunakan adalah LCD 16x2 yang berfungsi untuk menampilkan hasil pembacaan dari sensor dan motor servo untuk mengontrol pegas rem belakang. Spesifikasi dari Arduino Uno ada di tabel 1 berikut.

Tabel 1. Spesifikasi Arduino Uno

Nama	Keterangan
VCC	3.3 to 5 VDC Supply Input
GND	Ground Input
Output	Output LOW jika halangan terdeteksi
Power LED	Menyala jika power terpasang
Obstacle LED	Menyala jika halangan terdeteksi
Kontrol Jarak	Searah jarum jam untuk menurunkan jarak, Berlawanan arah jarum jam untuk menaikkan jarak
IR Transmitter	Untuk memancarkan gelombang sinyal
IR Receiver	Untuk menerima gelombang sinyal

Sensor Infrared

Modul sensor infra merah adalah modul rangkaian yang digunakan untuk mendeteksi 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 khusus yang berfungsi untuk menerima sinar *infrared*. Pada penelitian ini sensor *infrared* digunakan sebagai pendeteksi kecepatan dengan menggunakan rumus tertentu. Gambar sensor inframerah ditunjukkan pada.



Gambar 2. Sensor Infrared

Sensor Jarak JSN – SR04T

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



Gambar 3. Sensor jarak JSN-SR04T

Sensor ini merupakan sensor jarak JSN-SR04T dengan pembacaan data yang stabil. Sensor jarak ini bekerja dengan cara memancarkan gelombang ultrasonik. Gelombang ini dipantulkan oleh suatu objek dan dideteksi oleh sensor. Dapatkan jarak antara sensor dan objek dengan menghitung waktu antara mengirim dan menerima gelombang suara.

Di sini, waktunya sangat mirip dengan sensor HC-SR04 karena merupakan waktu antara transmisi dan penerimaan gelombang suara dalam mikrodetik. Perbedaan utama antara sensor ini dan HC-SR04 kecuali kedap air adalah sensor ini hanya

menggunakan satu transduser ultrasonik. Transduser bertindak sebagai pemancar dan penerima ultrasonik. Adapun spesifikasi dari sensor JSN-SR04T ada di tabel 2 berikut.

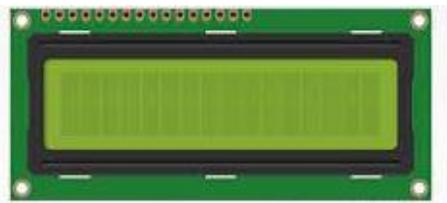
Tabel 2. Spesifikasi sensor JSN-SR04T

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

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 seperti televisi, kalkulator, ataupun layar komputer.

LCD yang digunakan pada penelitian ini adalah LCD 16x2. Artinya, LCD terdiri dari 16 kolom dan 2 baris karakter. LCD ini nantinya berfungsi sebagai penampil dari nilai nilai yang telah diproses di mikrokontroler Arduino. Gambar LCD 16x2 ditunjukkan pada gambar 4.



Gambar 4. LCD 16x2

Adapun spesifikasi dari LCD 16x2 ada di tabel 3 berikut.

Tabel 3. Pinout 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

Servo Motor

Servo motor adalah perangkat listrik yang digunakan dalam mesin industri cerdas yang dirancang untuk mendorong dan memutar objek menggunakan kontrol yang memerlukan presisi tinggi dalam hal posisi sudut, akselerasi, dan kecepatan. Pada penelitian ini servo motor digunakan untuk menggerakkan pegas rem belakang untuk mengatur berapa tekanan rem yang diperlukan oleh system yang telah dibuat. Gambar servo ditunjukkan pada gambar 5.



Gambar 5. Servo motor

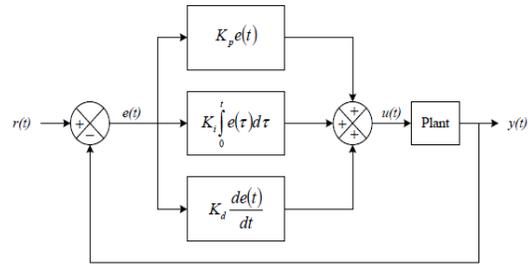
Adapun spesifikasi dari Servo Motor ditunjukkan pada tabel 4.

Tabel 4. Spesifikasi servo motor

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

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 Kp, Ki, dan Kd. (Ira Puspasari, 2018). Gambar 6 menunjukkan diagram skema dari kesalahan blok PID.



Gambar 6. Diagram blok PID

Berikut penjelasan blok diagram PID pada gambar 6:

Nilai $r(t)$: Nilai *setpoint*.

Nilai $y(t)$: Nilai hasil outputan sistem PID sampai sama dengan nilai *setpoint*.

Nilai $e(t)$: Nilai *error*.

Nilai $u(t)$: Penjumlahan dari hasil Kp, Ki, Kd.

Konstanta PID diubah dahulu ke bentuk persamaan diskrit sebagai berikut:

Bentuk Kontinu

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (1)$$

Keterangan rumus PID bentuk kontinu:

$u(t)$: sinyal kontrol

Kp: gain propotional

Ki: integral waktu

Td: diferensial waktu

$e(t)$: sinyal error

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*.

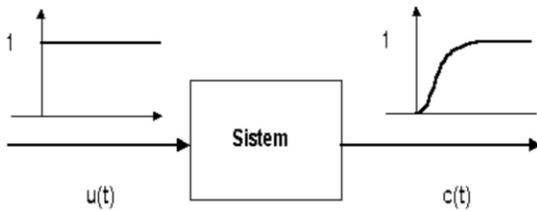
$e(t_{k-1})$: *error* sebelumnya.

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.

Ziegler-Nichols Tipe 1 (Open Loop)

Untuk mendapatkan parameter Kp, Ki, dan Kd 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 7.



Gambar 7. Respon unit step

Fungsi kurva S adalah untuk menentukan nilai jeda waktu (L) dan konstanta waktu (T). Proses penentuan kurva L dan T ditunjukkan pada gambar 8, dari nilai L dan T tersebut nilai – nilai parameter dapat diperoleh.



Gambar 8. Kurva respon berbentuk S

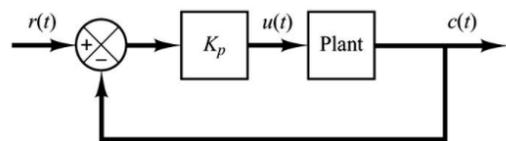
Setelah mendapatkan nilai L dan T, perhitungan parameter PID menggunakan aturan tuning (Anita Puput Widiastuti, 2016) pada tabel 5.

Tabel 5. Aturan tuning Ziegler-Nichols tipe open loop

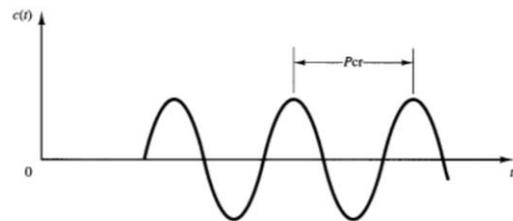
Type Kontroler	Kp	Ti	Td	Ki	Kd
P	T/L	∞	0		
PI	$0.9 T/L$	$L/0.3$	0		
PID	$1.2 T/L$	$2L$	$0.5L$	Kp/Ti	$Kp \times Td$

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 proporsional (K_p) untuk menggetarkan sistem dengan mengatur nilai K_p dari 0 hingga nilai kritis K_{cr} . Sebuah plot getaran terus menerus ditunjukkan pada gambar 9. Metode ini tidak berlaku jika sistem tidak berosilasi.



Gambar 9. Blok diagram Ziegler-Nichols tipe 2



Gambar 10. Proses oscilasi parameter Pcr

Perhitungan PID menggunakan rumus pada tabel 6.

Tabel 6. Aturan Ziegler-Nichols tipe 2

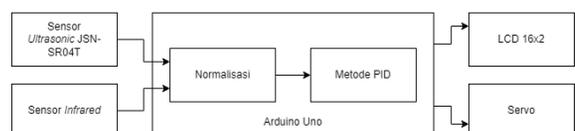
Type Kontroler	Kp	Ti	Td	Ki	Kd
P	$0.5 K_{cr}$	∞	0		
PI	$0.45 K_{cr}$	$\frac{1}{1.2} P_{cr}$	0		
PID	$0.6 K_{cr}$	$0.5 P_{cr}$	$0.125 P_{cr}$	$\frac{2}{T_i} \times K_p$	$\frac{K_p}{T_d}$

METODE PENELITIAN

Metode penelitian yang digunakan dalam penelitian 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.

Rancangan Sistem

Berikut ini adalah diagram blok dari sistem yang dibuat. Gambar diagram blok ditunjukkan pada 11.



Gambar 11. Blok diagram sistem

Pada gambar 11 diagram blok sistem di atas, setiap bagian terhubung ke Arduino Uno memiliki fungsi:

1. Input

Sensor *Ultrasonic* JSN – SR04T Sensor ini digunakan untuk pengukuran jarak antar kendaraan. Sensor *Infrared* Komponen ini berfungsi untuk mendeteksi kondisi yang digunakan untuk perhitungan kecepatan kendaraan.

2. Proses

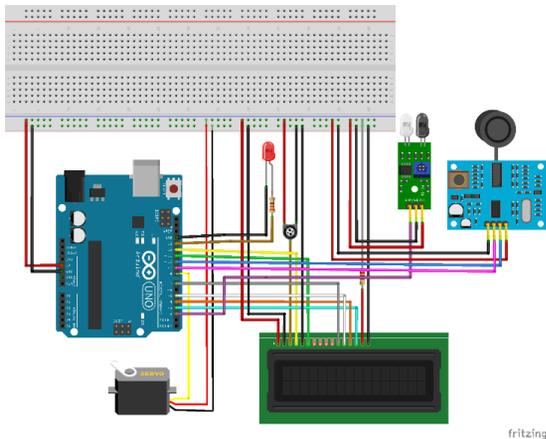
Arduino Uno Alat ini berfungsi untuk mengontrol *input* dan *output* pada sistem pengereman otomatis. Normalisasi digunakan untuk menghitung data dari sensor jarak dan kecepatan menjadi satu nilai normalisasi, hasil normalisasi nanti yang digunakan untuk setpoint dari PID. Metode PID Kontroler PID digunakan untuk menghitung perbandingan nilai input yang diterima menggunakan parameter K_p , K_i , dan K_d yang diberikan dan menghasilkan output yang diinginkan.

3. Output

Servo Motor Komponen ini bertindak sebagai pengatur tekanan rem. LCD 16x2 berfungsi untuk menampilkan kondisi jarak dan kecepatan yang telah diproses di arduino.

Rangkaian Skematik

Rangkaian pada perangkat yang dibuat dapat dilihat pada gambar 12.



Gambar 12. Rangkaian skematik sistem

Keterangan kaki-kaki komponen ada pada gambar 12 terdapat pada tabel 7.

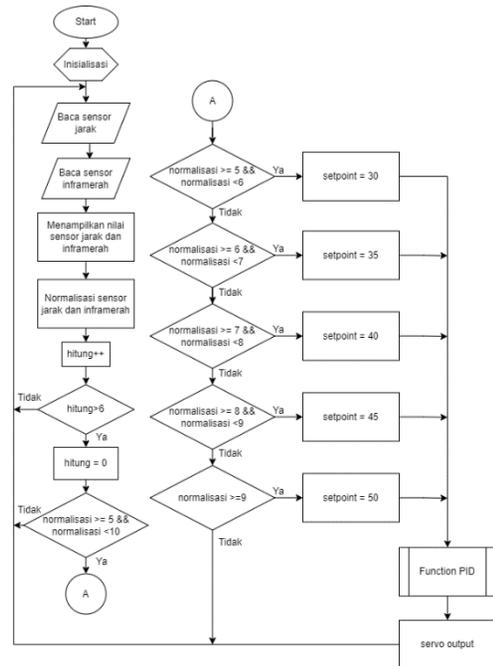
Tabel 7. Pinout komponen ke Arduino Uno

No	Komponen	Pin Out
1	Sensor <i>Infrared</i>	D2

No	Komponen	Pin Out
2	Sensor <i>Ultrasonic</i>	TRIG(D10) dan ECHO(D9)
3	LED	D13
4	Servo	D6
5	LCD 16x2	RS(12), EN(11), D7(3), D6(4), D5(5), D6(7)

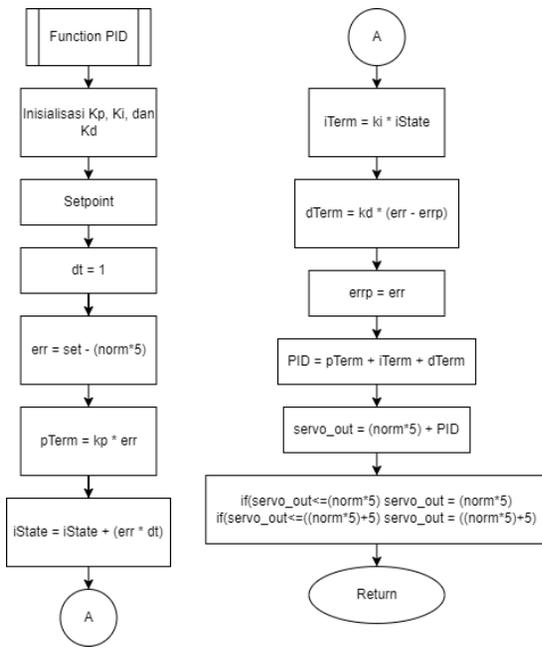
Flowchart Sistem

Diagram alir keseluruhan untuk sistem ini dapat dijelaskan sebagai berikut:



Gambar 13. Flowchart sistem pengereman otomatis

Flowchart gambar 13 pada sistem, dimulai dengan inisialisasi variabel pada sistem. Kemudian sensor jarak mendapatkan hasil pembacaan jarak, sensor *infrared* melakukan perhitungan yang menghasilkan data kecepatan dan mengirimkan hasil pembacaan ke Arduino Uno. Nilai tersebut ditampilkan di LCD 16x2 yang terhubung dengan mikrokontroler. Kemudian data nilai sensor jarak dan data kecepatan dihitung kembali menggunakan normalisasi yang kemudian hasil dari normalisasi tersebut yang menentukan setpoint dan masuk atau tidaknya nilai tersebut ke sistem PID.



Gambar 14. Flowchart sistem PID

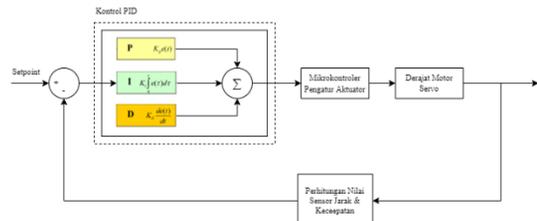
Pada gambar 14 adalah *flowchart* dari sistem PID pengereman otomatis, nisialisasi variabel terlebih dahulu dilakukan di awal program. Perhitungan PID pengereman otomatis dimulai dengan mendapatkan nilai error dari penurunan nilai yang ditetapkan (variabel setpoint) dengan hasil perhitungan nilai jarak dan kecepatan yaitu normalisasi (variabel norm).

Untuk memasukkan perhitungan PID, harus terlebih dahulu mendapatkan integral dari nilai *error*. Setelah itu dikonversi ke persamaan diskrit (2) menggunakan variabel *iState*. *iState* diperoleh dengan menambahkan nilai *iState* sebelumnya (nilai variabel *iState* pertama = 0) ke nilai kesalahan saat ini (variabel *err*) dikalikan dengan *dt*.

Proses penghitungan ekspresi PID dengan mengalikan nilai *Kp* dengan nilai *err* dan nilai *Ki* dengan nilai *iState*, kemudian mengalikan nilai *Kd* dengan selisih antara *error* saat ini (variabel *err*) dan nilai *error* sebelumnya (variabel *errp*). Nilai awal variabel *errp* diberi nilai 0 karena diasumsikan tidak ada nilai error pada saat *startup*. Kemudian 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 saat ini dijumlahkan dengan nilai hasil PID. 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 mendapatkan nilai akhir dari servo out, mikrokontroler kemudian mengirimkan data tersebut untuk mengirimkan data yang diterima ke servo.



Gambar 15. Blok diagram sistem PID

Pada blok diagram gambar 15 system pengereman otomatis pada sepeda motor membutuhkan hubungan antar dua sensor atau dua nilai yang ada di penelitian ini. Nilai sensor jarak dan nilai kecepatan 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 pengereman otomatis *greenhouse* jika nilai normalisasi kurang dari nilai normalisasi yang dibutuhkan system, maka servo bergerak sampai kondisi *setpoint* sensor tercapai.

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.

$$x_{new} = \frac{x_{old} - x_{min}}{x_{max} - x_{min}}$$

Metode normalisasi yang dipakai di Penelitian ini adalah metode normalisasi min – max, cara kerjanya nilai nilai yang terbaca

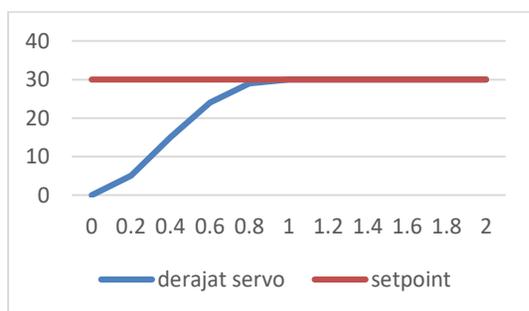
dikurangi dengan nilai minimum yang sudah ditetapkan, kemudian dibagi nilai maksimum yang telah ditentukan kemudian dikurangi dengan nilai minimum yang telah ditentukan. Metode ini dibuat hanya untuk memproses data jarak dan kecepatan yang telah terbaca oleh sensor *infrared*, setelah dilakukan normalisasi dari nilai sensor jarak dan kecepatan kemudian dilakukan normalisasi Kembali untuk menentukan rentang nilai yang digunakan untuk penentuan *setpoint* system PID.

$$norm = (-2.5 * (xJarak)) + (14 * xKecepatan)$$

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. Variabel normalisasi dari kecepatan yaitu xKecepatan 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.

Penentuan nilai Kp, Ki, dan Kd

Metode PID Ziegler-Nichols tipe 1 (loop terbuka) digunakan untuk menentukan nilai Kp, Ki, dan Kd 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 mendapatkan nilai penguatan.



Gambar 16. Uji open loop sistem

Dengan menarik garis pada nilai *setpoint* pada kurva, dari kurva tersebut didapatkan nilai parameter $L = 0,09$ dan $T = 0,62$ dengan *setpoint* sebesar 30 derajat. Setelah mendapatkan tersebut, maka dilanjutkan ke dalam rumus Ziegler- Nichols

Tipe 2 dan kemudian diperoleh nilai Kp, Ki, dan Kd sebagai berikut:

$$K_p = 1.2 \times \frac{T}{L} = 1.2 \times \frac{0,62}{0,09} = 8.27$$

$$T_i = 2 \times L = 2 \times 0.09 = 0.18$$

$$T_d = 0.5 \times L = 0.5 \times 0.09 = 0.045$$

$$K_i = \frac{K_p}{T_i} = \frac{8.27}{0.18} = 45.95$$

$$K_d = K_p \times T_d = 8.27 \times 0.045 = 0.37$$

Dari hasil hitungan rumus diatas, nilai – nilai Kp, Ki, dan Kd dimasukkan ke system pengereman otomatis. Proses perhitungan dihitung dengan menggunakan respon system *open loop*.

Model Perancangan Alat

Model desain untuk alat ini ditunjukkan pada gambar 17 dimana sistem pengereman otomatis memiliki beberapa komponen penting, sebagai berikut:

1. Sensor Jarak JSN – SR04T
2. Sensor *Infrared*
3. Servo Motor
4. LCD 16x2
- 5.



Gambar 17. Kotak alat dan sensor

HASIL DAN PEMBAHASAN

Peneliti membahas pengujian ”Pengereman Otomatis pada Sepeda Motor menggunakan Metode PID”. Tes parametrik digunakan untuk melakukan analisis untuk 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 baik saat digunakan. Pengujian komponen meliputi pengujian sensor jarak, pengujian sensor

kecepatan, pengujian LCD, dan pengujian komponen lainnya.

Pengujian Sensor Jarak JSN-SR04T

Tujuan Pengujian Sensor Jarak JSN-SR04T. Pengujian Sensor Jarak JSN-SR04T dilakukan untuk mengetahui tingkat akurasi jarak yang dideteksi dari sensor dan jarak secara aktual.

Alat yang digunakan:

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

Prosedur pengujian sensor jarak JSN-SR04T:

1. Menjalakan laptop dan menjalankan aplikasi Arduino IDE.
2. Menghubungkan Arduino Uno ke Laptop menggunakan kabel USB.
3. Mengubungkan kaki - kaki sensor jarak JSN-SR04T ke Arduino Uno.
4. Menjalakan *source code* pembacaan sensor jarak JSN-SR04T.
5. Mengambil data jarak pada sensor jarak JSN-SR04T dan meteran untuk membandingkan jarak yang terbaca dari sensor dan yang terbaca secara aktual.

Hasil pengujian sensor jarak JSN-SR04T



Gambar 18. Pengujian sensor jarak

Pada gambar 18 merupakan cara pengujian jarak menggunakan sensor dan meteran sebagai pembanding. Perhitungan *error* pada pengujian Sensor Jarak JSN – SR04T dengan meteran menggunakan rumus:

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

Tabel 8. 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

No	Jarak		Error (%)
	Sensor (cm)	Meteran (cm)	
4	79	82	3,66
5	87	90	3,33
.	.	.	.
.	.	.	.
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			2,42

Analisis Data

Dari percobaan sensor jarak JSN-SR04T pada tabel 8 terdapat perbedaan nilai terukur dari sensor jarak JSN-SR04T yang dilengkapi dengan alat ukur meteran. Hasil error rata-rata adalah 2.4206%.

Pengujian LCD 16x2

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

Alat yang digunakan:

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

Prosedur pengujian LCD 16x2:

1. Menghidupkan laptop dan membuka *software* Arduino IDE.
2. Menghubungkan Arduino Uno ke Laptop menggunakan kabel USB.
3. Mengubungkan kaki - kaki yang terdapat di LCD 16x2 ke Arduino Uno.
4. Menjalankan *source code* untuk menampilkan data di LCD 16x2.

Hasil pengujian LCD 16X2



Gambar 19. Hasil pengujian LCD 16x2

Analisis Data

Dari hasil pengujian gambar 19 menunjukkan bahwa data yang dikirim oleh mikrokontroler Arduino Uno sesuai dengan apa yang ditampilkan di LCD 16x2.

Pengujian Sensor Infrared

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

Alat yang digunakan:

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

Prosedur pengujian sensor infrared:

1. Menghidupkan laptop dan membuka Arduino IDE.
2. Menghubungkan Arduino Uno ke Laptop menggunakan kabel USB.
3. Mengubungkan kaki - kaki modul sensor *infrared* dengan Arduino Uno.
4. Menjalankan *source code* sensor *infrared*.

Hasil Pengujian Sensor *Infrared*



Gambar 20. Pengujian sensor infrared

Pada gambar 20 merupakan cara pengujian sensor infrared yang dibandingkan dengan *speedometer* motor. Perhitungan *error* pada pengujian Sensor Kecepatan yang dihasilkan dari sensor *infrared* dengan *speedometer* menggunakan rumus:

$$error = \left(\left| \frac{Speedometer - \text{Sensor Kecepatan}}{Speedometer} \right| \right) \times 100\%$$

Tabel 9. Hasil pengujian sensor infrared

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
.	.	.	.
.	.	.	.
.	.	.	.
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			4,42

Analisis Data

Dari percobaan sensor kecepatan yang dihasilkan dari sensor *infrared* pada tabel 9, terdapat perbedaan nilai pembacaan sensor kecepatan yang dihasilkan oleh sensor infra merah yang dibandingkan dengan *speedometer*. Rata-rata nilai *error* adalah 4.4229%.

Pengujian Sistem Otomasi

Pengujian sistem dirancang untuk menguji seluruh sistem. Pengujian ini dimaksudkan untuk memastikan bahwa semua komponen berjalan dengan baik. Uji seluruh sistem ini dan sesuaikan sistem pengereman otomatis.

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*

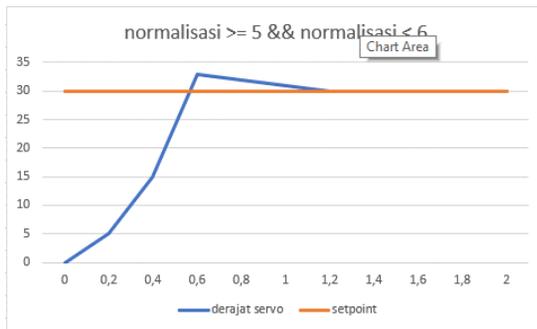
Prosedur pengujian:

1. Menyambungkan semua perangkat
2. Menghubungkan *powerbank* pada alat ke Arduino Uno.
3. Meng-upload program keseluruhan sistem ke dalam Arduino Uno.
4. Setelah selesai, melihat data yang ditampilkan pada LCD 16x2 untuk melihat nilai sensor jarak dan kecepatan, serta *setpoint* dan derajat servo.

Hasil Pengujian

Pada pengujian kestabilan dilakukan dengan menargetkan salah satu nilai normalisasi untuk mengambil nilai *setpoint* yang 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 $K_d = 0.37$. Pengujian dilakukan dengan mengatur nilai normalisasi supaya mendapat nilai *setpoint* yang diuji.



Gambar 21. Grafik pengujian Sistem

Analisis Data

Percobaan system pengereman otomatis menggunakan metode PID dapat mengatur nilai servo, sehingga pengereman dapat dilakukan secara baik. Pengujian dengan menggunakan parameter K_p 8.27 K_i 45.95 K_d 0.37. Sistem pengereman otomatis dengan metode PID memiliki $rise\ time = 0.6$ detik, $overshoot = 3.33\%$, $settling\ time = 0.6$ detik dan $error\ steady\ state$ 2.12%.

KESIMPULAN

Hasil pengujian kinerja alat yang dilakukan 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 system 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%.

Saran

Berikut adalah beberapa saran yang ditambahkan untuk melanjutkan penelitian ini.

1. Menggunakan output tambahan selain servo seperti pengatur gas secara otomatis agar dapat mendapatkan system pengereman 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://pintarelektro.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*. Suarabaya.
- Reza Muhandian, K. (2020). Kendali Kecepatan Motor DC Dengan Kontroller PID dan Antarmuka. *JTEV (Jurnal Teknik Elektro Dan Vokasional)*, 3.