

PROJECT AKHIR SISTEM AUTOMASI SUMO ROBOT



Disusun oleh :
Kelompok 6

Ruasa Azizan Zihni	235150301111046
Barru Wira Yasa	235150301111021
Ibadurahman Faiz Usman	235150301111032
Rayhan Sulistyawan	235150301111019
Dzaki Rabbani	215150307111021
M. Radhi Rasyidi Rafli	235150307111041

Mata Kuliah Sistem Automasi
Dosen Pengampu : Sabriansyah Rizqika Akbar, S.T., M.Eng., Ph.D

**PROGRAM STUDI TEKNIK KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2025**

DAFTAR ISI

ABSTRAK	3
BAB 1	4
PENDAHULUAN	4
1.1 Latar Belakang	4
1.2. Rumusan Masalah	5
1.3 Tujuan	5
1.4 Manfaat Program	5
BAB 2	6
PRINSIP KERJA	6
2.1 Cara Kerja	6
2.2 Input Sensor	6
2.3 Hasil Yang Diharapkan	6
BAB 3	8
PROTOTIPE SISTEM	8
3.1 Alat dan Bahan	8
3.2 Block Diagram	16
1. Input — Jarak Objek	17
2. Deteksi Error — Baca Sensor Jarak	17
3. PID Controller — Arduino (PID Controller)	17
4. Plant — Motor DC	17
5. Output — Pergerakan Motor	17
6. Umpan Balik — Sensor Jarak	17
3.3 Source Code	17
3.4 Flowchart	21
BAB 4	22
KESIMPULAN	22
4.1 Pembagian Kinerja Kelompok	22
4.2 Kesimpulan	22
4.3 Daftar Pustaka	23
REFLEKSI AKHIR MATA KULIAH SISTEM AUTOMASI	24

ABSTRAK

Pada proyek ini telah dirancang dan diimplementasikan sebuah sistem kontrol berbasis PID (Proportional-Integral-Derivative) menggunakan mikrokontroler ESP32 untuk mengendalikan pergerakan motor DC agar dapat mempertahankan jarak objek sesuai dengan nilai setpoint yang diinginkan. Sistem ini memanfaatkan sensor jarak sebagai umpan balik guna mendeteksi error atau selisih antara jarak aktual dengan jarak target. Nilai error yang diperoleh digunakan sebagai input bagi pengendali PID yang kemudian menghitung sinyal koreksi untuk mengatur kecepatan motor DC. Dengan menggunakan ESP32, sistem ini mendukung pemrosesan data secara real-time serta kemudahan dalam komunikasi nirkabel, sehingga mempermudah pemantauan dan pengendalian jarak secara dinamis. Implementasi PID pada ESP ini bertujuan untuk menghasilkan respons sistem yang cepat, akurat, dan stabil. Hasil pengujian menunjukkan bahwa sistem kontrol PID yang dirancang mampu menjaga jarak objek dengan akurasi yang memadai, serta menunjukkan performa yang lebih baik dibandingkan pengendali sederhana lainnya.

Video Presentasi: <https://www.youtube.com/watch?v=8qFzm3vdtHY>

Video Demo Alat: <https://youtu.be/-q4ZjlIR-U4?si=6y-QIw76LEkfft7m>

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Mobil otonom atau mobil tanpa pengemudi kini mulai menjadi inovasi yang potensial dalam mengurangi angka kecelakaan lalu lintas serta kemacetan di berbagai belahan dunia. Mobil yang sepenuhnya otonom idealnya dapat bergerak dan mengambil keputusan dalam segala kondisi tanpa perlu intervensi dari pengemudi manusia. Walaupun saat ini mobil dengan otonomi penuh belum sepenuhnya terealisasi, banyak mobil modern sudah dilengkapi berbagai fitur bantuan mengemudi, seperti sistem pengereman otomatis, peringatan keluar jalur, hingga cruise control adaptif. Beberapa kendaraan bahkan sudah mampu mengemudi sendiri dalam kondisi tertentu, misalnya di jalan tol yang memiliki marka jalur yang jelas. Tesla menjadi salah satu contoh perusahaan otomotif yang terkenal dengan pengembangan fitur-fitur mengemudi otonom.

Seperti halnya manusia yang menggunakan indra untuk memahami kondisi sekitar saat berkendara, mobil otonom juga memanfaatkan berbagai sensor elektronik untuk memantau jalan dan lingkungan di sekitarnya. Kamera dipadukan dengan sistem pengolahan citra komputer untuk mengenali objek visual, sementara sensor radar dan ultrasonik digunakan untuk mengukur jarak antara mobil dengan objek lain. Berdasarkan data dari sensor-sensor tersebut, mobil dapat mengambil keputusan secara mandiri, tanpa perlu dikendalikan oleh manusia. Misalnya, ketika sensor mendeteksi kendaraan di depan yang mendadak melambat, mobil secara otomatis akan mengaktifkan sistem pengereman untuk mencegah tabrakan. Atau ketika kendaraan mulai keluar dari jalurnya, sistem akan mengoreksi arah setir agar mobil tetap berada di jalur yang benar.

Pada proyek ini, akan dibuat sebuah robot sederhana berbasis Arduino yang dapat bergerak secara otonom menggunakan sensor ultrasonik untuk mendeteksi jarak terhadap hambatan di depannya. Robot ini akan diprogram agar dapat berhenti secara otomatis jika mendekati rintangan dengan jarak tertentu.

Pengujian dilakukan dalam dua skenario berbeda. Pada skenario pertama, robot akan secara otomatis berhenti saat mendeteksi rintangan. Sedangkan pada skenario kedua, robot akan berhenti setelah mendapatkan perintah dari manusia. Pada masing-masing skenario, akan dicatat seberapa jauh robot bergerak sebelum benar-benar berhenti.

Pengujian dilakukan dalam dua skenario berbeda. Pada skenario pertama, robot akan secara otomatis berhenti saat mendeteksi rintangan. Sedangkan pada skenario kedua, robot akan berhenti setelah mendapatkan perintah dari manusia. Pada masing-masing skenario, akan dicatat seberapa jauh robot bergerak sebelum benar-benar berhenti.

1.2. Rumusan Masalah

1. Bagaimana cara membuat sistem penggereman otomatis pada mobil robot sederhana untuk menghindari tabrakan?
2. Bagaimana sensor ultrasonik dapat mendeteksi jarak objek di depan mobil dan mengaktifkan rem secara mandiri?
3. Faktor-faktor apa saja yang mempengaruhi kecepatan respons sistem penggereman otomatis?

1.3 Tujuan

Proyek ini bertujuan untuk merancang sekaligus mengimplementasikan sistem penggereman otomatis pada robot mobil dengan memanfaatkan sensor ultrasonik. Selain itu, proyek ini juga bertujuan untuk memahami cara kerja sensor jarak dan bagaimana aplikasinya dalam teknologi kendaraan otonom. Untuk menguji efektivitas sistem yang dibuat, robot akan diuji dengan variasi kecepatan gerak dan jarak terhadap rintangan guna mengevaluasi keandalan performa penggereman otomatisnya.

1.4 Manfaat Program

1. Mengenalkan konsep dasar teknologi kendaraan otonom dan dasar-dasar pemrograman robotika secara sederhana
2. Memberikan contoh nyata bagaimana sistem keselamatan modern seperti Automatic Emergency Braking (AEB) bekerja di kendaraan sungguhan.
3. Melatih kemampuan dalam pemecahan masalah, pemrograman, elektronik melalui kegiatan praktik langsung (hands-on).
4. Memberikan peluang pengembangan lebih lanjut, seperti penambahan fitur kontrol kecepatan atau sistem penghindaran rintangan.

BAB 2

PRINSIP KERJA

2.1 Cara Kerja

Proyek ini mengembangkan sebuah mobil robotik yang mampu melakukan pengereman secara otomatis maupun manual. Sistem ini menggunakan sensor ultrasonik yang dipasang di bagian depan mobil untuk mendeteksi jarak terhadap objek atau halangan di depannya.

Ketika sensor mendeteksi objek dalam jarak tertentu, terdapat dua mode pengereman yang dapat dilakukan. Pada mode *automatic braking*, mobil akan berhenti secara otomatis tanpa intervensi dari pengguna. Sedangkan pada mode *manual braking*, lampu LED akan menyala sebagai peringatan dan pengguna harus menekan tombol pada remote untuk menghentikan mobil secara manual.

Mobil ini dikendalikan oleh mikrokontroler ESP32 yang bertugas membaca data dari sensor ultrasonik dan mengontrol motor penggerak melalui IC L293D H-Bridge. Pada mode manual, sistem dilengkapi dengan remote berisi tombol untuk mengaktifkan fungsi pengereman secara manual.

2.2 Input Sensor

Sensor yang digunakan dalam sistem ini adalah Ultrasonic Distance Sensor (HC-SR04), yang berfungsi untuk membaca jarak antara mobil dan objek di depannya. Proses pembacaan dilakukan dengan cara mengirimkan gelombang ultrasonik dan mengukur waktu pantulannya, sehingga jarak ke objek dapat dihitung secara akurat. Jika jarak yang terdeteksi lebih kecil dari batas yang telah ditentukan, yaitu 30 cm secara default, maka sistem akan merespons sesuai mode yang digunakan. Dalam mode otomatis, sistem akan langsung mengaktifkan pengereman tanpa memerlukan intervensi pengguna. Sementara itu, pada mode manual, sistem hanya menyalakan lampu LED sebagai peringatan dan menunggu pengguna menekan tombol untuk menghentikan kendaraan.

2.3 Hasil Yang Diharapkan

1. Pada Mode Automatic Braking:

Dalam mode ini, mobil akan secara otomatis berhenti ketika mendeteksi adanya objek dalam jarak tertentu, tanpa memerlukan intervensi atau keterlibatan pengguna. Sistem menggunakan sensor ultrasonik untuk mengukur jarak, dan jika objek berada di bawah ambang batas yang telah ditentukan, sistem akan langsung mengaktifkan pengereman guna mencegah tabrakan.

2. Pada Mode Manual Braking:

Pada mode manual, ketika sensor mendeteksi objek dalam jarak tertentu, sistem akan memberikan peringatan kepada pengguna dengan menyalakan lampu LED. Mobil tidak langsung berhenti, melainkan menunggu pengguna untuk menekan tombol pada remote sebagai perintah untuk melakukan pengereman.

Sistem ini juga dilengkapi dengan fitur untuk menghitung jarak berhenti (stopping distance) berdasarkan respons pengereman yang terjadi. Selain itu, dilakukan perbandingan performa antara pengereman otomatis dan manual untuk mengetahui efektivitas masing-masing mode dalam menghentikan kendaraan secara cepat dan aman.

4.4 Motor DC

Motor DC (Direct Current) adalah jenis aktuator elektromekanis yang mengubah energi listrik searah menjadi energi gerak putar. Prinsip kerja motor DC didasarkan pada gaya Lorentz, di mana arus listrik yang mengalir melalui kumparan dalam medan magnet menghasilkan torsi yang menyebabkan rotor berputar. Secara umum, motor DC terdiri dari komponen utama seperti stator (penyedia medan magnet), rotor (armature), sikat (brush), dan komutator.

Untuk menganalisis perilaku dinamis motor DC, digunakan model matematis dalam bentuk fungsi alih (transfer function). Fungsi alih motor DC yang sering digunakan adalah hubungan antara tegangan input $V_a(s)$ dan kecepatan sudut output $\omega(s)$, yang dirumuskan sebagai:

$$\frac{\omega(s)}{V_a(s)} = \frac{K}{(Js+b)(Ls+)+k^2} =$$

di mana:

- K adalah konstanta motor (torsi dan back-EMF),
- J adalah momen inersia rotor,
- b adalah koefisien friksi viskos,
- L adalah induktansi armatur,
- R adalah resistansi armatur.

Dalam banyak kasus, jika L kecil (diabaikan), fungsi alih dapat disederhanakan menjadi orde satu:

$$\frac{\omega(s)}{V_a(s)} = \frac{K}{JsR + bR + K^2}$$

Respon sistem motor DC terhadap masukan step (tegangan konstan) akan menunjukkan karakteristik sistem orde satu, yaitu eksponensial naik hingga mencapai keadaan tunak. Bentuk umum respon waktunya adalah:

$$\omega(t) = \omega_{ss}(1 - e^{-t/\tau})$$

di sini ω_{ss} adalah kecepatan steady-state dan τ taut adalah konstanta waktu sistem yang menunjukkan seberapa cepat sistem mencapai 63% dari nilai akhirnya. Respons ini penting untuk merancang sistem kendali seperti PID agar motor dapat dikontrol dengan stabil dan cepat sesuai kebutuhan aplikasi, seperti robot, aktuator otomasi, atau kendaraan listrik skala kecil.

udah mas

BAB 3

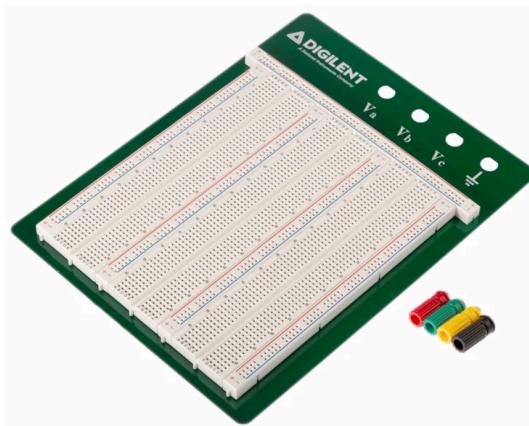
PROTOTIPE SISTEM

3.1 Alat dan Bahan

Komponen yang digunakan dalam proyek ini adalah sebagai berikut:

1. Breadboard

Breadboard adalah papan tempat kita dapat merangkai rangkaian elektronik sementara tanpa perlu menyolder. Ini sangat berguna dalam tahap prototipe, di mana kita mungkin perlu mengubah atau menambahkan komponen dengan cepat. Dengan breadboard, kita dapat menghubungkan berbagai komponen menggunakan kabel jumper, dan membuat koneksi yang fleksibel dan mudah diubah. Dalam proyek ini, dua breadboard digunakan untuk memberikan ruang yang cukup untuk menghubungkan semua komponen yang diperlukan, seperti ESP32, motor driver, sensor, LED, dan regulator.

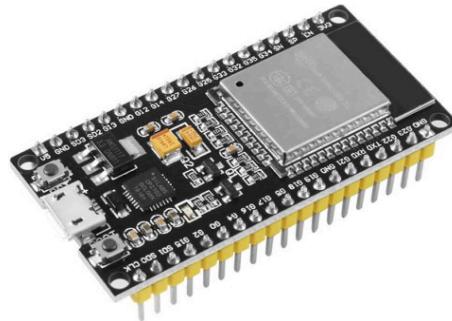


Gambar 3.1 Breadboard

Sumber: <https://www.rs-online.id/p/solderless-breadboard-kit-large/>

2. ESP32

ESP32 adalah mikrokontroler yang menjadi otak dari robot ini. Mikrokontroler ini memiliki kemampuan pemrosesan yang cukup kuat, serta kemampuan Wi-Fi dan Bluetooth, meskipun dalam proyek ini fokus utamanya adalah pemrosesan data sensor dan kontrol motor. ESP32 bertugas membaca data dari sensor ultrasonik, menentukan apakah ada objek di depan robot, dan memberikan perintah yang sesuai ke motor driver untuk menggerakkan atau menghentikan motor. ESP32 juga bertanggung jawab untuk mengontrol LED sebagai indikator visual.



Gambar 3.2 ESP32

Sumber:

https://grobotronics.com/esp32-development-board-devkit-v1.html?srsltid=AfmBOoqC_7it3vx9CoWj97U8FtCHpSI7v2aWmyfb6zEkKlOpFfRGaZb4

3. Motor DC / Dinamo

Motor DC atau dinamo adalah komponen yang mengubah energi listrik menjadi energi gerak. Dalam proyek ini, dua motor DC digunakan untuk menggerakkan roda robot. Setiap motor DC akan memutar satu roda, dan dengan mengontrol kecepatan dan arah putaran kedua motor, kita dapat mengendalikan pergerakan robot, baik maju, mundur, maupun berbelok.



Gambar 3.3 Motor DC

Sumber: <https://www.arduinoindonesia.id/2018/08/motor-dc-dinamo-dc.html>

4. Motor Driver L29N

Motor driver L29N adalah modul yang digunakan untuk mengendalikan motor DC. Mikrokontroler seperti ESP32 tidak dapat memberikan daya yang cukup untuk menggerakkan motor secara langsung. Motor driver L29N berfungsi sebagai jembatan antara mikrokontroler dan motor, memungkinkan mikrokontroler untuk mengontrol arah dan kecepatan motor dengan sinyal kontrol yang kecil. Modul ini memiliki transistor yang dapat mengalirkan arus yang lebih besar ke motor, sesuai dengan sinyal kontrol dari mikrokontroler.



Gambar 3.4 Motor Driver L29N

Sumber:

<https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/>

5. DC Step-Up Boost Converter

Step-up converter adalah modul catu daya yang berfungsi untuk meningkatkan tegangan dari sumber daya yang lebih rendah ke tegangan yang lebih tinggi. Dalam proyek ini, step-up converter digunakan untuk menaikkan tegangan dari baterai ke level yang dibutuhkan oleh motor agar dapat bekerja dengan optimal.



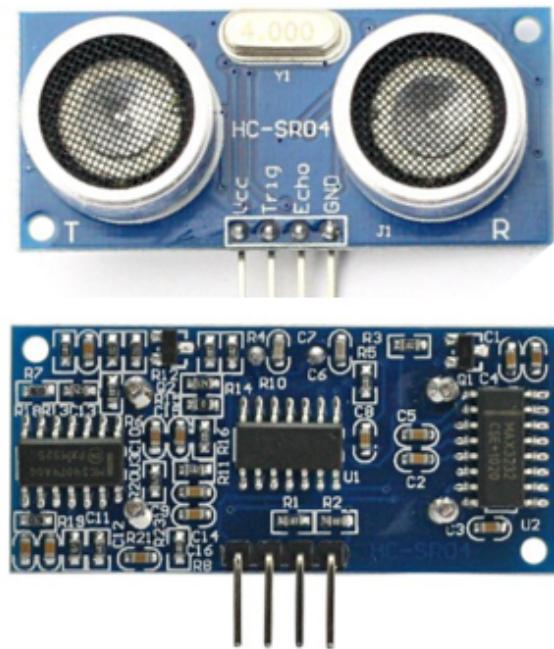
Gambar 3.5 Step-Up Boost Converter

Sumber:

<https://www.bodegaaurrera.com.mx/ip/electricidad/150w-dc-dc-boost-converter-step-up-power-supply-module-10-32v-to-12-35v-board/00738757575456>

6. Sensor Ultrasonik HC-SR04

Sensor ultrasonik HC-SR04 adalah sensor yang digunakan untuk mengukur jarak suatu objek. Sensor ini bekerja dengan mengirimkan gelombang ultrasonik, dan kemudian mengukur waktu yang dibutuhkan gelombang tersebut untuk kembali setelah memantul dari objek. Berdasarkan waktu pantulan ini, sensor dapat menghitung jarak objek. Dalam proyek ini, sensor ultrasonik HC-SR04 digunakan untuk mendeteksi keberadaan dan jarak objek di depan robot, yang merupakan bagian penting dari fitur Automatic Emergency Braking (AEB).



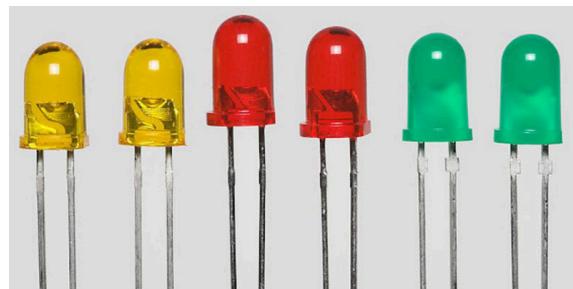
Sumber 3.6 Sensor HC-SR04

Sumber:

<https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/>

7. LED

LED atau Light Emitting Diode adalah komponen yang mengeluarkan cahaya ketika arus listrik melewatkinya. Dalam proyek ini, LED digunakan sebagai indikator visual. Misalnya, satu LED mungkin digunakan untuk menandakan bahwa robot dalam mode penggereman otomatis, dan LED lainnya mungkin digunakan untuk menandakan mode penggereman manual. LED memberikan umpan balik visual yang berguna bagi pengguna untuk memahami status robot.



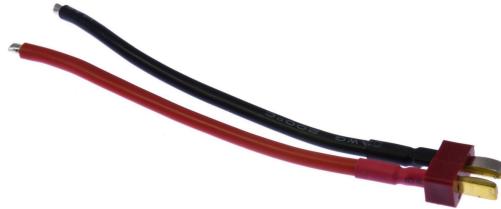
Gambar 3.7 LED

Sumber: <https://www.energysavinglighting.org/the-history-of-led-lighting/>

8. Batrei T Plug Dean

Baterai T Plug Dean adalah sumber daya listrik yang digunakan untuk menyediakan energi bagi seluruh sistem robot. Baterai ini memiliki konektor T Plug Dean, yang merupakan jenis konektor yang umum digunakan dalam hobi model remote control dan elektronik lainnya. Baterai ini menyediakan tegangan dan arus

yang cukup untuk menggerakkan motor dan memberi daya pada semua komponen elektronik lainnya.

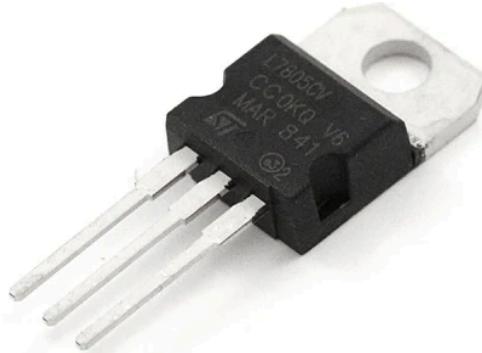


Gambar 3.8 Batrai T Plug Dean

Sumber: <https://www.solarbotics.com/product/14223>

9. Regulator 5V

Regulator 5V adalah komponen yang digunakan untuk menurunkan dan menstabilkan tegangan dari sumber daya, seperti baterai, ke level 5V. Tegangan ini dibutuhkan oleh ESP32 dan berbagai komponen elektronik lainnya agar dapat beroperasi dengan baik. Saat hanya menggunakan satu regulator, tegangan yang dihasilkan tidak stabil sehingga menyebabkan ESP32 sering mengalami restart. Oleh karena itu, kami memutuskan untuk menggunakan dua regulator guna memastikan kestabilan suplai tegangan 5V.

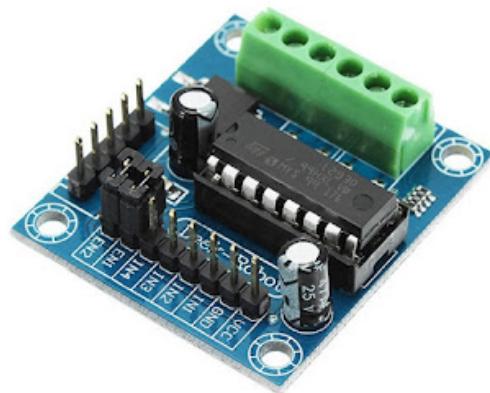


Gambar 3.9 Regulator 5V

Sumber: <https://www.sparkfun.com/voltage-regulator-5v.html>

10. Mosfet L293D

Mosfet L293D adalah driver motor lain yang juga dapat digunakan untuk mengendalikan motor DC. Mirip dengan L29N, L293D memungkinkan mikrokontroler untuk mengontrol arah dan kecepatan motor dengan sinyal kontrol yang kecil. Mosfet L293D memiliki keunggulan dalam efisiensi daya dan kemampuan menangani arus yang lebih tinggi dibandingkan dengan beberapa driver motor lainnya.



Gambar 3.10 Mosfet L29D

Sumber:

<https://www.andalanelektronik.id/2020/06/skema-rangkaian-ic-l293d-untuk-driver-motor-dc.htm>

11. Switch

Switch adalah saklar yang digunakan untuk menghidupkan dan mematikan daya pada robot. Switch ini berfungsi sebagai saklar utama, memungkinkan pengguna untuk dengan mudah mengaktifkan atau menonaktifkan seluruh sistem robot.



Gambar 3.11 Switch

Sumber: <https://www.chinadaier.com/kcd1-102-3pin-spdt-on-on-rocker-switch/>

12. Push Button

Push button adalah tombol tekan yang digunakan sebagai input manual. Dalam proyek ini, push button dapat digunakan untuk mengaktifkan mode penggereman manual. Ketika tombol ditekan, ESP32 akan menerima sinyal input dan menjalankan fungsi penggereman manual, menghentikan motor dan mengaktifkan LED indikator.



Gambar 3.12 Push Button

Sumber: <https://components101.com-switches/push-button>

13. Roda

Roda adalah bagian yang dipasang pada motor DC untuk memungkinkan robot bergerak. Dengan mengontrol putaran motor, kita dapat menggerakkan roda, dan dengan demikian menggerakkan robot ke arah yang diinginkan.



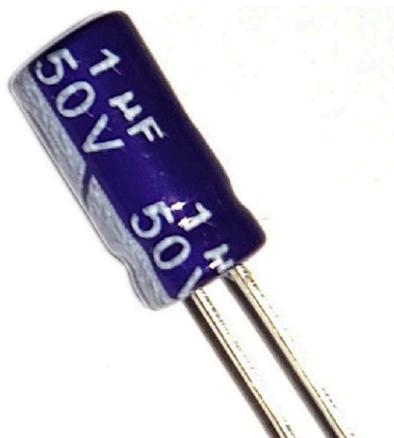
Gambar 3.13 Roda

Sumber:

<https://shopee.co.id/Roda-Robot-Smart-Car-67mm-Blue-RC-Whell-Velg-Ban-Karet-Radius-i.292092948.22702306335>

14. Capacitor 5V 1uF

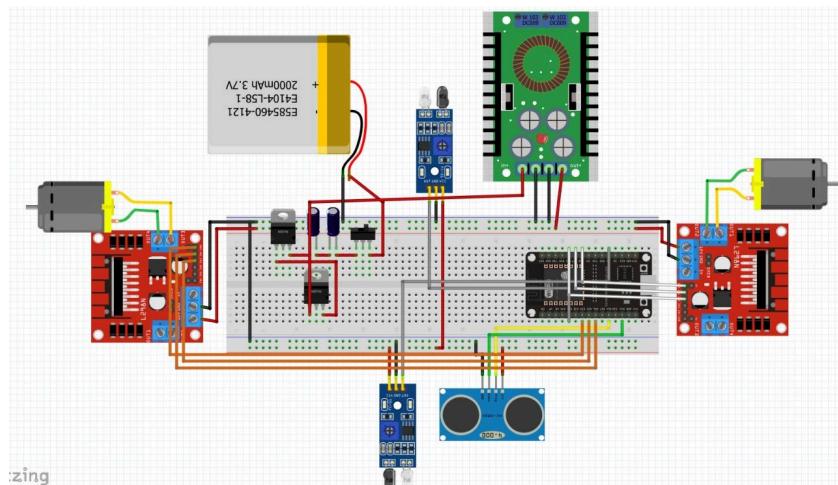
Kapasitor 5V 1 μ F berfungsi sebagai penyimpan dan penstabil tegangan pada rangkaian elektronik yang bekerja pada tegangan 5 volt. Kapasitor ini membantu mengurangi fluktuasi atau noise pada suplai daya, sehingga perangkat elektronik dapat beroperasi dengan lebih stabil. Selain itu, kapasitor 1 μ F juga berguna untuk menyuplai arus sesaat saat terjadi lonjakan beban ringan, serta membantu menjaga kestabilan tegangan pada gangguan atau penurunan tegangan jangka pendek.



Gambar 3.14 Kapasitor
Sumber:

<https://www.flipkart.com/wizzo-pack-100-pieces-1uf-5v-electrolytic-capacitors-electronic-component-components-hobby-kit/p/itm4da2781c04546>

3.2 Sketsa Rangkaian



Gambar 3.15 Sketsa Rangkaian

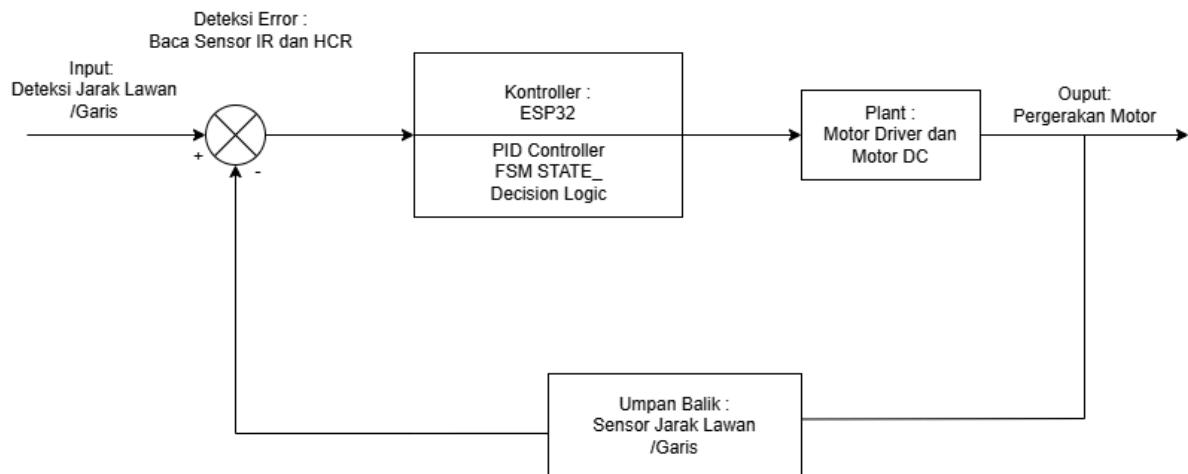
Rangkaian yang digunakan pada proyek ini merupakan sistem robot mobil yang mampu melakukan pengereman secara otomatis maupun manual. Sumber daya utama berasal dari baterai Li-Po 3,7V 2000mAh, yang kemudian dinaikkan tegangannya menjadi 5V menggunakan modul step-up converter MT3608. Tegangan ini kemudian distabilkan dengan dua buah kapasitor elektrolit yang dipasang pada

jalur output untuk menghindari lonjakan arus dan menjaga kestabilan suplai ke seluruh komponen. Mikrokontroler ESP32 berfungsi sebagai pusat kendali sistem, menerima input dari sensor ultrasonik HC-SR04 dan modul penerima infrared (IR). Sensor HC-SR04 digunakan untuk mendeteksi jarak antara mobil dan objek di depannya. Ketika jarak objek lebih dekat dari ambang batas yang telah ditentukan (misalnya 30 cm), ESP32 akan memproses data tersebut dan mengirimkan sinyal ke transistor NPN yang akan memutus arus ke motor, sehingga mobil berhenti secara otomatis.

Di sisi lain, sistem juga mendukung penggereman manual melalui remote inframerah. Ketika objek terdeteksi, ESP32 hanya menyalakan LED sebagai peringatan. Mobil baru akan berhenti jika pengguna menekan tombol pada remote, dan sinyal tersebut diterima oleh modul IR lalu diproses oleh ESP32 untuk menghentikan motor. Dua motor DC sebagai penggerak utama dikendalikan melalui dua buah driver motor L298N yang menerima sinyal logika dan PWM dari ESP32. Selain itu, terdapat dua sensor infrared tambahan pada rangkaian, yang dapat digunakan untuk fitur opsional seperti pendekripsi garis atau rintangan tambahan. Secara keseluruhan, rangkaian ini mendemonstrasikan sistem penggereman cerdas berbasis mikrokontroler dengan dua mode kerja: otomatis dan manual, yang responsif terhadap kondisi lingkungan sekitar.

3.2 Block Diagram

Closed Loop



Gambar 3.17 Diagram Closed Loop

Dalam sistem kendali closed-loop, tindakan kontrol bergantung pada output dari sistem. Sistem terus-menerus mengukur apa yang sebenarnya terjadi, membandingkannya dengan apa yang ingin terjadi, dan kemudian menyesuaikan tindakannya.

1. Input — Jarak Objek

- Input adalah jarak objek yang ingin dipertahankan atau diukur oleh sistem.

2. Deteksi Error — Baca Sensor Jarak

- Sensor jarak digunakan untuk membaca jarak aktual objek secara real-time.
- Nilai ini dibandingkan dengan nilai input (setpoint) untuk menghitung error.

3. PID Controller — Arduino (PID Controller)

- ESP32 digunakan untuk mengimplementasikan kontrol PID.
PID (Proportional-Integral-Derivative) menghitung nilai koreksi untuk mengurangi error berdasarkan:
 - **P** (Proportional): mengoreksi kesalahan saat ini.
 - **I** (Integral): mengoreksi kesalahan kumulatif masa lalu.
 - **D** (Derivative): mengoreksi prediksi kesalahan masa depan.
- Blok ini menerima error dari deteksi error dan menghasilkan sinyal kontrol.

4. Plant — Motor DC

- Motor DC menerima sinyal kontrol dari Arduino dan menggerakkan aktuator untuk memperbaiki jarak objek sesuai keinginan.

5. Output — Pergerakan Motor

- Output sistem adalah pergerakan motor yang dihasilkan sesuai dengan sinyal kontrol dari PID Controller.

6. Umpang Balik — Sensor Jarak

- Sensor jarak mengukur jarak aktual yang dicapai oleh motor.
- Data ini dikirim kembali ke sistem deteksi error sebagai umpan balik untuk memperbarui error secara terus-menerus (loop tertutup).

3.3 Source Code

```
1      const int IN1 = 26;
2      const int IN2 = 25;
3      const int ENA = 33;
4
5      const int IN3 = 18;
6      const int IN4 = 21;
7      const int ENB = 19;
8
9      const int trigPin = 14;
10     const int echoPin = 13;
11
12     const int IR_FRONT = 4;
13     const int IR_sensor_back = 35;
14
15     long duration;
16     float distance;
17
18     const int IR_THRESHOLD_BLACK_DETECT = 3000;
19
20     enum RobotState {
21         STATE_IDLE,
22         STATE_FORWARD,
23         STATE_AVOID_EDGE_BACK,
24         STATE_AVOID_EDGE_FRONT,
25         STATE_ATTACK
```

```

26    };
27
28    RobotState currentState = STATE_IDLE;
29
30 #define LEDC_TIMER_BIT      8
31 #define LEDC_BASE_FREQ      5000
32
33 const int DEFAULT_MOTOR_SPEED = 180;
34 const int DEFEND_MOTOR_SPEED = 255;
35
36 void setup() {
37     Serial.begin(115200);
38     Serial.println("--- Sumobot ESP32 Siap Tempur! ---");
39
40     pinMode(IN1, OUTPUT);
41     pinMode(IN2, OUTPUT);
42     pinMode(IN3, OUTPUT);
43     pinMode(IN4, OUTPUT);
44
45     pinMode(trigPin, OUTPUT);
46     pinMode(echoPin, INPUT);
47
48     pinMode(IR_FRONT, INPUT);
49     pinMode(IR_sensor_back, INPUT);
50
51     ledcAttach(ENA, LEDC_BASE_FREQ, LEDC_TIMER_BIT);
52     ledcAttach(ENB, LEDC_BASE_FREQ, LEDC_TIMER_BIT);
53
54     Serial.println("Konfigurasi Sensor dan Motor Selesai.");
55     Serial.print("IR Black Detect Threshold (Garis): ");
56     Serial.println(IR_THRESHOLD_BLACK_DETECT);
57     Serial.print("Kecepatan Motor Normal: ");
58     Serial.println(DEFAULT_MOTOR_SPEED);
59     Serial.print("Kecepatan Motor Bertahan: ");
60     Serial.println(DEFEND_MOTOR_SPEED);
61
62     Serial.println("Robot akan mulai bergerak dalam 2 detik...");
63     delay(2000);
64     currentState = STATE_FORWARD;
65     Serial.println("Robot memulai dalam keadaan: FORWARD");
66 }
67
68 void loop() {
69     int IR_back = analogRead(IR_sensor_back);
70     int IR_front = analogRead(IR_FRONT);
71     distance = getDistance();
72
73     if (IR_back >= IR_THRESHOLD_BLACK_DETECT) {
74         if (currentState != STATE_AVOID_EDGE_BACK) {
75             currentState = STATE_AVOID_EDGE_BACK;
76             Serial.println("Transisi ke keadaan: AVOID_EDGE_BACK");
77             (Garis Hitam Belakang Terdeteksi");
78         }
79     }
80
81     else if (IR_front >= IR_THRESHOLD_BLACK_DETECT) {
82         if (currentState != STATE_AVOID_EDGE_FRONT) {
83             currentState = STATE_AVOID_EDGE_FRONT;
84             Serial.println("Transisi ke keadaan: AVOID_EDGE_FRONT");
85             (Garis Hitam Depan Terdeteksi");

```

```

86         }
87     }
88     else if (distance > 0 && distance < 20) {
89         if (currentState != STATE_ATTACK) {
90             currentState = STATE_ATTACK;
91             Serial.println("Transisi ke keadaan: ATTACK (Lawan
92 Terdeteksi oleh Ultrasonik)");
93         }
94     }
95     else {
96         if (currentState != STATE_FORWARD) {
97             currentState = STATE_FORWARD;
98             Serial.println("Transisi ke keadaan: FORWARD (Area
99 Aman, Maju)");
100        }
101    }

102    switch (currentState) {
103        case STATE_IDLE:
104            motorStop();
105            break;

106        case STATE_FORWARD:
107            Serial.println("Aksi: MAJU TERUS! (Kecepatan
108 Normal)");
109            FORWARD(DEFAULT_MOTOR_SPEED);
110            break;

111        case STATE_AVOID_EDGE_BACK:
112            Serial.println("Aksi: AVOID_EDGE_BACK (Maju Kencang
113 Mempertahankan Batas)");
114            FORWARD(DEFEND_MOTOR_SPEED);
115            break;

116        case STATE_AVOID_EDGE_FRONT:
117            Serial.println("Aksi: AVOID_EDGE_FRONT (Berhenti &
118 Mundur dari Garis Depan)");
119            motorStop();
120            delay(100);
121            BACKWARD(DEFAULT_MOTOR_SPEED);
122            delay(300);
123            motorStop();
124            delay(50);
125            ROTATE_RIGHT(DEFAULT_MOTOR_SPEED);
126            delay(200);
127            motorStop();
128            delay(50);
129            break;

130        case STATE_ATTACK:
131            Serial.println("Aksi: MAJU MENYERANG! (Kecepatan
132 Normal)");
133            FORWARD(DEFAULT_MOTOR_SPEED);
134            break;
135    }

136    Serial.print("IR Depan: "); Serial.print(IR_front);
137    Serial.print(" | IR Belakang: "); Serial.print(IR_back);
138    Serial.print(" | Jarak Ultrasonik: ");
139    Serial.print(distance); Serial.println(" cm");

```

```

156     Serial.print("State Saat Ini: ");
157     switch (currentState) {
158         case STATE_IDLE: Serial.println("IDLE"); break;
159         case STATE_FORWARD: Serial.println("FORWARD"); break;
160         case STATE_AVOID_EDGE_BACK: break;
161     Serial.println("AVOID_EDGE_BACK"); break;
162     case STATE_AVOID_EDGE_FRONT: break;
163     Serial.println("AVOID_EDGE_FRONT"); break;
164         case STATE_ATTACK: Serial.println("ATTACK"); break;
165     }
166     Serial.println("---");
167
168     delay(50);
169 }
170
171     float getDistance() {
172         digitalWrite(trigPin, LOW);
173         delayMicroseconds(2);
174         digitalWrite(trigPin, HIGH);
175         delayMicroseconds(10);
176         digitalWrite(trigPin, LOW);
177         duration = pulseIn(echoPin, HIGH, 30000);
178         if (duration == 0) {
179             return -1;
180         } else {
181             float jarak = (duration * 0.0343) / 2;
182             return jarak;
183         }
184     }
185
186     void motorForward(int speed) {
187         digitalWrite(IN1, HIGH); digitalWrite(IN2, LOW);
188         digitalWrite(IN3, HIGH); digitalWrite(IN4, LOW);
189         ledcWrite(ENA, speed);
190         ledcWrite(ENB, speed);
191     }
192
193     void motorBackward(int speed) {
194         digitalWrite(IN1, LOW); digitalWrite(IN2, HIGH);
195         digitalWrite(IN3, LOW); digitalWrite(IN4, HIGH);
196         ledcWrite(ENA, speed);
197         ledcWrite(ENB, speed);
198     }
199
200     void motorLeft(int speed) {
201         digitalWrite(IN1, LOW); digitalWrite(IN2, HIGH);
202         digitalWrite(IN3, HIGH); digitalWrite(IN4, LOW);
203         ledcWrite(ENA, speed);
204         ledcWrite(ENB, speed);
205     }
206
207     void motorRight(int speed) {
208         digitalWrite(IN1, HIGH); digitalWrite(IN2, LOW);
209         digitalWrite(IN3, LOW); digitalWrite(IN4, HIGH);
210         ledcWrite(ENA, speed);
211         ledcWrite(ENB, speed);
212     }
213
214     void motorStop() {
215         digitalWrite(IN1, LOW); digitalWrite(IN2, LOW);

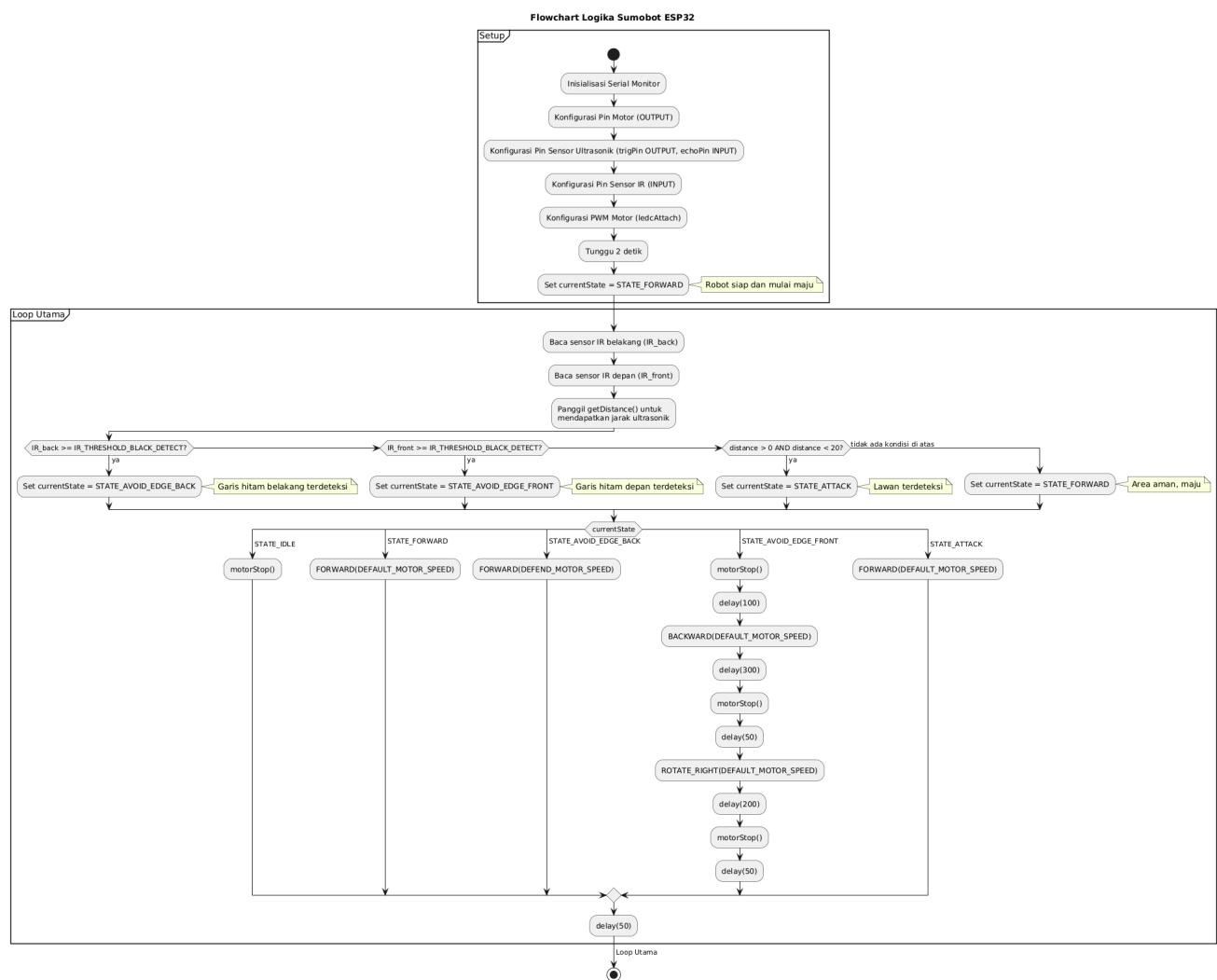
```

```

216     digitalWrite(IN3, LOW); digitalWrite(IN4, LOW);
217     ledcWrite(ENA, 0);
218     ledcWrite(ENB, 0);
219 }
220
221 void ROTATE_RIGHT(int speed) {
222     motorRight(speed);
223 }
224
225 void ROTATE_LEFT(int speed) {
226     motorLeft(speed);
227 }
228
229 void FORWARD(int speed) {
230     motorForward(speed);
231 }
232
233 void BACKWARD(int speed) {
234     motorBackward(speed);
235 }

```

3.4 Flowchart



BAB 4

KESIMPULAN

4.1 Pembagian Kinerja Kelompok

No	Nama/Nim	Kinerja
1	Ruasa Azizan Zihni	Desain & implementasi perangkat keras, Pengujian komponen
2	Barru Wira Yasa	Pemrograman mikrokontroler, Debugging & optimasi kode
3	Ibadurahman Faiz Usman	Integrasi sistem, Pembuatan robot
4	Rayhan Sulistyawan	Pengujian keseluruhan sistem, Analisis data & hasil
5	Dzaki Rabbani	Dokumentasi proyek dan demo
6	M. Radhi Rasyidi R	Dokumentasi proyek, Persiapan materi presentasi dan laporan

4.2 Kesimpulan

Program Sumobot ini berhasil mengintegrasikan ESP32 dengan berbagai sensor dan aktuator untuk menciptakan robot sumo otonom. Menggunakan Finite State Machine (FSM), robot dapat beradaptasi dengan lingkungan. Sensor ultrasonik HC-SR04 mendeteksi lawan, memicu STATE_ATTACK untuk maju agresif. Sementara itu, dua sensor inframerah (IR) di depan dan belakang berfungsi sebagai detektor garis arena, dengan sensor belakang memicu STATE_AVOID_EDGE_BACK untuk maju kuat mempertahankan posisi, dan sensor depan memicu STATE_AVOID_EDGE_FRONT untuk berhenti, mundur, dan berputar, mencegah robot keluar arena. Pergerakan motor DC dikontrol oleh driver L298N menggunakan PWM dari LEDC ESP32, memungkinkan pengaturan kecepatan yang halus (0-255).

Kecepatan motor yang berbeda didefinisikan untuk berbagai skenario, seperti kecepatan normal (DEFAULT_MOTOR_SPEED) dan kecepatan maksimal untuk bertahan (DEFEND_MOTOR_SPEED). Arsitektur FSM memberikan prioritas tinggi pada penghindaran garis (terutama belakang), diikuti oleh deteksi lawan, dan perilaku maju sebagai default. Serial monitor sangat membantu dalam debugging dan kalibrasi sistem ini. Meskipun fungsional, kalibrasi akurat nilai threshold IR (IR_THRESHOLD_BLACK_DETECT) sangat penting untuk deteksi garis yang optimal. Dengan kalibrasi yang tepat dan potensi penyempurnaan algoritma gerakan,

robot sumobot ini memiliki potensi besar untuk menjadi kompetitor yang tangguh di arena.

4.3 Daftar Pustaka

HC-SR04 Ultrasonic Sensor Datasheet. (2017). *Accuracy and Usage Guide*. Manufacturer.

Horswill, I., McKenna, F., & Thomson, J. (2008). Visual Reaction Times in Driving Simulations. *Journal of Applied Psychology*, 93(4), 971–984.

Texas Instruments (January 2016). L293x Quadruple Half-H Drivers. Retrieved July 8, 2020.

ElecFreaks (n.d.). Ultrasonic Ranging Module HC-SR04. Retrieved October 23, 2023.

Everlight Electronics Co., Ltd. (2007, April 4). Technical Data Sheet Opto Interrupter. Retrieved October 2nd, 2019.

REFLEKSI AKHIR MATA KULIAH SISTEM AUTOMASI

No	Nama/Nim	Refleksi Akhir Matkul	Foto Mahasiswa
1	Ruasa Azizan Zihni	<p>Saya ingin mengucapkan terima kasih banyak kepada Bapak Sabriansyah atas kesabaran dan semua ilmu yang telah diberikan selama satu semester ini di mata kuliah Sistem Automasi. Saya sangat senang dan bersyukur karena berkat bimbingan Bapak, saya yang tadinya tidak tahu apa-apa tentang Sistem Automasi kini menjadi lebih paham, bahkan bisa mengerti cara kerja PID Tuning dan motor DC. Semoga ilmu yang Bapak berikan menjadi amal jariyah dan sangat bermanfaat bagi saya dan teman-teman.</p>	
2	Barru Wira Yasa	<p>Terima kasih banyak saya sampaikan kepada Bapak Sabriansyah atas kesabaran dan semua ilmu yang diberikan dalam mata kuliah Sistem Automasi semester ini. Saya sungguh senang dan bersyukur, berkat bimbingan Bapak, pemahaman saya tentang Sistem Automasi meningkat pesat, dari yang awalnya tidak tahu apa-apa hingga kini mengerti cara kerja PID Tuning dan motor DC. Semoga ilmu ini menjadi amal jariyah bagi Bapak dan sangat bermanfaat untuk saya dan teman-teman</p>	

3	Ibadurahman Faiz Usman	<p>Pak Sabriansyah, terima kasih banyak atas kesabaran dan semua ilmu yang sudah Bapak berikan di kelas Sistem Automasi semester ini. Saya senang dan bersyukur banget, karena bimbingan Bapak, saya yang tadinya blank soal Sistem Automasi sekarang jadi jauh lebih paham, bahkan ngerti soal PID Tuning dan motor DC. Semoga ilmu dari Bapak jadi amal jariyah dan bermanfaat banget buat saya dan teman-teman.</p>	
4	Rayhan Sulistyawan	<p>Setelah mengikuti mata kuliah Sistem Otomasi dari awal pertemuan hingga pengumpulan projek akhir, saya belajar banyak mengenai sistem kontrol secara otomatis yang diajarkan Pak Sabriansyah. Pembelajaran dimulai dari banyak perhitungan matematis sampai implementasi kode program memberikan saya wawasan baru mengenai mata kuliah ini. Saya sangat senang sekali dapat belajar hal baru seperti PID, PLC, kontrol motor DC, Flux Weakening, dan lain sebagainya yang dapat menjadi bekal saya untuk melanjutkan studi ke tingkat akhir nantinya. Semoga Bapak diberikan kemudahan dalam segala hal dan menjadi amal jariyah karena sudah mengajarkan ilmu yang bermanfaat.</p>	

5	Dzaki Rabbani	<p>Selama mengikuti mata kuliah Sistem Automasi, saya mendapatkan pemahaman yang lebih dalam mengenai bagaimana sistem kendali otomatis bekerja secara teori maupun praktik. Materi seperti PID control, motor DC, hingga penerapan Flux Weakening tidak hanya diajarkan secara konseptual, tetapi juga langsung diterapkan dalam bentuk proyek nyata, yang sangat membantu saya memahami hubungan antara teori dan implementasi. Meskipun beberapa konsep seperti perhitungan kontrol dan tuning PID cukup kompleks, pendekatan pembelajaran yang diberikan oleh Pak Sabriansyah sangat sistematis dan mudah dipahami.</p>	
6	M. Radhi Rasyidi Rafli	<p>Selama saya mengikuti mata kuliah Sistem Automasi di semester ini, saya merasa pembelajaran yang diberikan sangat menyenangkan meskipun dihadapkan dengan berbagai tantangan, terutama dalam memahami dan menghafal berbagai rumus yang cukup kompleks. Melalui mata kuliah ini, saya menjadi lebih memahami cara kerja motor DC serta menyadari bahwa banyak perangkat di sekitar kita yang menggunakan motor DC sebagai bagian dari sistem kerjanya.</p>	