

**PENGEMBANGAN SISTEM *INTERNET OF THINGS* (IoT)
UNTUK MONITORING DAN KONTROL *TRAY OTOMATIS*
PADA ROBOT PENGANTAR MAKANAN**

**(DEVELOPMENT OF AN INTERNET OF THINGS (IoT) SYSTEM FOR
MONITORING AND CONTROLLING THE AUTOMATIC TRAY ON FOOD
DELIVERY ROBOTS)**

BUKU TUGAS AKHIR

Disusun sebagai syarat mata kuliah Tugas Akhir

Program Studi S1 Teknik Elektro

Disusun oleh:

GIOVANNI SALIM

1102213024



**Universitas
Telkom**

FAKULTAS TEKNIK ELEKTRO

UNIVERSITAS TELKOM

BANDUNG

2025

LEMBAR PENGESAHAN

TUGAS AKHIR

PENGEMBANGAN SISTEM *INTERNET OF THINGS (IoT)* UNTUK MONITORING DAN KONTROL TRAY OTOMATIS PADA ROBOT PENGANTAR MAKANAN

**(DEVELOPMENT OF AN *INTERNET OF THINGS (IoT)* SYSTEM FOR
MONITORING AND CONTROLLING THE AUTOMATIC TRAY ON
FOOD DELIVERY ROBOTS)**

Telah disetujui dan disahkan untuk mengikuti sidang Tugas Akhir

Program Studi Teknik Elektro

Fakultas Teknik Elektro

Universitas Telkom

Disusun oleh:

GIOVANNI SALIM

1102213024

Bandung, tanggal Juli 2025

Pembimbing I



Dr. ERWIN SUSANTO, S.T., M.T.

NIP.07740045

Pembimbing II



IRHAM MULKAN RODIANA, S.T., M.T.

NIP. 20950004

Dosen Pengusul Topik



Dr. ERWIN SUSANTO, S.T., M.T.

NIP.07740045

ABSTRAK

Pada abad ke-21, industri Food and Beverage (F&B) mengalami perkembangan pesat dengan memanfaatkan teknologi, termasuk robotika, untuk meningkatkan efisiensi layanan. Salah satu inovasi terkini adalah robot pengantar makanan, yang berpotensi menggantikan tenaga manusia dalam pengantaran makanan untuk meningkatkan keamanan, kebersihan, dan konsistensi layanan. Namun, tantangan teknis dalam desain dan implementasi robot ini, seperti *tray* statis dan perlindungan makanan yang kurang, menghambat kinerjanya. Penelitian ini bertujuan untuk mengembangkan sistem mekatronik yang dapat meningkatkan efektivitas robot pengantar makanan, termasuk *tray* otomatis yang steril dan sistem navigasi yang akurat.

Metode penelitian melibatkan studi literatur, permodelan, simulasi menggunakan software desain, pengujian sensor dan aktuator, serta analisis hasil berdasarkan pengukuran sudut dan posisi robot. Sistem yang dirancang mencakup mikrokontroler Arduino Mega untuk kontrol *tray* dan Raspberry Pi untuk pergerakan robot, serta teknologi Internet of Things (IoT) untuk monitoring melalui perangkat lunak Kodular.

Hasil pengujian menunjukkan bahwa konfigurasi PID menghasilkan pergerakan *tray* yang stabil dan presisi ditandai dengan rata-rata *error* hanya 2,66% pada jarak 27,5 cm dan 3,21% pada jarak 8,5 cm. Hal ini membuktikan bahwa sistem memiliki tingkat akurasi tinggi, pergerakan stabil, serta minim osilasi dan overshoot. Adanya *tray* otomatis yang termonitoring dengan sistem IoT membuat robot pengantar makanan menjadi lebih inovatif serta memberikan kemudahan dan kenyamanan lebih bagi pengguna.

Kata Kunci: *efisiensi layanan, IoT, robot pengantar makanan, tray otomatis, sistem mekatronik.*

ABSTRACT

In the 21st century, the Food and Beverage (F&B) industry has experienced rapid development by utilizing technology, including robotics, to improve service efficiency. One of the latest innovations is the food delivery robot, which has the potential to replace human labor in delivering food to enhance safety, hygiene, and service consistency. However, technical challenges in the design and implementation of these robots, such as static *trays* and inadequate food protection, hinder their performance. This study aims to develop a mechatronic system that can improve the effectiveness of food delivery robots, including an automatic sterile *tray* and an accurate navigation system.

The research methodology involved literature review, modeling, simulation using design software, testing of sensors and actuators, and analysis of results based on the robot's angle and position measurements. The designed system integrates an Arduino Mega microcontroller for *tray* control and a Raspberry Pi for robot movement, along with Internet of Things (IoT) technology for monitoring through the Kodular software.

The test results show that the PID configuration produces stable and precise tray movements, as indicated by an average error of only 2,66% at a distance of 27,5 cm and 3,21% at a distance of 8,5 cm. This proves that the system has a high level of accuracy, stable movement, and minimal oscillation and overshoot. The presence of an automatically monitored tray integrated with an IoT system makes the food delivery robot more innovative, providing greater convenience and comfort for users.

Keywords: *service efficiency, IoT, food delivery robot, automatic tray, mechatronic system.*

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah SWT atas segala rahmat, karunia, dan kemudahan yang telah diberikan sehingga penulis dapat menyelesaikan penyusunan tugas akhir yang berjudul "**Pengembangan Sistem Internet of Things (IoT) untuk *Monitoring* dan Kontrol *Tray* Otomatis pada Robot Pengantar Makanan**" dengan lancar.

Tugas akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik Elektro, Telkom University. Dalam proses penyusunan Tugas Akhir ini, penulis menyadari bahwa tidak mungkin dapat menyelesaiakannya tanpa adanya dukungan, bantuan, dan doa dari berbagai pihak.

Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu, baik secara langsung maupun tidak langsung, dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun demi perbaikan di masa yang akan datang. Semoga Tugas Akhir ini dapat memberikan manfaat, baik bagi penulis sendiri, maupun bagi pembaca dan pihak-pihak yang berkepentingan.

UCAPAN TERIMA KASIH

Dalam proses penyusunan Tugas Akhir ini, penulis menerima berbagai bentuk bantuan yang sangat berarti, baik dukungan moral, bimbingan akademik, maupun saran yang membangun dari berbagai pihak. Oleh karena itu dengan penuh rasa hormat dan tulus, penulis ingin menyampaikan ucapan terima kasih kepada pihak-pihak yang telah berkontribusi dalam proses penyusunan Tugas Akhir ini:

1. Allah SWT, atas segala nikmat, kekuatan, dan kesehatan yang telah diberikan.
2. Orang tua tercinta, atas doa, dukungan moral, dan semangat yang diberikan serta motivasi yang tiada henti-hentinya selalu diberikan kepada penulis selama melakukan masa studi hingga penyusunan Tugas Akhir ini.
3. Bapak Dr. Erwin Susanto, S.T., M.T. dan Bapak Irham Mulkan Rodiana, S.T., M.T. selaku Dosen pembimbing pertama dan kedua, yang telah dengan sabar membimbing, mengarahkan, dan memberikan masukan dalam setiap tahap penyusunan Tugas Akhir ini.
4. Teman-teman tim T.I.F.A yang telah banyak membantu, mendukung, dan menemani penulis selama proses penggeraan Tugas Akhir ini.
5. Teman-teman Keluarga Besar Kang Gabut, atas canda tawa, kebersamaan, dan dukungan emosional yang sangat membantu dalam menjaga semangat penulis.
6. Teman-teman Kost Eyang, atas kebersamaan yang menyenangkan dan semangat belajar bersama yang sangat dikenang.
7. Teman-teman ELITIA Laboratory, atas bantuan teknis dan kerja sama selama menempuh studi di lingkungan laboratorium.
8. Teman-teman kelas EL-45-04, atas kebersamaan, kerja sama, dan semangat belajar yang senantiasa membantu penulis untuk terus maju.
9. Nidya Intani Dwi Saka Putri, atas dukungan dan teman projek dalam mengerjakan Tugas Akhir.
10. Seluruh dosen Program Studi S1 Teknik Elektro Telkom University, atas ilmu, wawasan, dan dedikasi dalam proses pembelajaran selama ini.

11. Dan semua pihak yang tidak dapat disebutkan satu per satu, yang telah memberikan bantuan maupun dukungan selama proses penyusunan Tugas Akhir ini.

Tanpa kehadiran, bimbingan, serta dukungan dari semua pihak di atas, tentu Tugas Akhir ini tidak akan dapat terselesaikan dengan baik. Semoga segala bantuan dan kebaikan yang telah diberikan menjadi amal yang dibalas dengan kebaikan yang berlipat ganda.

DAFTAR ISI

LEMBAR PENGESAHAN	2
ABSTRAK	3
ABSTRACT	4
KATA PENGANTAR.....	5
UCAPAN TERIMA KASIH	6
DAFTAR ISI.....	8
DAFTAR GAMBAR.....	11
DAFTAR TABEL	12
BAB I PENDAHULUAN.....	13
1.1 Latar Belakang Masalah.....	13
1.2. Rumusan Masalah	14
1.3. Tujuan	14
1.4. Manfaat Hasil Penelitian	15
1.5. Batasan Masalah	15
1.6. Metode Penelitian	16
1.7. Proyeksi Pengguna	17
BAB II TINJAUAN PUSTAKA.....	18
2.1 Robotika	18
2.1.1 BellaBot	19
2.2 <i>Internet Of Things</i>	19
2.2.1 Kodular	20
2.3 Mekanisme <i>Tray</i>	20
2.4 Sistem Kontrol	20
2.4.1 Sistem Kontrol PID	22
2.4.2 Respon Transien.....	23
BAB III PERANCANGAN SISTEM	25
3.1 Desain Sistem	25
3.1.1 Diagram Blok.....	25
3.1.2 Fungsi dan Fitur.....	26

3.2	Desain Perangkat Keras	27
3.3	Desain UI.....	28
3.4	Desain PCB	29
3.5	Desain Fisik Robot.....	30
 3.5.1	 Desain <i>Tray</i> Pada Robot	31
3.6	Desain Parameter PID	33
 3.6.1	 Parameter Desain $K_p=8.0, K_i=0.8, K_d=1.2$	34
 3.6.2	 Parameter Desain $K_p=2.0, K_i=0.5, K_d=1.0$	35
 3.6.3	 Parameter Desain $K_p=3.6, K_i=0.5, K_d=3.0$	36
3.7	Spesifikasi Komponen.....	38
3.8	Desain Kinerja Sistem.....	46
 3.8.1	 Diagram Alir Sistem <i>Tray</i>.....	46
BAB IV HASIL DAN ANALISIS		48
4.1.	Hasil Percobaan.....	48
 4.1.1	 Pengujian IR sensor FC-51	48
 4.1.2	 Pengujian Motor Driver BTS7960.....	49
 4.1.3	 Pengujian Motor DC JGY370.....	51
 4.1.4	 Pengujian <i>Limit Switch</i> KW10-Z1P.....	52
 4.1.5	 Pengujian <i>Ultrasonic</i> HC-SR04.....	52
 4.1.6	 Pengujian Sistem Kontrol Motor DC Menggunakan PID Berbasis Jarak <i>Ultrasonic</i> HC-SR04	54
 4.1.7	 Pengujian Sistem Jarak Jauh (<i>Internet Of Things</i>).....	57
4.2.	Pengujian Parameter PID.....	58
 4.2.1	 Pengujian Menggunakan Parameter $K_p=3.6, K_i=0, K_d=0$.....	58
 4.2.2	 Pengujian Menggunakan Parameter $K_p=3.6, K_i=0, K_d=3.0$....	61
 4.2.3	 Pengujian Menggunakan Parameter $K_p=3.6, K_i=0.5, K_d=3.0$.63	
4.3.	Analisis	66
 4.3.1	 Analisis Performa Sensor IR FC-51 Dalam Deteksi Makanan.	66
 4.3.2	 Analisis Fungsi <i>Limit Switch</i> KW10-Z1P sebagai Pengaman dan Pengontrol Posisi.....	67
 4.3.3	 Analisis Integrasi Sistem <i>Monitoring</i> dan Kontrol IoT	67
 4.3.4	 Analisis Penggunaan Kontrol PID Pada Pergerakan <i>Tray</i>	68
BAB V KESIMPULAN DAN SARAN		70

5.1	Kesimpulan	70
5.2	Saran.....	71
	DAFTAR PUSTAKA	72
	LAMPIRAN.....	74

DAFTAR GAMBAR

Gambar 2. 1 BellaBot	19
Gambar 2. 2 Internet of Things	19
Gambar 2. 3 Blok Diagram Loop Tertutup	21
Gambar 2. 4 Blok Diagram Loop Terbuka.....	21
Gambar 2. 5 Bentuk Sinyal Tanggapan Transien.....	23
Gambar 3. 1 Blok Diagram Sistem.....	25
Gambar 3. 2 Blok Diagram Perangkat Keras	27
Gambar 3. 3 Diagram Sistem Tray	28
Gambar 3. 4 Desain UI Screen 1	28
Gambar 3. 5 Desain UI Screen 2	29
Gambar 3. 6 Desain wiring PCB	30
Gambar 3. 7 Desain Prototipe robot tampak samping kanan	30
Gambar 3. 8 Desain Prototipe robot tampak depan.....	31
Gambar 3. 9 Tampak Atas Tray Aktif.....	31
Gambar 3. 10 Tampak Bawah Tray Aktif	32
Gambar 3. 11 Tampak Atas Tray Aktif dengan nampan.....	32
Gambar 3. 12 Tampak Depan Bracket sensor IR	32
Gambar 3. 13 Tampak Belakang Bracket sensor IR	33
Gambar 3. 14 Diagram blok pengontrol PID	33
Gambar 3. 15 Bentuk Transfer Function Desain 1	34
Gambar 3. 16 Bentuk grafik simulasi PID Desain 1	34
Gambar 3. 17 Bentuk Transfer Function Desain 2	35
Gambar 3. 18 Bentuk grafik simulasi PID Desain 2	36
Gambar 3. 19 Bentuk Transfer Function Desain 3	37
Gambar 3. 20 Bentuk grafik simulasi PID Desain 3	37
Gambar 3. 21 Arduino Mega 2560 Pro Mini.....	38
Gambar 3. 22 ESP-32 Dev-Kit	39
Gambar 3. 23 Step Down LM2596	40
Gambar 3. 24 IR Sensor FC-51	41
Gambar 3. 25 Limit Switch Microswitch KW10-Z1P	41
Gambar 3. 26 Ultrasonic HC-SR04.....	42
Gambar 3. 27 Motor DC JGY370 wormgear	43
Gambar 3. 28 Motor Driver BTS7960.....	44
Gambar 3. 29 Baterai Lithium Polymer 48V	45
Gambar 3. 30 Diagram alir sistem tray.....	46
Gambar 4. 1 Proses Pengujian sensor Ultrasonic HC-SR04	53
Gambar 4. 2 Hasil monitoring melalui database	58
Gambar 4. 3 Grafik Respon Transien PID Kp setpoint 27,5 cm	59
Gambar 4. 4 Grafik Respon Transien PID Kp setpoint 8,5 cm	60
Gambar 4. 5 Grafik Respon Transien PID Kp dan Kd setpoint 27,5 cm	62
Gambar 4. 6 Grafik Respon Transien PID Kp dan Kd setpoint 8,5 cm	63
Gambar 4. 7 Grafik Respon Transien PID Lengkap setpoint 27,5 cm.....	64
Gambar 4. 8 Grafik Respon Transien PID Lengkap setpoint 8,5 cm.....	65

DAFTAR TABEL

Tabel 3. 1 Spesifikasi Arduino Mega 2560 Pro Mini.....	38
Tabel 3. 2 Spesifikasi ESP-32 Dev-Kit	39
Tabel 3. 3 Spesifikasi Step Down LM2596	40
Tabel 3. 4 Spesifikasi IR Sensor FC-51	41
Tabel 3. 5 Spesifikasi Limit Switch Microswitch KW10-1P	42
Tabel 3. 6 Spesifikasi Ultrasonic HC-SR04.....	42
Tabel 3. 7 Spesifikasi Motor DC JGY370 wormgear	43
Tabel 3. 8 Spesifikasi Motor Driver BTS7960.....	44
Tabel 3. 9 Spesifikasi Baterai Lithium Polymer 48V	45
Tabel 4. 1 Pengujian IR sensor FC-51	48
Tabel 4. 2 Pengujian Motor Driver BTS7960	50
Tabel 4. 3 Pengujian motor DC JGY370.....	51
Tabel 4. 4 Pengujian Sensor Limit Switch KW10-Z1P	52
Tabel 4. 5 Pengujian Sensor Ultrasonic HC-SR04.....	53
Tabel 4. 6 Pengujian Sistem kontrol PID jarak 27.5 cm	55
Tabel 4. 7 Pengujian Sistem kontrol PID jarak 8.5 cm	56
Tabel 4. 8 Pengujian Delay sistem Inter Of Things	57
Tabel 4. 9 Pengujian Parameter Kp pada Setpoint 27.5 cm	59
Tabel 4. 10 Pengujian Parameter Kp pada Setpoint 8.5 cm	60
Tabel 4. 11 Pengujian Parameter Kp dan Kd pada Setpoint 27.5 cm	61
Tabel 4. 12 Pengujian Parameter Kp dan Kd pada Setpoint 8.5 cm	62
Tabel 4. 13 Pengujian Semua Parameter pada Setpoint 27.5 cm	64
Tabel 4. 14 Pengujian Semua Parameter pada Setpoint 8.5 cm	65

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pada abad ke-21, industri *Food and Beverage* (F&B) telah tumbuh pesat secara global dengan kemampuan mengikuti kemajuan teknologi, termasuk robotika. Salah satu inovasinya adalah layanan pengantaran makanan di kafe, restoran, atau kantin. Layanan ini sangat penting karena berpengaruh langsung terhadap kepuasan dan loyalitas pelanggan. Sebelumnya, pengantaran makanan sepenuhnya bergantung pada tenaga kerja manusia, namun sistem tradisional ini dinilai kurang efisien baik dari sisi biaya maupun kualitas[1][2].

Ketika mengantar makanan menggunakan perantara manusia umumnya memiliki kekurangan seperti kebersihan, keamanan serta kesalahan dalam pengiriman. Ditambah kinerja dari tenaga manusia memiliki keterbatasan sehingga dapat terjadi *human error*, hal ini tentu saja dapat merugikan pihak penyaji makanan ditambah biaya yang dikeluarkan oleh penyaji untuk membayar gaji tenaga manusia cukup besar dan akan naik dari tahun ke tahun[3]. Sedangkan hal tersebut seharusnya dapat dihindari untuk mempertahankan bisnis mereka. Berbeda jika kita menggunakan robot sebagai alat bantu atau alternatif yang dapat menggantikan fungsi manusia untuk melakukan kegiatan mengantar makanan tersebut yang cenderung tetap konsisten dalam melakukan pekerjaannya [4].

Oleh karena itu dibutuhkan sebuah robot yang dapat melakukan kegiatan pengantaran makanan tersebut, namun penelitian yang sudah ada sebelumnya robot belum sempurna masih banyak hal yang belum bisa dilakukan oleh robot tersebut seperti contohnya pergerakan yang kurang baik, *tray* yang masih bersifat statis dan belum memiliki perlindungan seperti pintu pada bagian robot tersebut. Sehingga robot tersebut belum mampu menyajikan makanan secara baik dan mempermudahkan proses pengantaran makanan kepada pelanggan tersebut [5].

Jika ditinjau berdasarkan permasalahan tersebut maka, diperlukan robot yang dapat beroperasi serupa dengan beberapa inovasi yang ditambahkan seperti *tray* yang dapat menjulur untuk menyajikan makanan maupun minuman kepada pelanggan serta mobilitas yang baik tanpa adanya kesalahan dengan mengantarkan makanan ketitik tujuan dengan tepat.

1.2. Rumusan Masalah

Kemajuan di bidang robotika telah memunculkan inovasi baru dalam industri makanan, khususnya dengan hadirnya robot pengantar makanan. Teknologi ini menawarkan peningkatan efisiensi pelayanan dan meminimalkan kontak langsung, yang menjadi sangat relevan di masa yang akan datang. Meski demikian, penerapan robot pengantar makanan di Indonesia masih menemui sejumlah hambatan, terutama terkait proses penyajian dan efisiensi proses pengantaran. Studi pendahulu mengindikasikan adanya berbagai kendala teknis yang berdampak pada kualitas layanan, meskipun telah mengadopsi sistem robot pengantar. Situasi ini memunculkan kebutuhan untuk mengkaji strategi optimalisasi penggunaan robot pengantar makanan dalam meningkatkan layanan restoran. Berdasarkan hal tersebut, rumusan masalah yang akan diteliti adalah sebagai berikut:

1. Bagaimana merancang sistem mekatronik *tray* yang mampu mengantarkan dan menyajikan makanan dengan aman kepada pelanggan?
2. Bagaimana meningkatkan efektivitas dan efisiensi proses pengantaran makanan di restoran, kafe, dan kantin melalui penerapan robotika, serta mengurangi ketergantungan pada tenaga manusia sebagai pengantar?
3. Bagaimana meningkatkan efisiensi terhadap kontrol dan monitoring *tray*?

1.3. Tujuan

Tujuan merupakan target yang harus dicapai pada suatu penelitian. Tujuan yang hendaknya dicapai dari penelitian Tugas Akhir ini yakni:

1. *Tray* pada robot mampu mengantarkan dan menghidangkan makanan dengan aman kepada pelanggan.
2. Meningkatkan efisiensi pengantaran makanan dengan sistem mekatronik *tray* untuk mengurangi ketergantungan pada tenaga manusia.

3. Mengembangkan sistem kontrol dan *monitoring tray* yang efisien dan mudah di *monitoring*.

Dengan demikian proses pengantaran makanan pada restoran, kafe dan kantin akan lebih efektif dan efisien untuk kedepannya serta memungkinkan untuk tidak menggunakan tenaga manusia sebagai perantara antara penyaji dan pelanggan.

1.4. Manfaat Hasil Penelitian

Manfaat dari hasil penelitian yang dilakukan adalah sebagai berikut:

1. Efisiensi dan Produktivitas

Robot pengantar makanan dapat mengurangi beban kerja staf restoran atau kantin, memungkinkan mereka untuk fokus pada tugas-tugas lain. Ini juga dapat meningkatkan kecepatan pengantaran makanan

2. Keselamatan

Mengurangi interaksi langsung antara staf dan pelanggan dapat membantu mengurangi risiko penyebaran penyakit, terutama di restoran yang sedang mengalami peningkatan pengunjung.

3. Pengurangan Biaya

Dengan penggunaan robot, biaya tenaga kerja dapat dikurangi, dan juga dapat mengurangi kehilangan makanan yang mungkin terjadi selama pengantaran.

1.5. Batasan Masalah

Pada penelitian ini diberikan batasan masalah agar penelitian dapat berjalan sesuai tujuan dan efektif. Batasan masalah yang diberlakukan, antara lain:

1. Penelitian ini berfokus pada pengembangan sistem mekatronik dan kontrol *tray* pada robot pengantar makanan.
2. Sistem *tray* akan bergerak maju dan mundur ketika melakukan proses penyerahan makanan.
3. Sistem *tray* bergerak dengan menggunakan aktuator Motor DC
4. Robot dapat bermanuver ke berbagai arah dengan menggunakan roda dengan bantuan sensor.
5. Pengujian hanya dilakukan pada lingkungan Tel-U Café
6. *Tray* hanya dapat menopang beban maksimal 2 Kg

7. Ukuran gelas 16 Oz (9cm x 10,5 cm) atau 470 ml
8. Diameter piring bagian luar 12,5 cm dan bagian dalam 6,5 cm

1.6. Metode Penelitian

Dalam melakukan penyusunan dan penggerjaan tugas akhir ini, metode penelitian yang digunakan adalah sebagai berikut:

1. Studi Literatur

Penelitian diawali dengan pengumpulan data dan analisis dari berbagai sumber berupa jurnal dan publikasi ilmiah mengenai konsep dan sistem terkait pembahasan mekanisme penggerak *tray* pada robot serta penggerak robot. Pada tahap ini bertujuan agar mendapatkan pemahaman mendalam mengenai sistem penggerak *tray* dan robot dan mengetahui kelemahan dan keunggulan sistem yang dibuat.

2. Pengujian

Pengujian yang dilakukan pada tugas akhir ini berupa pengujian terhadap sensor-sensor yang digunakan serta aktuator yang terdapat pada sistem. Sensor dan aktuator yang diuji nantinya akan memiliki parameter serta pergerakan yang tepat sehingga dapat berfungsi untuk mengambil data yang akurat dan pergerakan yang optimal.

3. Permodelan dan Simulasi

Simulasi yang dilakukan pada tugas akhir ini menggunakan *software* Autodesk fusion. Simulasi berisikan tentang desain bentuk dari *Tray* serta ukurannya yang akan di letakan pada badan robot, serta melakukan simulasi mengenai bagaimana cara sistem tersebut yang nantinya akan bergerak. Lalu mendesain bentuk serta ukuran *base* yang akan dipakai sebagai wadah untuk meletakan komponen sistem robot.

4. Analisis

Hasil data dari pengukuran dan perhitungan akan melalui tahap analisis untuk melihat dan menilai hasil dari perhitungan tersebut akurat dan tepat. Analisis ini meliputi pengukuran terhadap sudut dan posisi robot serta pergerakan aktuator untuk menggerakkan *tray*.

5. Penulisan Buku

Penulisan buku tugas akhir ini bertujuan sebagai hasil dari laporan penelitian tugas akhir. Penulisan buku tugas akhir ini berisikan tentang hasil dan Kesimpulan terkait penelitian yang menghasilkan analisis dan solusi dari permasalahan yang telah dipaparkan.

1.7. Proyeksi Pengguna

Target serta proyeksi pengguna yang akan memanfaatkan hasil dari penelitian ini ialah restoran, kafe dan kantin karena dasar dari pembuatan robot ini adalah pengiriman makanan kepada pengunjung khususnya pada Tel-U Cafe.

BAB II

TINJAUAN PUSTAKA

Pada Bab II menyajikan tinjauan Pustaka mengenai konsep-konsep dasar, teknologi, dan metode yang digunakan sebagai dasar dari penelitian ini yang berjudul *PENGEMBANGAN SISTEM INTERNET OF THINGS (IoT) UNTUK MONITORING DAN KONTROL TRAY OTOMATIS PADA ROBOT PENGANTAR MAKANAN*. Tinjauan Pustaka ini memiliki tujuan untuk memberikan pemahaman mendalam mengenai komponen utama dan metode yang digunakan.

2.1 Robotika

Robotika adalah cabang ilmu yang berfokus pada desain, pembuatan, operasi, dan penggunaan robot. Robot adalah mesin yang dapat diprogram ulang untuk melakukan berbagai tugas, baik secara otomatis maupun dengan kendali jarak jauh [6][7]. Secara umum, robotika memadukan berbagai disiplin ilmu seperti ilmu komputer, elektronika, dan mekanika untuk menghasilkan mesin yang mampu menjalankan tugas-tugas tertentu secara otomatis dan efisien tanpa memerlukan campur tangan manusia[8]. Dalam perkembangan modern, robotika tidak hanya terbatas pada perangkat mekanis, tetapi juga mencakup sistem yang dilengkapi dengan kecerdasan buatan (AI) dan sensor untuk berinteraksi dengan lingkungan sekitar mereka[7][8][9].

Robot dapat dikategorikan berdasarkan fungsi dan aplikasi mereka, antara lain:

- a. Robot Industri: Digunakan dalam proses produksi untuk melakukan tugas-tugas berat atau berulang.
- b. Robot Medis: Digunakan dalam bidang kesehatan untuk membantu dalam prosedur medis atau rehabilitasi.
- c. Robot Penjelajah: Didesain untuk menjelajahi lingkungan yang ekstrem, seperti luar angkasa atau dasar laut.
- d. Robot Layanan: Berfungsi dalam interaksi dengan manusia dalam berbagai konteks, seperti layanan pelanggan atau pembersihan[3][4].

2.1.1 BellaBot

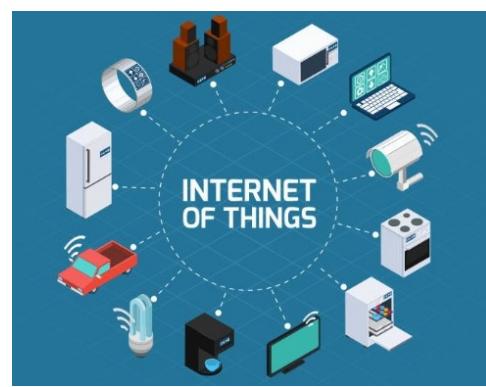
BellaBot adalah salah satu jenis robot pengantar makanan yang dikembangkan oleh perusahaan teknologi yang berada di China bernama Pudu Robotics.



Gambar 2. 1 Bellabot[10]

BellaBot dirancang khusus untuk melayani kebutuhan industri perhotelan, restoran, dan kafe. BellaBot berfungsi sebagai robot pengantar makanan yang bertugas membawa makanan atau minuman dari dapur ke meja pelanggan secara otomatis. Robot ini dirancang untuk meningkatkan efisiensi layanan, mengurangi kesalahan manusia, dan memberikan pengalaman unik bagi pelanggan[3][4][10].

2.2 *Internet Of Things*



Gambar 2. 2 *Internet of Things*[11]

Internet of Things (IoT) adalah jaringan yang menghubungkan perangkat fisik, objek, dan sistem sehingga dapat berkomunikasi melalui internet tanpa perlu

campur tangan manusia. Perangkat-perangkat ini dilengkapi dengan sensor, aktuator, dan teknologi komunikasi yang memungkinkan mereka untuk mengumpulkan, mengirimkan, dan menganalisis data secara langsung. Tujuan utama IoT adalah mengintegrasikan dunia fisik dan digital, sehingga mendukung proses yang lebih cerdas dan pengambilan keputusan yang lebih efektif[12].

2.2.1 Kodular

Kodular adalah platform berbasis web yang mirip dengan MIT *App Inventor*, dirancang untuk mempermudah pembuatan aplikasi Android menggunakan metode block programming. Salah satu keunggulan Kodular adalah menyediakan komponen palet yang lebih beragam dan kompleks, serta dilengkapi dengan fitur monetisasi bawaan. Pengguna tidak perlu menginstal perangkat lunak tambahan, cukup menggunakan browser untuk mengaksesnya. Dalam pembuatan aplikasi, pengguna hanya perlu mengatur parameter dan menyusun blok program dengan metode *drag and drop*, tanpa harus menulis kode dari awal. Kodular *Companion* memungkinkan pengguna untuk dapat langsung menguji aplikasi yang sedang dikembangkan tanpa perlu mengekspornya terlebih dahulu, sehingga perbaikan dapat dilakukan secara langsung apabila terdapat kesalahan pada blok program. Platform ini dirancang untuk menciptakan aplikasi Android dengan cara yang lebih cepat dan efisien[13].

2.3 Mekanisme *Tray*

Tray atau nampan pada robot pengantar makanan berfungsi sebagai platform untuk menampung makanan yang akan diantarkan. Mekanisme *tray* biasanya dirancang agar dapat menampung berbagai jenis makanan dan minuman, serta memastikan bahwa makanan tersebut tidak tumpah selama perjalanan. Beberapa robot memiliki beberapa lapisan *tray*, memungkinkan mereka untuk mengangkut lebih banyak item sekaligus. Misalnya, robot seperti BellaBot dilengkapi dengan empat lapisan *tray* yang dapat menampung hingga 10 kg per lapisan[10].

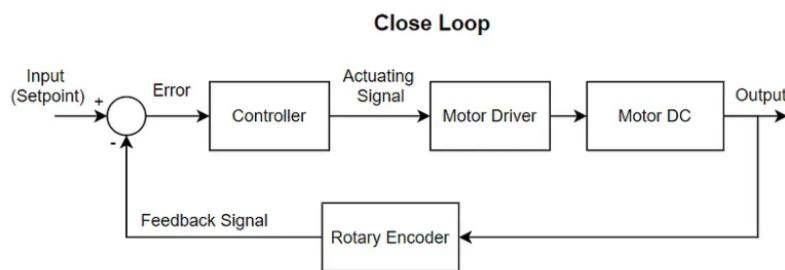
2.4 Sistem Kontrol

Sistem kontrol adalah sistem yang dirancang untuk mengelola dan mengatur perilaku perangkat atau proses tertentu agar mencapai tujuan atau nilai yang

diinginkan (*setpoint*). Sistem ini terdiri dari kumpulan komponen yang bekerja secara terintegrasi untuk mengontrol variabel tertentu, seperti suhu, tekanan, atau kecepatan, sesuai dengan masukan yang diberikan. Sistem kontrol dapat beroperasi baik secara manual maupun otomatis, dengan elemen utama yang mencakup sensor, pengontrol, aktuator, dan mekanisme umpan balik. Sistem ini memiliki aplikasi luas di berbagai bidang, termasuk industri, otomasi, dan teknologi[14]. Pada sistem kontrol terdapat dua jenis yakni sistem kontrol *loop* tertutup dan *loop* terbuka berikut penjelasannya:

1) *Loop* Tertutup

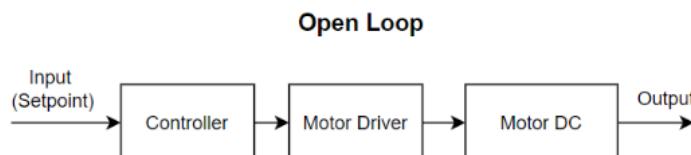
Sistem kendali *loop* tertutup, atau yang sering dikenal sebagai sistem kendali umpan balik, adalah jenis sistem kendali di mana nilai keluarannya secara langsung mempengaruhi tindakan pengendalian.



Gambar 2. 3 Blok Diagram *Loop* Tertutup[15]

2) *Loop* Terbuka

Sistem kendali yang memiliki ciri khas bahwa nilai keluarannya tidak memengaruhi tindakan pengendalian disebut sebagai sistem kendali *loop* terbuka[15].



Gambar 2. 4 Blok Diagram *Loop* Terbuka[15]

Perbedaan utama antara sistem kontrol *loop* terbuka dan *loop* tertutup terletak pada keberadaan umpan balik (*feedback*) dalam sistem. Pada sistem yang menggunakan umpan balik, *output* akan memengaruhi proses kontrol secara

langsung. Dalam banyak kasus, sistem kontrol memanfaatkan *error* sebagai bentuk umpan balik. *Error* ini adalah selisih antara nilai yang diinginkan *Setpoint* (SP) dan nilai aktual *Present Value* (PV)[14].

Sistem kontrol otomatis (*Automation Control System*) adalah sekumpulan perangkat mekanik atau elektronik yang berfungsi untuk mengatur operasi suatu perangkat atau sistem melalui mekanisme kontrol tertutup. Sistem ini umumnya terkomputerisasi dan beroperasi secara mandiri tanpa intervensi manual [15].

2.4.1 Sistem Kontrol PID

Proportional Integral Derivative (PID) merupakan salah satu metode yang terdapat pada sistem kontrol. Kontrol PID adalah alat yang menerima data masukan dari sensor, menghitung perbedaan antara nilai aktual dan titik pengaturan yang diinginkan, lalu menyesuaikan keluaran untuk mengendalikan variabel seperti suhu, laju aliran, kecepatan, tekanan, dan tegangan. Kontroler ini melakukan ini melalui tiga mekanisme:

1. Kontrol proporsional, yang bereaksi terhadap kesalahan saat ini.

$$P_{term} = K_p \cdot e(t) \quad (1)$$

2. Kontrol integral, yang menangani kesalahan yang terakumulasi dari masa lalu.

$$I_{term} = K_i \cdot \int_0^{\tau} e(\tau) dt \quad (2)$$

3. Kontrol derivatif, yang memprediksi kesalahan masa depan.

$$D_{term} = K_d \cdot \frac{de(t)}{dt} \quad (3)$$

Kontroler PID menggabungkan ketiga komponen ini untuk menghitung keluaran. Arsitektur ini memungkinkan kontroler PID mempertahankan kontrol proses dan stabilitas sistem secara efisien[16][17]. Uraian tersebut dapat dijelaskan dalam rumus:

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

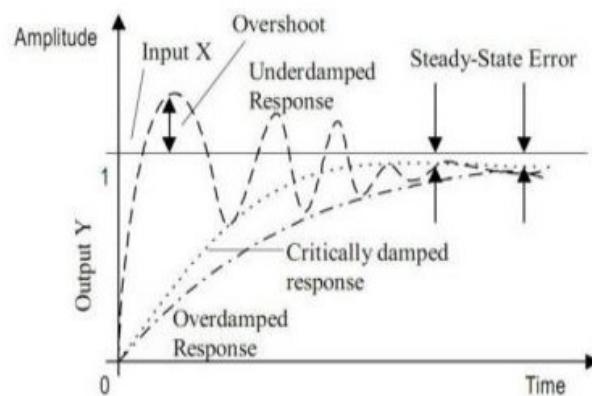
Di mana:

- $u(t)$ adalah dorongan yang datang dari kontroler, ke dalam proses, pada waktu t
- $e(t) = y_{sp}(t) - y(t)$ adalah perbedaan antara *setpoint* dan variabel proses yang diukur pada waktu t
- K_p, K_i, K_d masing-masing adalah konstanta P, I , dan D

Kinerja algoritma PID sangat ditentukan oleh kecocokan parameter PID yang dipilih. Jika konstanta PID sesuai dengan proses, kontrol akan berkonvergensi dengan halus. Namun, jika konstanta tersebut kurang tepat, sistem bisa mengalami osilasi, menjadi tidak stabil, atau kehilangan kontrol. Penetapan konstanta PID pada akhirnya ditentukan oleh pengguna dan dapat disempurnakan melalui kombinasi algoritma penyesuaian dan uji coba. Dua metode populer untuk penyetelan PID adalah metode Ziegler-Nichols dan Åström-Hägglund[18].

2.4.2 Respon Transien

Ketika *input* suatu sistem mengalami perubahan mendadak, keluaran atau *output* sistem memerlukan waktu tertentu untuk menyesuaikan atau merespon perubahan tersebut. Perilaku sistem selama proses penyesuaian ini dikenal sebagai *respon transien* atau transisi ditunjukkan dengan gambar dibawah ini.



Gambar 2. 5 Bentuk Sinyal Tanggapan Transien [19]

Pendekatan yang paling umum digunakan dalam sistem kendali adalah sistem orde kedua, yang memiliki beberapa parameter untuk mengukur kualitas respon transien, yaitu sebagai berikut:

1. *Rise Time* (TR): Waktu yang diperlukan bagi respons untuk pertama kali mencapai nilai *steady-state*, diukur dari $t = 0$ hingga respon memotong sumbu *steady-state*.
2. *Settling Time* (TS): Durasi waktu yang menunjukkan respon telah berada dalam kisaran 5%, 2%, atau 0,5% dari nilai *steady-state*.
3. *Delay Time* (TD): Waktu yang menunjukkan keterlambatan respon *output* terhadap *input*, dihitung dari $t = 0$ hingga respons mencapai 50% dari nilai *steady-state*.
4. *Overshoot* (MP): Besarnya nilai maksimum dari respon yang melebihi nilai *steady-state*, dinyatakan sebagai perbandingan relatif terhadap nilai *steady-state*.
5. *Offset*: Perbedaan atau *error* antara nilai yang diinginkan dan nilai aktual yang dicapai oleh sistem[19].

BAB III

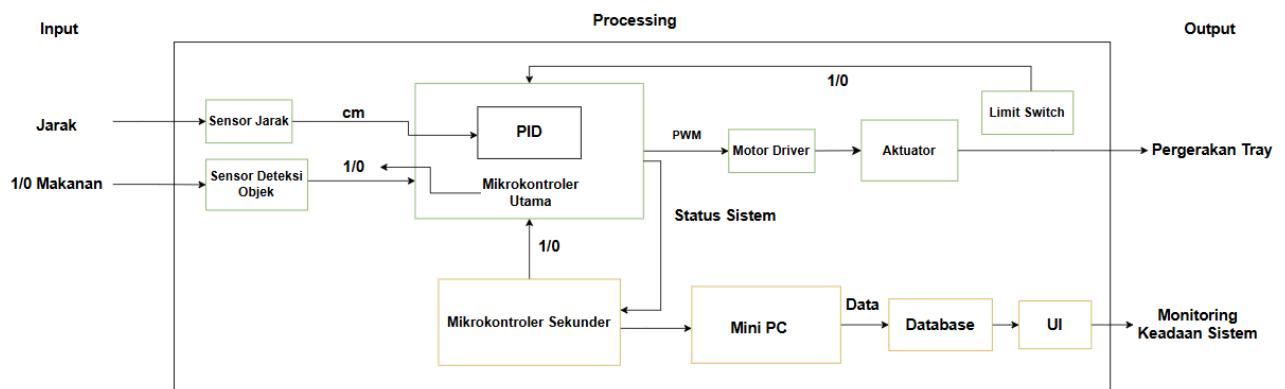
PERANCANGAN SISTEM

3.1 Desain Sistem

Pada desain sistem menggambarkan bagaimana *tray* pada robot dapat menjulurkan nampan untuk memberikan makanan ataupun minuman kepada konsumen. Sistem yang dirancang untuk memberikan makanan dan minuman dengan menggunakan motor DC *wormgear* sebagai aktuator yang akan menggerakkan *tray* untuk bergerak keluar dari dalam tubuh robot. Kemudian sistem akan mendeteksi batas keluar dari *tray* yang akan membuat *tray* berhenti bergerak dengan menggunakan kontrol PID dan sensor *Limit Switch*. Kemudian sensor IR yang berada didalam tubuh robot yang berada di tengah nampan letak penyimpanan makanan sudah mendeteksi bahwa makanan pada *tray* sudah kosong yang akan membuat *tray* akan bergerak masuk kembali kedalam tubuh robot .

3.1.1 Diagram Blok

Pada diagram blok berisikan tentang alur atau jalannya suatu sistem yang dirancang dan memiliki beberapa *input* untuk sensor dan aktuator serta proses yang terjadi pada sistem utama sebagai berikut:



Gambar 3. 1 Blok Diagram Sistem

Gambar 3.1 menunjukkan bagaimana sistem bekerja dengan menerima *input* dari sensor dan sistem akan memiliki *output* berupa aktuator seperti pada gambar. Berikut merupakan penjelasan mengenai proses dari jalannya sistem seperti pada gambar:

- a. Pada bagian *tray* memiliki dua sensor yakni *Ultrasonic* dan IR Sensor yang memiliki *input* berupa jarak (cm) untuk memproses PID dan sensor IR untuk mendeteksi kondisi benda berupa makanan atau minuman didalam *tray* dalam bentuk biner (1 atau 0) yang dihubungkan pada Arduino Mega Pro 2560.
- b. Pada Arduino Mega Pro 2560 terdapat proses kontrol PID yang datanya didapat dari sensor *Ultrasonic* dalam bentuk jarak (cm) yang berfungsi untuk mengatur kecepatan putaran motor pada *tray* agar tidak mengalami hentakan sehingga makanan atau minuman didalamnya dalam keadaan baik dan tidak tumpah atau berceciran. Lalu data tersebut dikirimkan ke Motor DC berupa PWM atau *pulse width modulation* ke *Motor Driver* yang kemudian disambungkan ke *Motor DC* yang kemudian memiliki *output* berupa pergerakan *tray*.
- c. Sensor *Limit switch* berfungsi sebagai saklar mekanik yang berfungsi sebagai pengaman gerakan dari *tray*. Pada saat *tray* bergerak keluar, ujung bagian dari *tray* akan menyentuh sensor *Limit switch* yang nantinya akan langsung memberikan sinyal untuk menghentikan pergerakan motor DC guna mencegah terjadinya kerusakan pada motor.
- d. Kemudian Kodular berfungsi sebagai *platform* ataupun UI untuk dapat memonitoring *tray* dengan cara mengambil *database* dari Firebase yang datanya diperoleh dari ESP32 dan Arduino Mega Pro 2560 Mini yang dikirim melalui Raspberry Pi.

3.1.2 Fungsi dan Fitur

Sesuai dengan implementasi alat yang sudah dirancang memiliki beberapa fungsi sebagai berikut:

- a. Melakukan kontrol pergerakan otomatis dari aktuator berupa motor DC agar *tray* dapat bergerak keluar dan masuk.

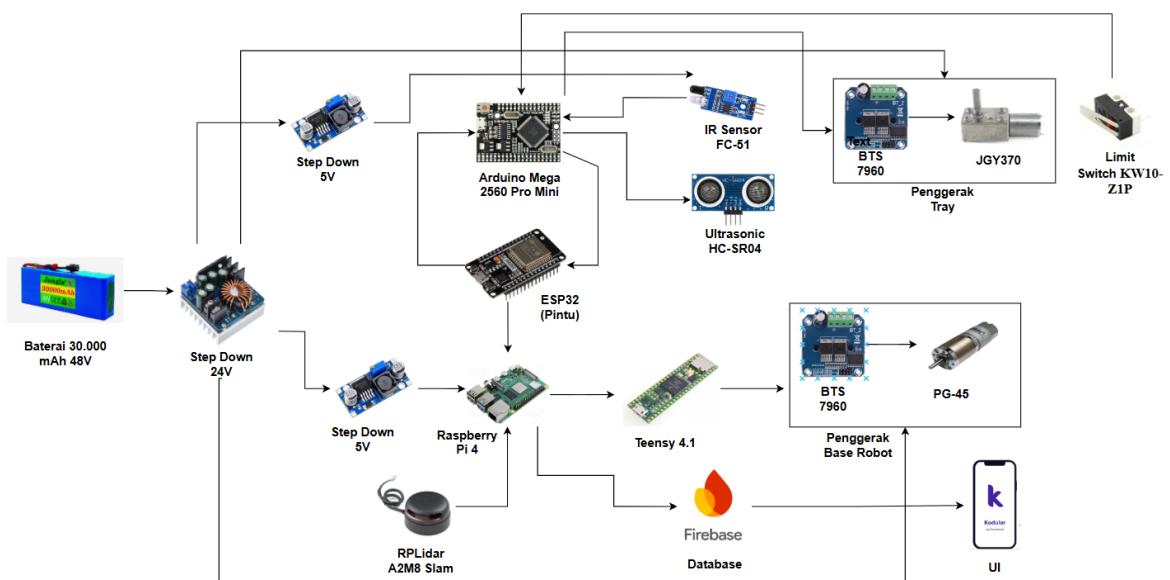
b. Meningkatkan keakuratan pergerakan *tray* saat beroperasi

Selain fungsi diatas, terdapat beberapa fitur yang akan dirancang, yaitu:

- Menampilkan status atau *monitoring* yang dapat memberikan informasi *tray* pada *smartphone*.
- Menjalankan beberapa algoritma dan proses secara bersamaan dan terhubung satu sama lain.

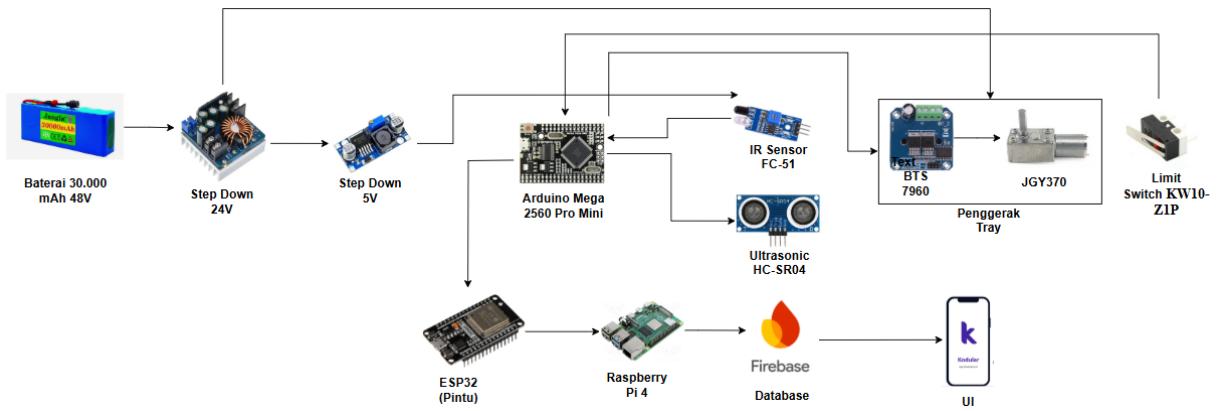
3.2 Desain Perangkat Keras

Pada sistem yang dirancang digunakan beberapa komponen-komponen perangkat keras seperti sensor, aktuator, mikrokontroler, *motor driver*, penurun tegangan dan baterai. Pada robot yang dirancang terdapat komponen yang digunakan seperti gambar dibawah ini.



Gambar 3. 2 Blok Diagram Perangkat Keras

Pada gambar 3.2, menampilkan rangkaian dari komponen yang digunakan. Baterai 30000 mAh 48V sebagai sumber daya utama untuk menyalurkan arus listrik menuju *motor driver* dan mikrokontroler Arduino Mega 2560 Pro mini, ESP32, Raspberry Pi dan Teensy. Untuk motor DC *wormgear* yang digunakan sebagai penggerak *Tray*, menggunakan tegangan 12V yang sebelumnya sudah di turunkan menggunakan *stepdown* dari baterai.



Gambar 3. 3 Diagram Sistem *Tray*

Sedangkan untuk sistem yang dibuat dapat dilihat pada gambar 3.3 dengan menggunakan Arduino Mega PRO Mini sebagai mikrokontroller utama untuk menerima *input* dari ESP32 dan sensor yang digunakan kemudian memberikan perintah berdasarkan *input* sensor kepada aktuator unutk menggerakan *tray*.

3.3 Desain UI

Berikut merupakan desain dari *user interface* yang dibuat menggunakan platform kodular yang terdiri dari 2 layar:

GO FOR BETTER HABITS
WITH T.I.F.A!

TEL-U INTERACTIVE FOOD ASSISTANT



Gambar 3. 4 Desain UI Screen 1



Gambar 3. 5 Desain UI Screen 2

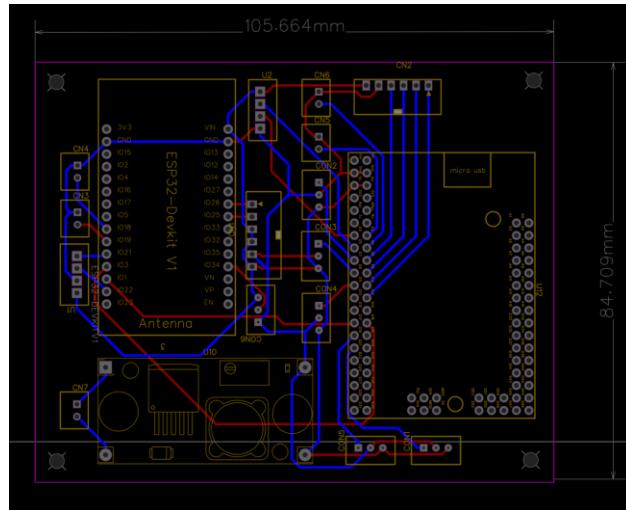
Desain UI pada gambar 3.4 dan gambar 3.5 ini dibuat sebagai *platform* untuk dapat memantau status dari kondisi robot ketika beroperasi secara *real-time* yang terdiri dari:

1. Status Pintu (Terbuka/Tertutup)
2. Status *Tray* (Keluar/Masuk)
3. Status Makanan (Ada/Tidak)

Data dari robot di peroleh melalui database yakni Firebase sebagai tempat untuk memproses data secara *real-time* yang dikirimkan oleh robot.

3.4 Desain PCB

Berikut merupakan desain PCB yang digunakan untuk meletakan mikrokontroler dan sensor yang digunakan pada proses pembuatan *tray* pada robot yang diletakan berada pada rangka *base*.



Gambar 3. 6 Desain wiring PCB

Pada gambar 3.6 merupakan desain rangkaian dari PCB yang dibuat untuk tugas akhir ini. Desain PCB diatas dibuat untuk meletakan mikrokontroller seperti ESP32, Arduino Mega 2560 Pro Mini, *step down 5V*, sensor IR FC-51, sensor *Ultrasonic HC-SR04*, serta *output* untuk aktuator yakni *motor driver* yang dihubungkan ke motor DC. PCB yang dibuat menggunakan tegangan 5V untuk menjalankan mikrokontroller dan sensor-sensor yang digunakan. Sedangkan untuk *motor driver* menggunakan tegangan 12V yang langsung dihubungkan tanpa melewati *stepdown 5V*.

3.5 Desain Fisik Robot

Berikut merupakan desain 3D *base* dan *body* robot yang digunakan dalam penelitian ini.



Gambar 3. 7 Desain Prototipe robot tampak samping kanan

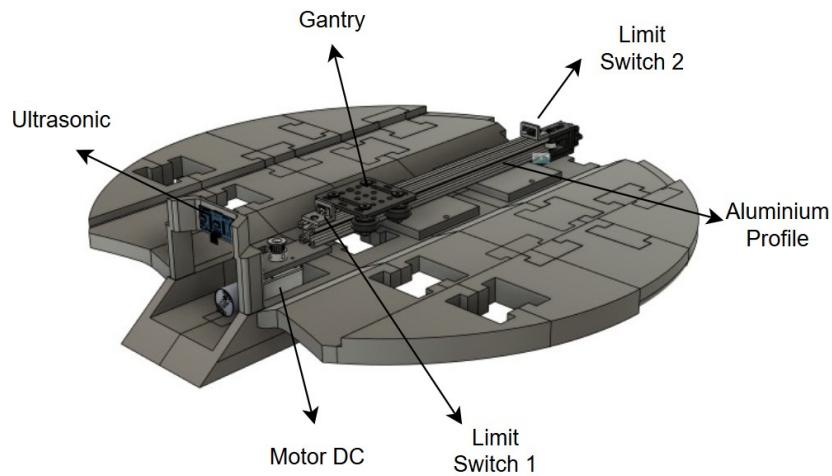


Gambar 3.8 Desain Prototipe robot tampak depan

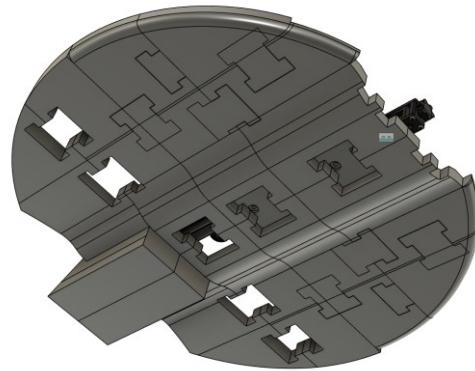
Pada gambar 3.7 terlihat bentuk desain dari rangka mekanisme *tray* yang terpasang pada bagian dalam badan tubuh robot, sedangkan pada gambar 3.8 terlihat desain penuh bagian robot pada tampak depan.

3.5.1 Desain *Tray* Pada Robot

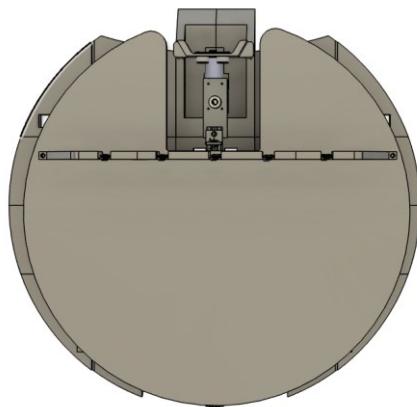
Berikut merupakan desain dari *tray* yang akan dirancang yang nantinya akan diletakan pada bagian tubuh robot:



Gambar 3.9 Tampak Atas *Tray* Aktif

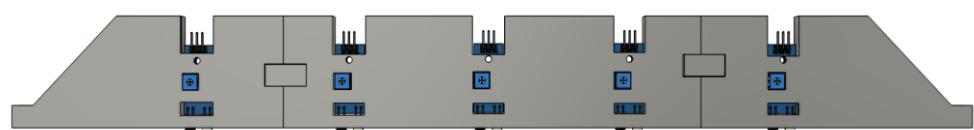


Gambar 3. 10 Tampak Bawah *Tray Aktif*

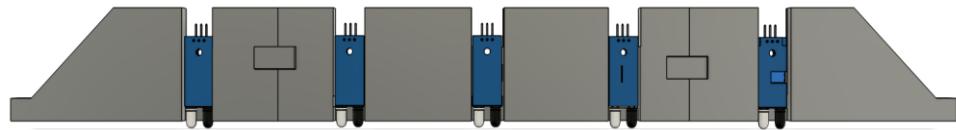


Gambar 3. 11 Tampak Atas *Tray Aktif* dengan nampan

Pada gambar 3.9 dan gambar 3.10 dan gambar 3.11 terlihat desain *tray aktif* yang memungkinkan untuk dapat menopang beban makanan hingga mencapai berat 2 Kg. *Tray pasif* diletakan pada bagian atas *gantry* kemudian diatasnya di sambungkan dengan *bracket sensor infrared*.



Gambar 3. 12 Tampak Depan *Bracket sensor IR*

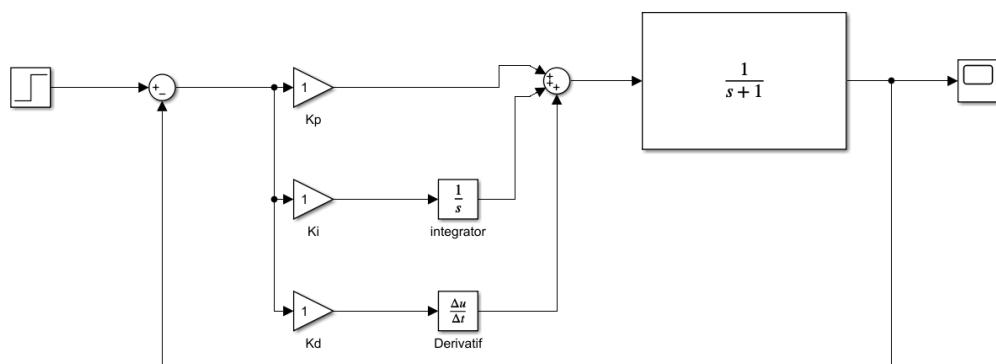


Gambar 3. 13 Tampak Belakang *Bracket sensor IR*

Pada gambar 3.12 dan gambar 3.13 merupakan desain dari *bracket* sensor *infrared* yang akan digunakan untuk meletakan sensor *infrared* yang digunakan untuk mendeteksi adanya makanan dan minuman pada *tray* di bagian dalam tubuh robot.

3.6 Desain Parameter PID

Berikut merupakan proses perancangan parameter PID yang digunakan untuk mengontrol pergerakan *tray* pada robot. Perancangan ini dilakukan untuk menentukan nilai konstanta K_p , K_i dan K_d yang optimal sehingga sistem dapat mencapai *setpoint* dengan respon yang cepat, akurat, dan minim *overshoot*. Proses penentuan parameter dilakukan melalui pendekatan teoritis dan pengujian pada *software Matlab* untuk mendapatkan performa kontrol terbaik sesuai karakteristik sistem.

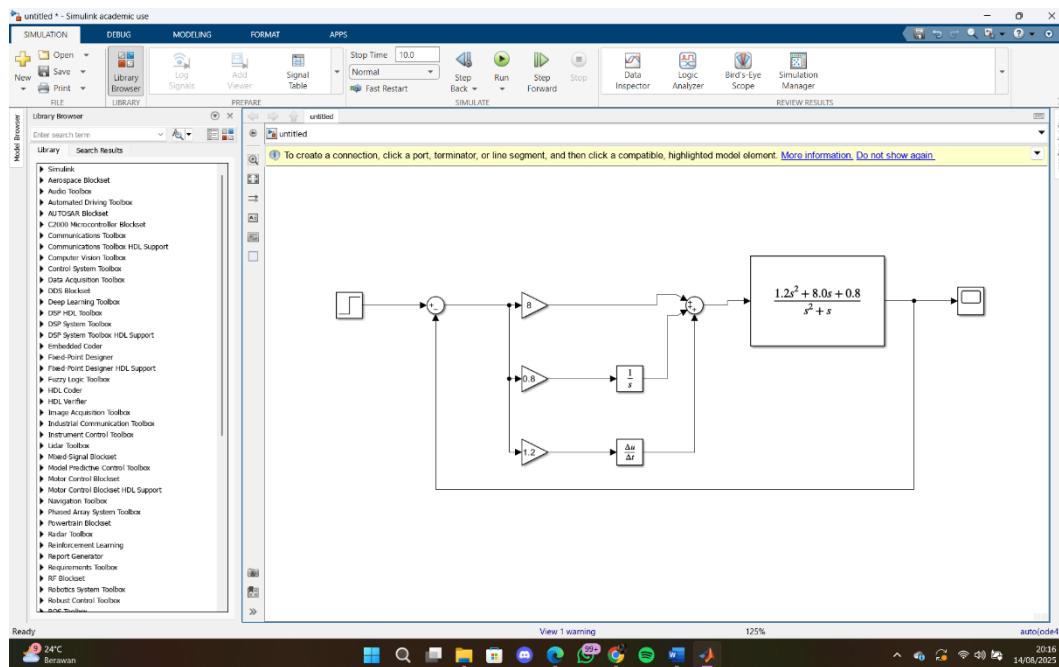


Gambar 3. 14 Diagram blok pengontrol PID

Pada gambar 3.14 terlihat bentuk diagram blok pengontrol PID yang terdiri dari *step*, *sum*, *gain*, *integrator*, *derivative*, *transfer function* dan *scope*. Komponen yang digunakan nantinya diisi dengan nilai parameter PID yang akan digunakan.

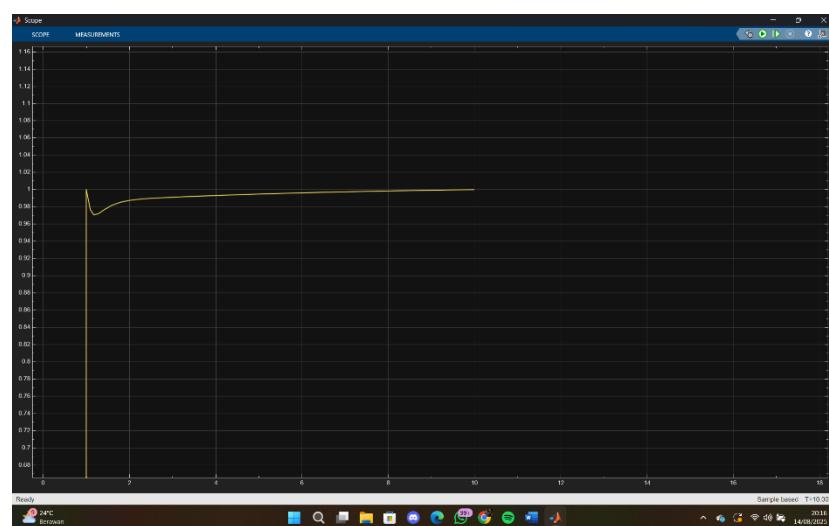
3.6.1 Parameter Desain Kp=8.0, Ki=0.8, Kd=1.2

Parameter pengujian pertama ialah dengan menggunakan Kp=8.0, Ki=0.8, Kd=1.2. Nilai tersebut diperoleh melalui proses menggunakan nilai yang ditingkatkan terdahulu dengan diagram blok PID sebagai berikut:



Gambar 3. 15 Bentuk *Transfer Function* Desain 1

Pada gambar 3.15 diatas pengujian menggunakan software Matlab, parameter yang ditentukan dimasukan ke dalam *transferfunction* dan didapat grafik sebagai berikut:

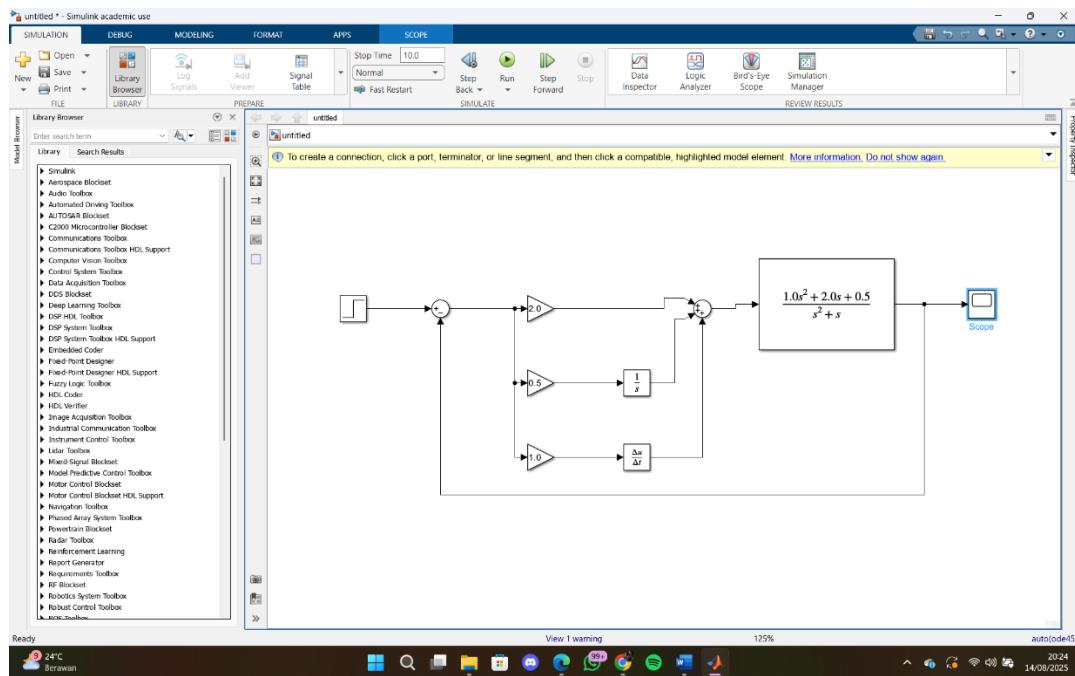


Gambar 3. 16 Bentuk grafik simulasi PID Desain 1

Dari pengujian gambar 3.16 diatas dapat dilihat jika terdapat *undershoot* yang cukup besar setelah *step input*. Lalu waktu *rise time* yang terlihat cukup cepat dan nilai *steady state* tercapai agak lambat dengan adanya sedikit *offset*. Terlihat nilai K_p yang tinggi menimbulkan respon yang cepat namun ada sedikit osilasi, kemudian nilai K_d yang ada yang cukup besar (menekan *overshoot* namun menimbulkan *undershoot*), dan nilai K_i yang rendah memperlambat perbaikan *steady state error*.

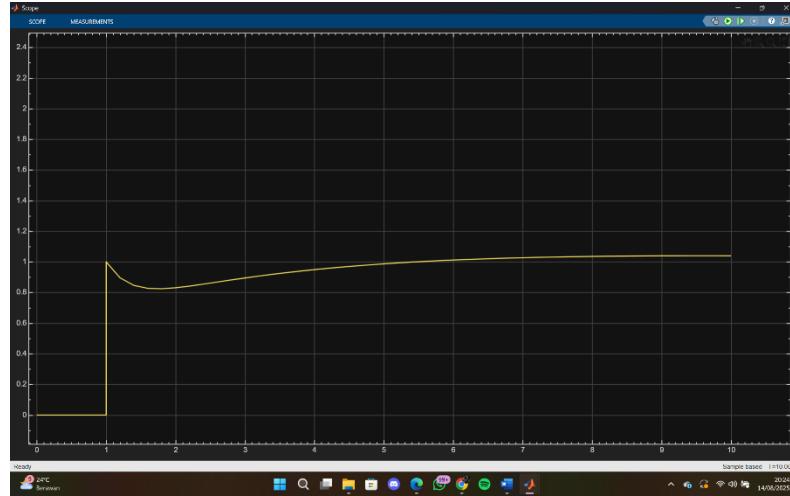
3.6.2 Parameter Desain $K_p=2.0$, $K_i=0.5$, $K_d=1.0$

Parameter pengujian kedua ialah dengan menggunakan $K_p=2.0$, $K_i=0.5$, $K_d=1.0$. Nilai tersebut diperoleh melalui proses menggunakan nilai yang diturunkan yang sebelumnya menggunakan $K_p=8.0$, $K_i=0.8$, $K_d=1.2$ dengan diagram blok PID sebagai berikut:



Gambar 3. 17 Bentuk Transfer Function Desain 2

Pada gambar 3.17 diatas pengujian menggunakan *software Matlab*, parameter yang ditentukan dimasukan ke dalam *transferfunction* dan didapat grafik sebagai berikut:

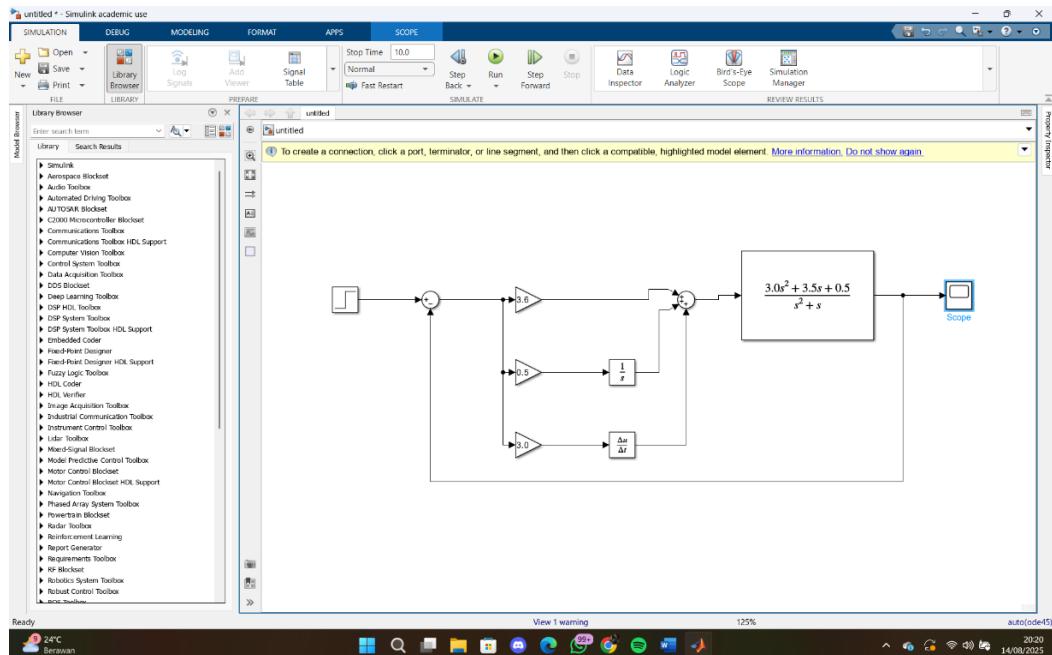


Gambar 3. 18 Bentuk grafik simulasi PID Desain 2

Dari pengujian gambar 3.18 diatas dapat dilihat jika terdapat *undershoot* yang tidak terlalu besar setelah *step input*. Lalu waktu *rise time* yang terlihat cukup cepat dan nilai *steady state* cukup kecil namun masih ada perbaikan agak lambat dengan adanya sedikit *offset*. Terlihat jika nilai K_p yang lebih kecil membuat respon lebih halus namun cenderung lambat, kemudian nilai K_d yang lebih rendah digunakan untuk mengurangi *undershoot* namun nilai K_i yang kecil berguna untuk memperbaiki *steady state* namun agak lama.

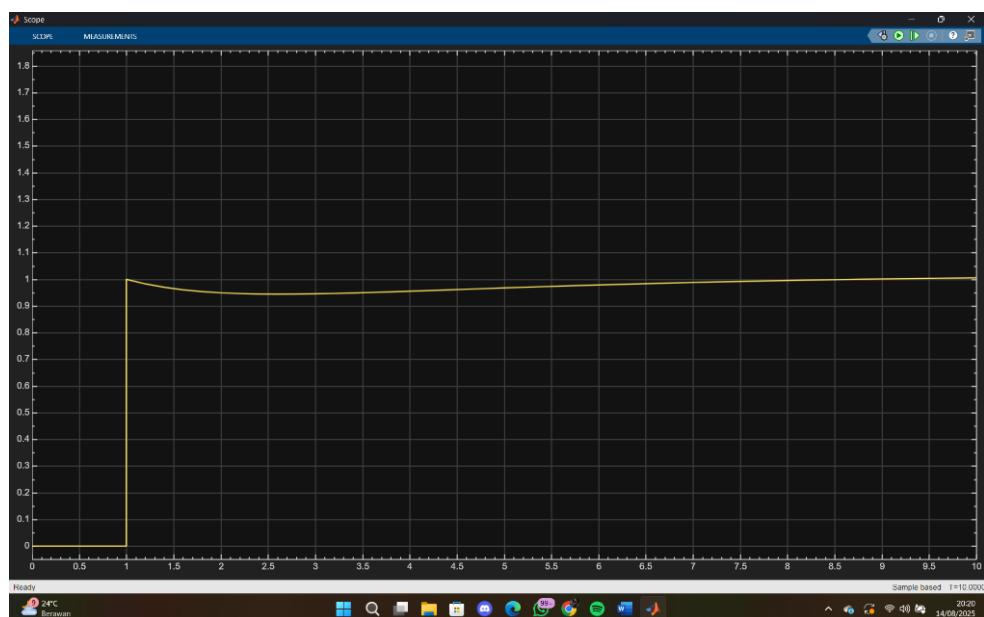
3.6.3 Parameter Desain $K_p=3.6$, $K_i=0.5$, $K_d=3.0$

Parameter pengujian kedua ialah dengan menggunakan $K_p=3.6$, $K_i=0.5$, $K_d=3.0$. Nilai tersebut diperoleh melalui proses menggunakan nilai K_p yang ditingkatkan dan nilai K_d yang ditingkatkan dengan diagram blok PID sebagai berikut:



Gambar 3. 19 Bentuk Transfer Function Desain 3

Pada gambar 3.19 diatas pengujian menggunakan *software* Matlab, parameter yang ditentukan dimasukan ke dalam *transferfunction* dan didapat grafik sebagai berikut:



Gambar 3. 20 Bentuk grafik simulasi PID Desain 3

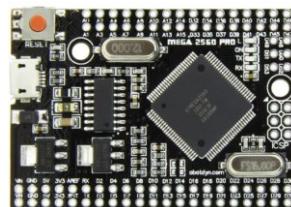
Dari pengujian gambar 3.20 diatas dapat dilihat jika *undershoot* yang ada cukup sedikit. Lalu waktu *settling* yang terlihat cukup cepat dan nilai *steady state* hampir mendekati nol. Hal ini menunjukan respon yang paling stabil diantara ketiganya. Terlihat jika nilai K_p cukup yang menyebabkan respon yang baik, lalu untuk nilai K_i sudah cukup pas yang berguna untuk menghilangkan *error steady state* tanpa membuat osilasi dan nilai K_d yang sudah optimal guna menekan *overshoot*.

3.7 Spesifikasi Komponen

Adapun terdapat beberapa komponen yang digunakan pada penelitian ini yaitu:

1. Arduino Mega 2560 Pro Mini

Arduino Mega 2560 Pro Mini adalah varian *compact* dari mikrokontroler Arduino Mega 2560 yang dirancang untuk aplikasi yang membutuhkan banyak pin I/O dalam ruang terbatas. Dengan ukuran yang lebih kecil namun tetap mempertahankan spesifikasi utama, mikrokontroler ini sangat cocok digunakan untuk proyek yang memerlukan performa tinggi dan fleksibilitas.



Gambar 3. 21 Arduino Mega 2560 Pro Mini

Tabel 3. 1 Spesifikasi Arduino Mega 2560 Pro Mini

Parameter	Keterangan
<i>Core</i>	ATmega2560
<i>Max Clock Speed</i>	16MHz
<i>Flash</i>	256KB
<i>SRAM</i>	8KB
<i>Input Voltage</i>	6-12V

<i>Operation Voltage</i>	5V
<i>Analog I/O</i>	16 Pin
<i>Digital I/O</i>	54 Pin

2. ESP-32 Dev-Kit

ESP-32 Dev-Kit merupakan sebuah papan pengembangan yang dirancang dengan menggunakan *chip* ESP-WROOM-32, yang dikenal karena kemampuannya dalam mendukung aplikasi *Internet of Things* (IoT) secara efisien. Papan ini dilengkapi dengan fitur konektivitas nirkabel, termasuk Wi-Fi dan Bluetooth, yang memungkinkan pengguna untuk mengembangkan berbagai proyek dengan mudah dan cepat. Dengan kemampuan ini, ESP-32 Dev-Kit sangat ideal untuk berbagai aplikasi yang memerlukan komunikasi tanpa kabel, seperti pengendalian perangkat secara jarak jauh, pengumpulan data dari sensor, dan integrasi dengan layanan *cloud*.



Gambar 3. 22 ESP-32 Dev-Kit

Tabel 3. 2 Spesifikasi ESP-32 Dev-Kit

Parameter	Keterangan
<i>Microprosesor</i>	Xtensa Dual-Core 32 Bit LX6
<i>Operating Voltage</i>	3.0-3.6V
<i>Input Voltage</i>	7-12V
Pin	32 Pin
<i>Flash Memory</i>	4 MB
SRAM	520 kB
Bluetooth	4.2 (BLE)

Wifi Protocols	802.11 b/g/n (802.11n up to 150 Mbps)
----------------	---------------------------------------

3. Step Down LM2596

Step down adalah perangkat listrik yang dirancang untuk menurunkan tegangan dari tingkat yang lebih tinggi ke tingkat yang lebih rendah. Dalam transformator ini, jumlah lilitan pada kumparan primer lebih banyak dibandingkan dengan kumparan sekunder. Dengan cara ini, transformator dapat mengubah tegangan *input AC* yang tinggi menjadi tegangan *output* yang lebih rendah, sehingga dapat memenuhi kebutuhan berbagai perangkat elektronik.



Gambar 3. 23 Step Down LM2596

Tabel 3. 3 Spesifikasi Step Down LM2596

Parameter	Keterangan
<i>Input Voltage</i>	DC 4.5V-40V
<i>Output Voltage</i>	DC 3.3V-37V
<i>Output Current</i>	5A Max
<i>Switching Frequency</i>	110-173KHz

4. IR Sensor FC-51

Sensor *Infrared* (IR) adalah perangkat elektronik yang dirancang untuk mendekksi radiasi inframerah di lingkungan sekitarnya. Setiap objek yang memancarkan panas juga memancarkan radiasi inframerah. Sensor ini terbagi menjadi dua kategori yaitu aktif dan pasif. Sensor IR aktif berfungsi untuk

memancarkan sekaligus mendeteksi radiasi inframerah, sementara sensor IR pasif hanya berfungsi untuk mendeteksi radiasi yang ada di sekitarnya.



Gambar 3. 24 IR Sensor FC-51

Tabel 3. 4 Spesifikasi IR Sensor FC-51

Parameter	Keterangan
<i>Voltage</i>	3-5V
<i>Measuring Range</i>	2-30 cm
<i>Detection Angle</i>	35 °
<i>Current</i>	20mA

5. Limit switch Microswitch KW10-Z1P

Limit switch adalah perangkat elektromekanis yang dirancang untuk mendeteksi posisi akhir atau batas pergerakan suatu objek dalam sistem mekanis. Komponen ini banyak digunakan dalam berbagai aplikasi industri untuk mengirimkan sinyal kepada sistem kontrol, sehingga dapat menghentikan atau mengubah arah pergerakan ketika batas tertentu telah tercapai.



Gambar 3. 25 Limit Switch Microswitch KW10-Z1P

Tabel 3. 5 Spesifikasi *Limit Switch* Microswitch KW10-1P

Parameter	Keterangan
<i>Actuator</i>	Roller Level
<i>Voltage</i>	125V AC
<i>Max Voltage</i>	1.0A
<i>Contact Type</i>	Normally Open (NO), Normally Closed (NC), atau NO+NC
<i>Lifespan</i>	30,000
<i>Operating Temperature</i>	-25°C hingga +65°C

9. *Ultrasonic HC-SR04*

HC-SR04 adalah sensor ultrasonik yang berfungsi untuk mengukur jarak dengan memanfaatkan pantulan gelombang suara frekuensi tinggi. Sensor ini bekerja dengan mengirimkan sinyal melalui pin Trig dan menerima pantulan melalui pin Echo, kemudian menghitung jarak berdasarkan waktu tempuh gelombang tersebut. Sensor ini mampu mengukur jarak dari 2 cm hingga 400 cm dengan tingkat resolusi sekitar 3 mm, sehingga cukup akurat untuk berbagai kebutuhan. Karena menggunakan antarmuka digital sederhana dan tegangan kerja 5V.

**Gambar 3. 26** *Ultrasonic HC-SR04***Tabel 3. 6** Spesifikasi *Ultrasonic HC-SR04*

Parameter	Keterangan
<i>Voltage</i>	3.3V-5V
<i>Current</i>	15mA
<i>Frequency</i>	40Hz
<i>Range</i>	2cm-4m

<i>Measuring angle</i>	15 Degrees
------------------------	------------

6. Motor DC JGY370 *wormgear*

Motor DC JGY370 *wormgear* adalah jenis motor listrik yang dirancang untuk kinerja optimal, terutama dalam aplikasi yang membutuhkan kombinasi kecepatan dan torsi tinggi. Motor ini sering dimanfaatkan dalam berbagai perangkat, seperti peralatan listrik, kendaraan model, serta sistem otomatisasi.



Gambar 3. 27 Motor DC JGY370 *wormgear*

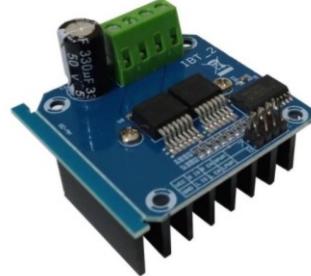
Tabel 3. 7 Spesifikasi Motor DC JGY370 *wormgear*

Parameter	Keterangan
<i>Voltage</i>	6V- 24V
<i>Power Output</i>	1-200W
<i>Rotation Speed</i>	2-210 RPM
<i>Torque</i>	0.12 - 2.45 Nm
<i>Brush Type</i>	Karbon
<i>Motor Size</i>	25 mm x 30 mm

7. Motor Driver BTS7960

Motor Driver BTS7960 adalah modul pengendali motor yang menggunakan IC BTS7960 dengan konfigurasi *H-Bridge*. Modul ini dirancang untuk mengendalikan motor DC dan memiliki kemampuan untuk memutar motor dalam dua arah, yaitu searah jarum jam (CW) dan berlawanan arah jarum jam (CCW). Dengan kemampuan ini, BTS7960 sangat efektif

dalam mengatur pergerakan motor sesuai kebutuhan aplikasi yang diinginkan.



Gambar 3. 28 Motor Driver BTS7960

Tabel 3. 8 Spesifikasi Motor Driver BTS7960

Parameter	Keterangan
<i>Voltage</i>	5.5-27Vdc
<i>Driver Power Supply</i>	5V
PIN	8
<i>Control input</i>	3.3-5V
<i>Peak Current</i>	43-Amp
<i>Board Size</i>	50mm X 50mm X 43mm

8. Baterai Lithium Polymer 48V

Baterai Lithium Polymer (Li-Po) 30000 mAh 48V merupakan baterai isi ulang berkapasitas besar yang dirancang untuk aplikasi yang membutuhkan daya tinggi dan efisiensi energi dalam bentuk yang ringan dan kompak. Dengan tegangan 48V dan kapasitas 3000 Ah, baterai ini mampu menyimpan energi hingga 1.440 Watt-jam, sehingga sangat cocok untuk sistem yang memerlukan pasokan daya stabil dalam durasi yang panjang.



Gambar 3. 29 Baterai Lithium Polymer 48V

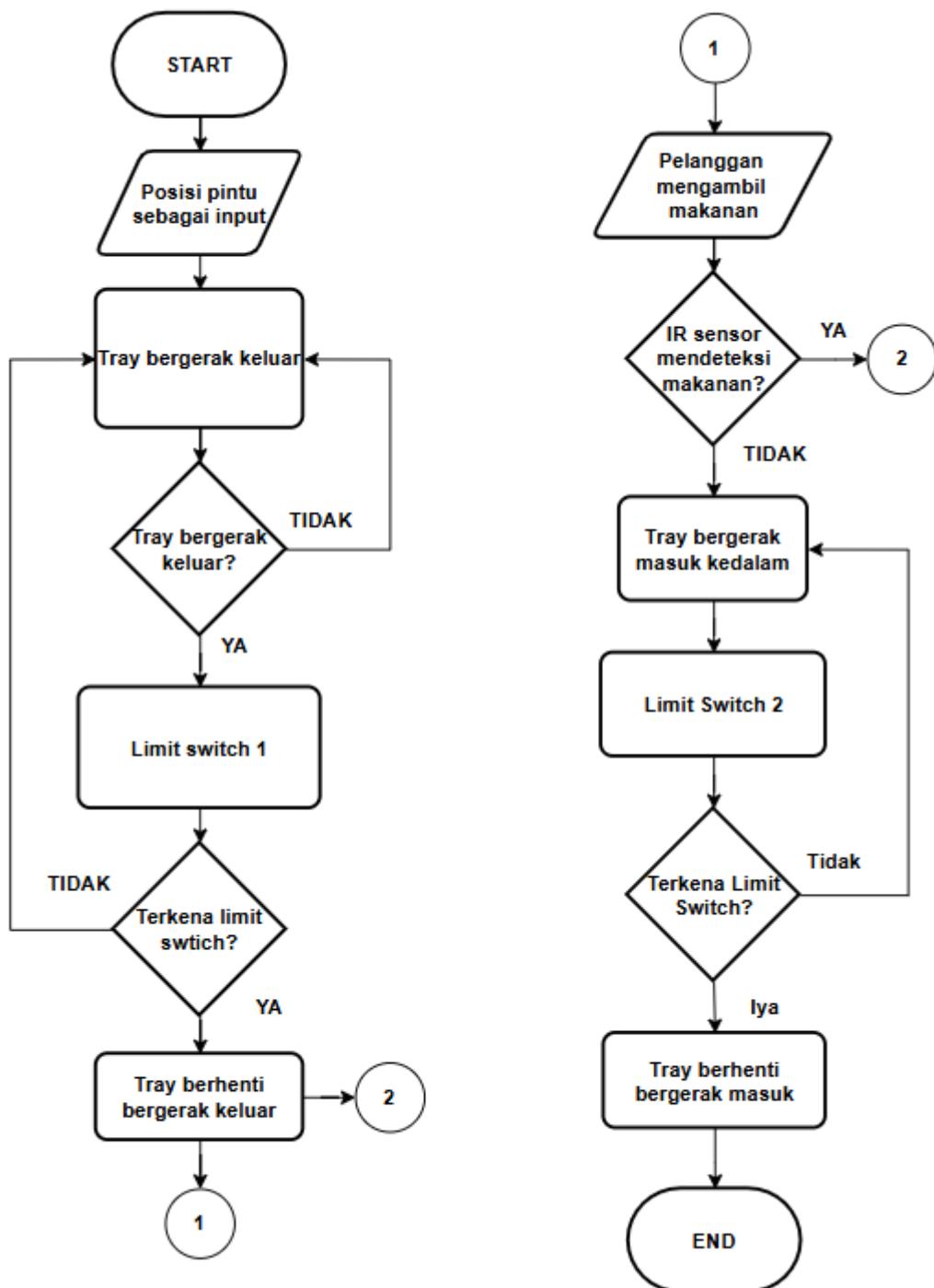
Tabel 3. 9 Spesifikasi Baterai Lithium Polymer 48V

Parameter	Keterangan
<i>Capacity</i>	30000mAh
<i>Max Voltage</i>	48V
<i>Massa</i>	1800 Gram
<i>Dimensi</i>	60 mm x 66 mm x 238 mm

3.8 Desain Kinerja Sistem

3.8.1 Diagram Alir Sistem *Tray*

Berikut merupakan diagram alir terkait dengan sistem mekanisme *tray*



Gambar 3. 30 Diagram alir sistem *tray*

Pada gambar 3.30 diatas dapat dijelaskan diagram alir untuk sistem *tray* yakni posisi pintu digunakan sebagai *input* utama untuk memulai proses. *Tray* akan bergerak keluar hingga mencapai posisi maksimal dan *tray* berhenti ketika mengenai *limit switch*. Pelanggan akan mengambil makanan setelah *tray* berhenti bergerak keluar. Robot dapat mengetahui apakah makanan telah diambil dengan menggunakan IR sensor. Kemudian *tray* akan bergerak masuk kembali ke dalam sampai mengenai limit switch kemudian *tray* akan berhenti bergerak. Pintu akan otomatis tertutup setelah itu. Setelah itu pintu akan tertutup.

BAB IV

HASIL DAN ANALISIS

4.1. Hasil Percobaan

Pada bagian hasil percobaan, sensor dan aktuator diuji atau di kalibrasi untuk dapat mengetahui fungsi dan kinerja dari sensor dan aktuator apakah sesuai dengan *datasheet* atau tidak, sehingga sensor dan aktuator dapat berjalan sesuai dengan spesifikasi yang ada. Hal ini bertujuan agar pada sistem *tray* dan robot dapat berfungsi dan bergerak dengan minimnya *error* yang disebabkan oleh kedua hal tersebut.

Hasil yang didapat pada saat melakukan kalibrasi yakni sensor dan aktuator dapat bekerja dengan baik, namun terdapat sedikit kekurangan pada daya dari aktuator itu sendiri. Faktor seperti tegangan listrik dan perkabelan dapat mempengaruhi hal tersebut. Sehingga dapat diketahui jika komponen tersebut dapat diganti dengan komponen lainnya atau tidak sehingga dapat meningkatkan optimalisasi berjalanannya sistem yang dibuat.

4.1.1 Pengujian IR sensor FC-51

Pengujian sensor IR FC-51 dilakukan dengan menggunakan bantuan alat ukur berupa meteran dengan panjang 60 cm, dan 1 buah benda sebagai objek agar dapat di deteksi oleh IR sensor. Pertama-tama meteran dibentangkan pada permukaan datar lalu diletakkan pada bagian ujung depan sensor IR. Kemudian dilakukan pengujian sebanyak 15 kali dengan kelipatan jarak sebanyak 2 cm, lalu letakkan objek di depan IR sensor dengan jarak 2 cm, 4 cm, 6 cm dan seterusnya. Lalu lihat pada *serial monitor* apakah objek tersebut terdeteksi atau tidak dengan *output* merupakan bilangan biner dengan keterangan “1” untuk terdeteksi dan “0” jika tidak terdeteksi. Ulangin pengujian sebanyak 15 kali. Berikut merupakan tabel dari pengujian sensor IR FC-51:

Tabel 4. 1 Pengujian IR sensor FC-51

Jarak	Keterangan
2 cm	“1” atau Terdeteksi

4 cm	“1” atau Terdeteksi
6 cm	“1” atau Terdeteksi
8 cm	“1” atau Terdeteksi
10 cm	“1” atau Terdeteksi
12 cm	“1” atau Terdeteksi
14 cm	“1” atau Terdeteksi
16 cm	“1” atau Terdeteksi
18 cm	“1” atau Terdeteksi
20 cm	“1” atau Terdeteksi
22 cm	“0” atau Tidak Terdeteksi
24 cm	“0” atau Tidak Terdeteksi
26 cm	“0” atau Tidak Terdeteksi
28 cm	“0” atau Tidak Terdeteksi
30 cm	“0” atau Tidak Terdeteksi

Dari hasil diatas dapat diketahui jika IR sensor FC-51 dapat mendeteksi objek hingga jangkauan 20 cm. Hal ini tidak sesuai dengan *datasheet* yang tertera bahwa IR sensor dapat mendeteksi objek dengan jarak sampai dengan 30 cm, namun untuk penelitian ini jarak yang dibutuhkan sensor IR untuk mendeteksi objek tidak terlalu jauh, sehingga dengan cakupan 10 cm sudah cukup baik untuk digunakan. Namun sensor IR FC-51 harus beroperasi pada kondisi pencahayaan yang sesuai agar pantulan yang dikeluarkan oleh LED dapat diterima dengan baik oleh sensor tersebut.

4.1.2 Pengujian Motor Driver BTS7960

Pada pengujian motor driver dilakukan dengan menguji tegangan dengan cara menyambungkan *motor driver* ke *Power Supply* 12 VDC kemudian menggunakan multimeter sebagai alat ukur tegangan Hal ini bertujuan agar arus yang mengalir ke *motor driver* dapat di lihat dan *motor driver* bekerja dalam batas aman.

Pada pengujian *motor driver* yakni menguji tegangan (V), yang dilakukan adalah cara mengukur tegangan pada motor dengan mengatur PWM yang berbeda dengan cara mengukur arus listrik secara parallel. Pertama-tama adalah nyalakan multimeter dan atur ke mode *DC Voltage* (V). Lalu hubungkan kabel probe merah multimeter ke salah satu *output* atau terminal pada *motor driver*, kemudian hubungkan kabel probe hitam multimeter ke *output* lainnya. (Dalam pengujian ini menggunakan *out 1* dan *out 2*). Cukup tempelkan saja kabel probe tanpa harus memutus kabel yang terhubung. Kemudian nyalakan motor dan jalankan sistem, kemudian catat nilai tegangan yang terbaca pada multimeter, ulangi pengujian sebanyak 10 kali dengan 2 kondisi PWM berbeda. Untuk hasil pengujian terdapat pada tabel berikut:

Tabel 4. 2 Pengujian Motor Driver BTS7960

Pengujian	PWM 255	PWM 191	PWM 128	PWM 64	PWM 0
1	12,1 V	9,9 V	8,9 V	5,6 V	0 V
2	12,2 V	9,9 V	8,9 V	5,6 V	0 V
3	12,1 V	9,9 V	8,8 V	5,5 V	0 V
4	12,0 V	9,8 V	8,9 V	5,6 V	0 V
5	12,1 V	9,9 V	8,8 V	5,5 V	0 V
6	12,0 V	9,8 V	8,8 V	5,5 V	0 V
7	12,0 V	9,9 V	8,8 V	5,5 V	0 V
8	12,0 V	9,9 V	8,8 V	5,5 V	0 V
9	12,1 V	9,9 V	8,8 V	5,6 V	0 V
10	12,2 V	9,8 V	8,8 V	5,5 V	0 V

Dari hasil pengujian tegangan diatas dapat dilihat jika perbedaan nilai tidak terlalu signifikan namun kekurangannya ialah terdapat di *motor driver* yang kurang optimal menyalurkan daya nya sehingga terjadi penurunan tegangan yang mengakibatkan kinerja dari motor kurang optimal.

4.1.3 Pengujian Motor DC JGY370

Pada pengujian motor DC dilakukan dengan menguji arus dengan cara menyambungkan motor DC ke *Power Supply* 12 VDC melalui *motor driver* kemudian menggunakan multimeter sebagai alat ukur arus. Hal ini bertujuan agar arus yang mengalir dari *motor driver* ke motor bekerja dalam batas aman dan tidak kelebihan beban.

Pengujian yang pertama dilakukan adalah mengukur arus pada motor dengan beban maupun tanpa adanya beban dengan cara mengukur arus listrik secara seri dengan multimeter harus menjadi bagian dari rangkaian. Pertama-tama adalah nyalakan multimeter dan atur ke mode *DC Current (A)*. Lalu memutus jalur kabel positif (+) motor yang disambungkan ke *motor driver*, lalu hubungkan kabel probe merah multimeter ke *output* motor yang berada di *motor driver*, sedangkan untuk kabel probe hitam multimeter dihubungkan ke kabel positif (+) motor yang sebelumnya dilepas. Kemudian nyalakan motor dan jalankan sistem, kemudian catat nilai arus yang terbaca pada multimeter, ulangi pengujian sebanyak 10 kali dengan 2 kondisi berbeda yakni dengan beban dan tanpa beban. Beban yang digunakan memiliki berat 2 Kg sesuai dengan kapasitas yang dapat ditampung oleh *tray* itu sendiri. Untuk hasil pengujian terdapat pada tabel berikut:

Tabel 4. 3 Pengujian motor DC JGY370

Pengujian	Tanpa Beban	Dengan Beban
1	0,09 A	0,20 A
2	0,09 A	0,21 A
3	0,09 A	0,22 A
4	0,09 A	0,20 A
5	0,09 A	0,25 A
6	0,09 A	0,21 A
7	0,09 A	0,23 A
8	0,09 A	0,25 A
9	0,09 A	0,20 A
10	0,09 A	0,22 A

Pada pengujian diatas terlihat nilai arus yang di hasilkan tanpa beban stabil dengan rata-rata 0.09 A sedangkan pada kondisi dengan beban arus terlihat naik dan turun namun tidak signifikan dan masih dalam batas kategori yang normal dengan nilai rata-rata 0.22 A.

4.1.4 Pengujian *Limit Switch* KW10-Z1P

Pada pengujian sensor *limit switch* KW10-Z1P dilakukan dengan menggunakan bantuan aktuator motor DC dan *serial monitor*. Pengujian dilakukan sebanyak 10 kali dengan menekan tuas pada sensor *limit switch* untuk melihat keterangan pada *serial monitor* dengan parameter nilai “1” atau “0” dan kondisi dari aktuator motor DC yakni berhenti atau berjalan. Pengujian ini dilakukan agar dapat mengetahui fungsi dari sensor tersebut bekerja secara optimal.

Tabel 4. 4 Pengujian Sensor *Limit Switch* KW10-Z1P

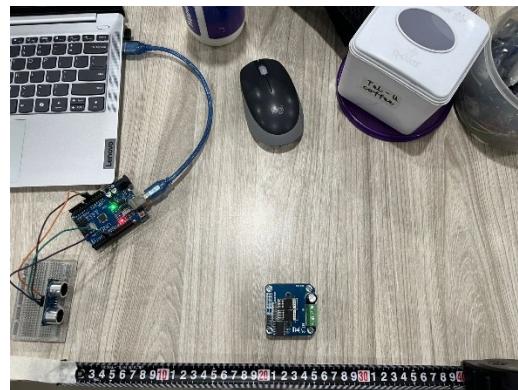
Percobaan	Kondisi Limit Switch	Kondisi Motor DC
1	1	Berhenti
2	0	Berjalan
3	1	Berhenti
4	0	Berjalan
5	1	Berhenti
6	0	Berjalan
7	1	Berhenti
8	0	Berjalan
9	1	Berhenti
10	0	Berjalan

Dapat dilihat pada tabel jika sensor *limit Switch* bekerja dengan baik untuk menghentikan putaran pada motor DC sehingga dapat disimpulkan sensor tersebut bekerja dengan optimal.

4.1.5 Pengujian *Ultrasonic* HC-SR04

Pada Pengujian Sensor *Ultrasonic* HC-SR04 dilakukan dengan menggunakan alat ukur yakni meteran dan objek atau benda sebagai alat bantu

untuk menerima pantulan yang dikeluarkan oleh *Trig* sensor *Ultrasonic* dan akan diterima oleh *Echo* sensor.



Gambar 4. 1 Proses Pengujian sensor *Ultrasonic* HC-SR04

Pada gambar 4.1 terlihat pada pengujian pembacaan sensor *Ultrasonic* HC-SR04 yang dibantu menggunakan alat pengukur berupa meteran sebagai acuan jarak sebenarnya.

Pengukuran dilakukan sebanyak 20 kali dengan menempatkan benda atau objek tepat didepan sensor *Ultrasonic* dengan jarak yang sudah disesuaikan dengan alat ukur yakni setiap kelipatan 2 cm yang dimulai dari 2 cm sampai dengan 40 cm, kemudian lihat hasil dari pengukuran tersebut di *serialmonitor*. Hasil pengukuran dapat dilihat pada tabel dibawah ini:

Tabel 4. 5 Pengujian Sensor *Ultrasonic* HC-SR04

No	Percobaan	Jarak Alat Ukur (cm)	Jarak Hasil Sensor (cm)	Selisih (cm)
1	1	2 cm	2,25 cm	+ 0,25 cm
2	2	4 cm	3,98 cm	- 0,02 cm
3	3	6 cm	5,89 cm	- 0,11 cm
4	4	8 cm	8,09 cm	+ 0,09 cm
5	5	10 cm	10,21 cm	+ 0,21 cm
6	6	12 cm	12,73 cm	+ 0,73 cm
7	7	14 cm	15,76 cm	+ 1,76 cm
8	8	16 cm	16,39 cm	+ 0,39 cm

9	9	18 cm	18,34 cm	+ 0,34 cm
10	10	20 cm	19,26 cm	- 0,74 cm
11	11	22 cm	21,17 cm	- 0,83 cm
12	12	24 cm	23,44 cm	- 0,56 cm
13	13	26 cm	25,86 cm	- 0,14 cm
14	14	28 cm	27,57 cm	- 0,43 cm
15	15	30 cm	29,50 cm	- 0,50 cm
16	16	32 cm	31,98 cm	- 0,02 cm
17	17	34 cm	34,29 cm	+ 0,29 cm
18	18	36 cm	36,79 cm	+ 0,79 cm
19	19	38 cm	37,60 cm	- 0,40 cm
20	20	40 cm	40,24 cm	+ 0,24 cm
TOTAL				0,0425 cm

Dari hasil pengujian diatas didapatkan jika rata-rata selisih dari pembacaan sensor *ultrasonic* HC-SR04 dengan alat ukur berupa meteran ialah 0,0425 cm. Ini menunjukan jika kesalahan setiap pengukuran cenderung relatif kecil (kurang dari 0,1 cm), sehingga sensor masih memiliki fungsi dan tingkat akurasi yang baik dalam melakukan pengukuran jarak.

4.1.6 Pengujian Sistem Kontrol Motor DC Menggunakan PID Berbasis Jarak *Ultrasonic* HC-SR04

Pengujian ini dilakukan untuk mengevaluasi kinerja sistem kendali motor DC yang mengimplementasikan algoritma PID dengan umpan balik dari sensor *ultrasonic* HC-SR04. Tujuan utamanya adalah untuk menilai kemampuan motor dalam berhenti secara otomatis dan presisi pada jarak yang telah ditentukan (8,5 cm dan 27,5 cm), serta mengamati stabilitas dan respon sistem saat mendekati jarak target (*setpoint*). *Setpoint* yang didapat ditentukan dengan mengukur jarak dari ujung sensor *ultrasonic* dengan *bracket* sensor *infrared* yang berada diatas *tray*, kemudian diposisikan pada posisi terdekat dan terjauh dari sensor *ultrasonic*.

Sistem ini memanfaatkan algoritma PID untuk mengatur kecepatan motor berdasarkan selisih antara jarak yang terdeteksi dan nilai *setpoint*. Nilai *setpoint*

akan disesuaikan secara dinamis tergantung pada posisi objek. Untuk menghitung persentase *error*, dapat menggunakan rumus dibawah:

$$\text{Persentase } Error = \frac{[Error]}{Target} \times 100\%$$

Untuk Parameter Kp, Ki, dan Kd yang digunakan adalah:

1. Kp = 3.6
 2. Ki = 0.5
 3. Kd = 3.0
- *Setpoint* 27.5 cm

Tabel 4. 6 Pengujian Sistem kontrol PID jarak 27.5 cm

Percobaan	Target (cm)	Jarak Aktual (cm)	Error (cm)	Waktu (ms)	Persentase Error (%)
1	27,5	26,79	0,71	5392	2,58%
2	27,5	26,40	1,1	6323	4,00%
3	27,5	26,98	0,52	6219	1,89%
4	27,5	26,21	1,29	6219	4,69%
5	27,5	27,27	0,23	6011	0,84%
6	27,5	26,50	1,00	6116	3,64%
7	27,5	26,50	1,00	6220	3,64%
8	27,5	26,98	0,52	6116	1,89%
9	27,5	27,08	0,42	5907	1,53%
10	27,5	26,98	0,52	6219	1,89%
TOTAL					2,66%

Setelah melakukan pengujian sebanyak 10 kali untuk melihat hasil dari kontrol PID dalam mengontrol pergerakan *Tray* untuk jarak *Setpoint* 27,5 cm dengan menggunakan sensor *ultrasonic* HC-SR04, dapat diketahui bahwa sistem mampu mencapai akurasi yang sangat baik. Berdasarkan data, rata-rata *error* yang diperoleh adalah $\pm 0,73$ cm, dengan rata-rata persentase *error* sebesar 2,66%. *Error* yang tercatat berada dalam rentang 1,29 cm hingga 0,29 cm, dengan persentase

error antara 0,84% hingga 4,69%. Sistem mampu menstabilkan *tray* dalam waktu sekitar 5,39 hingga 6,32 detik.

- *Setpoint* 8,5 cm

Tabel 4. 7 Pengujian Sistem kontrol PID jarak 8,5 cm

Percobaan	Target (cm)	Jarak Aktual (cm)	Error (cm)	Waktu (ms)	Persentase <i>Error (%)</i>
1	8,5	8,13	0,37	5913	4,35%
2	8,5	8,02	0,48	5500	5,65%
3	8,5	8,45	0,05	5500	0,59%
4	8,5	8,13	0,37	5705	4,35%
5	8,5	8,02	0,48	5395	5,65%
6	8,5	8,35	0,15	5500	1,76%
7	8,5	8,45	0,05	5395	0,59%
8	8,5	8,35	0,15	5605	1,76%
9	8,5	8,35	0,15	5395	1,76%
10	8,5	8,02	0,48	5605	5,65%
TOTAL					3,21%

Setelah melakukan pengujian sebanyak 10 kali pada *setpoint* 8,5 cm menunjukkan bahwa sistem *tray* otomatis memiliki akurasi yang akurat dan stabil. *Error* yang tercatat berada dalam rentang 0,05 hingga 0,48 cm, dengan rata-rata sebesar 0,27 cm atau sekitar 3,21%. *Tray* umumnya mencapai kondisi stabil dalam waktu 5,39 hingga 5,91 detik, menandakan sistem cukup responsif dalam mencapai posisi yang diinginkan. Secara keseluruhan, hasil ini menunjukkan bahwa kontrol PID yang digunakan mampu menjaga akurasi posisi *tray* dan memastikan gerakan yang stabil.

4.1.7 Pengujian Sistem Jarak Jauh (*Internet Of Things*)

Pengujian ini dilakukan untuk menilai performa sistem pemantauan jarak jauh berbasis *Internet of Things* (IoT) yang telah dikembangkan. Fokus utamanya adalah mengukur *delay* antara proses pengiriman data oleh Arduino Mega ke ESP32 dan penerimaan serta penyimpanannya di Firebase melalui jaringan internet. Nilai *delay* dihitung berdasarkan selisih waktu antara pencatatan data pada *serial monitor* sebagai pihak pengirim (*transmitter*) dan waktu data tersebut tersimpan di *database* Firebase sebagai pihak penerima (*receiver*).

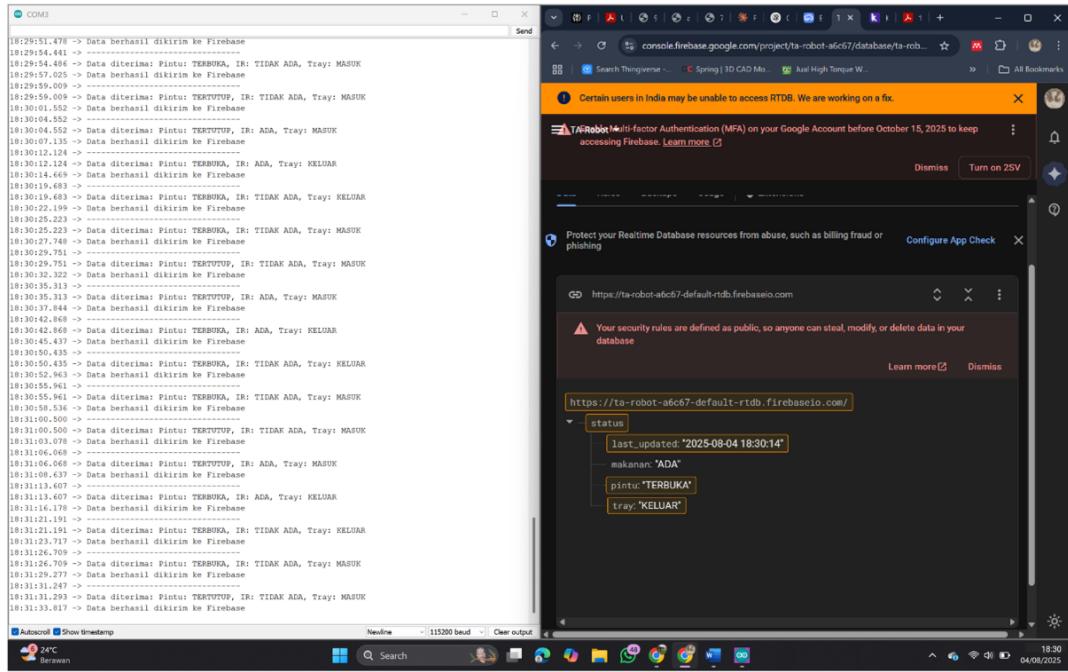
Pada Tabel dibawah ini menyajikan hasil dari pengujian sistem pada tray yang berawalan saat ESP32 mengirimkan data jika pintu sudah terbuka, kemudian melakukan pencatatan terhadap *delay* yang terjadi saat proses *tray* terbuka dan pada database.

Tabel 4. 8 Pengujian Delay sistem Inter Of Things

No	Status Pintu	Status IR	Status Tray	Waktu di serial monitor	Waktu di database	Delay (ms)
1	Tertutup	Ada	Masuk	18:29:59	18:30:01	0,2
2	Tertutup	Ada	Masuk	18:30:04	18:30:07	0,3
3	Terbuka	Ada	Keluar	18:30:12	18:30:14	0,2
4	Terbuka	Tidak Ada	Keluar	18:30:19	18:30:22	0,3
5	Terbuka	Tidak Ada	Masuk	18:30:25	18:30:27	0,2
6	Tertutup	Tidak Ada	Masuk	18:30:29	18:30:32	0,3
7	Tertutup	Ada	Masuk	18:35:06	18:35:08	0,2
8	Tertutup	Ada	Masuk	18:35:11	18:35:14	0,3
9	Terbuka	Ada	Keluar	18:35:19	18:35:21	0,2
10	Terbuka	Tidak Ada	Keluar	18:35:25	18:35:27	0,2
11	Terbuka	Tidak Ada	Masuk	18:35:32	18:35:35	0,3
12	Tertutup	Tidak Ada	Masuk	18:35:40	18:35:43	0,3
Rata-rata						0,25 ms

Pada tabel 4.8 pengujian *delay* diatas terlihat jika pada saat pengiriman data kepada *database* mengalami rata-rata *delay* 0,25 ms ini membuktikan jika *delay*

yang muncul relatif kecil dan cenderung responsif karena dibawah 1 detik dan untuk melakukan *monitoring* dapat langsung dipantau dengan baik.



Gambar 4. 2 Hasil *monitoring* melalui *database*

Pada gambar 4.2 terlihat hasil pengiriman data dari kondisi pintu, *tray*, dan keadaan makanan yang terdapat pada *database* firebase dalam keadaan terupdate sesuai dengan pengiriman data yang ditampilkan pada *serialmonitor*.

4.2. Pengujian Parameter PID

Pengujian dilakukan secara bertahap unutk mengevaluasi pengaruh dari masing-masing parameter PID pada sensor *ultrasonic* HC-SR04 untuk mengontrol pergerakan *tray* dalam mengatur jarak pemberhentian dari *tray*.

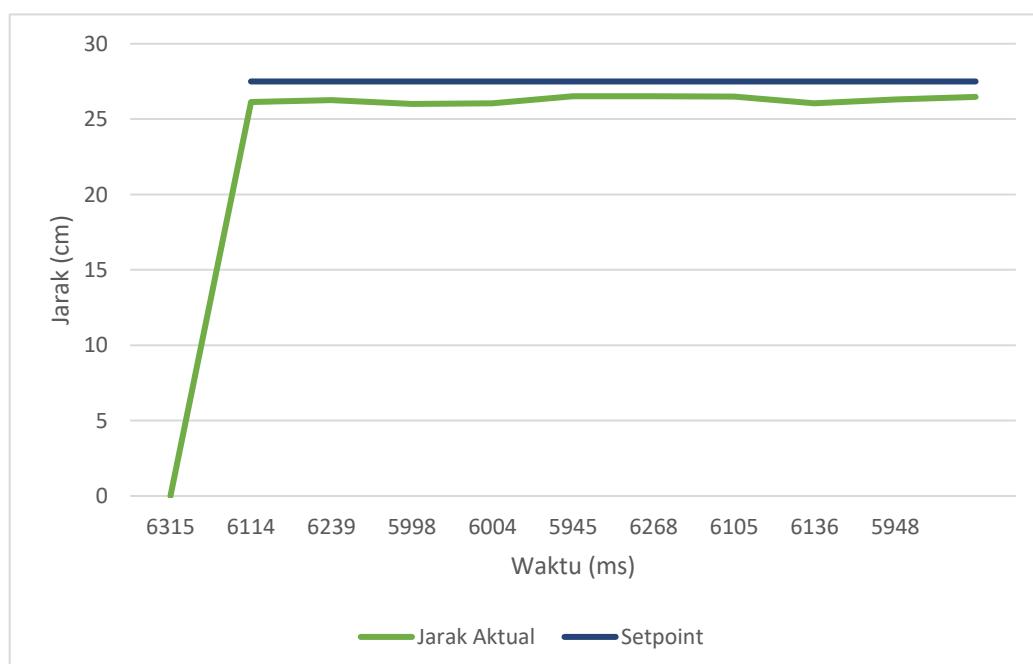
4.2.1 Pengujian Menggunakan Parameter Kp=3.6, Ki=0, Kd=0

Konfigurasi ini hanya menggunakan parameter Kp atau kontrol proporsional untuk mengontrol pergerakan *Motor DC*. Semakin jauh *error* dari *setpoint*, semakin besar *output PWM* ke *Motor DC*.

- Setpoint 27.5

Tabel 4. 9 Pengujian Parameter Kp pada Setpoint 27.5 cm

Percobaan	Jarak Aktual (cm)	Error (cm)	Waktu (ms)	Persentase error (%)
1	26,13	1,37	6315	4,98 %
2	26,26	1,24	6114	4,51 %
3	26,01	1,49	6239	5,42 %
4	26,06	1,44	5998	5,24 %
5	26,52	0,98	6004	3,56 %
6	26,52	0,98	5945	3,56 %
7	26,49	1,01	6268	3,67 %
8	26,06	1,44	6105	5,24 %
9	26,3	1,20	6136	4,36 %
10	26,48	1,02	5948	3,71 %
TOTAL				4,42 %



Gambar 4. 3 Grafik Respon Transien PID Kp setpoint 27,5 cm

- Setpoint 8.5

Tabel 4. 10 Pengujian Parameter Kp pada Setpoint 8.5 cm

Percobaan	Jarak Aktual (cm)	Error (cm)	Waktu (ms)	Persentase error (%)
1	7,96	0,54	5682	6,35 %
2	7,95	0,55	5867	6,47 %
3	8,1	0,40	5645	4,71 %
4	8,15	0,35	5375	4,12 %
5	7,9	0,60	5598	7,06 %
6	8,17	0,33	5811	3,88 %
7	8,01	0,49	5486	5,76 %
8	7,92	0,58	5422	6,82 %
9	8,09	0,41	5346	4,82 %
10	8,18	0,32	5475	3,76 %
TOTAL				5,38 %



Gambar 4. 4 Grafik Respon Transien PID Kp setpoint 8,5 cm

Pada pengujian ini, sistem hanya menggunakan kontrol proporsional (Kp). Berdasarkan data, rata-rata persentase *error* pada *setpoint* 27,5 cm adalah 4,42%, sedangkan pada *setpoint* 8,5 cm sebesar 5,38%. Nilai *error* yang relatif lebih tinggi ini menunjukkan adanya *steady-state error*, yang umum terjadi ketika hanya menggunakan kontrol proporsional tanpa menggunakan kontrol integral. *Tray* masih mampu mendekati *setpoint*, namun belum sepenuhnya akurat, terutama pada jarak pendek (8,5 cm) di mana *tray* lebih sensitif terhadap *overshoot* dan *undershoot*.

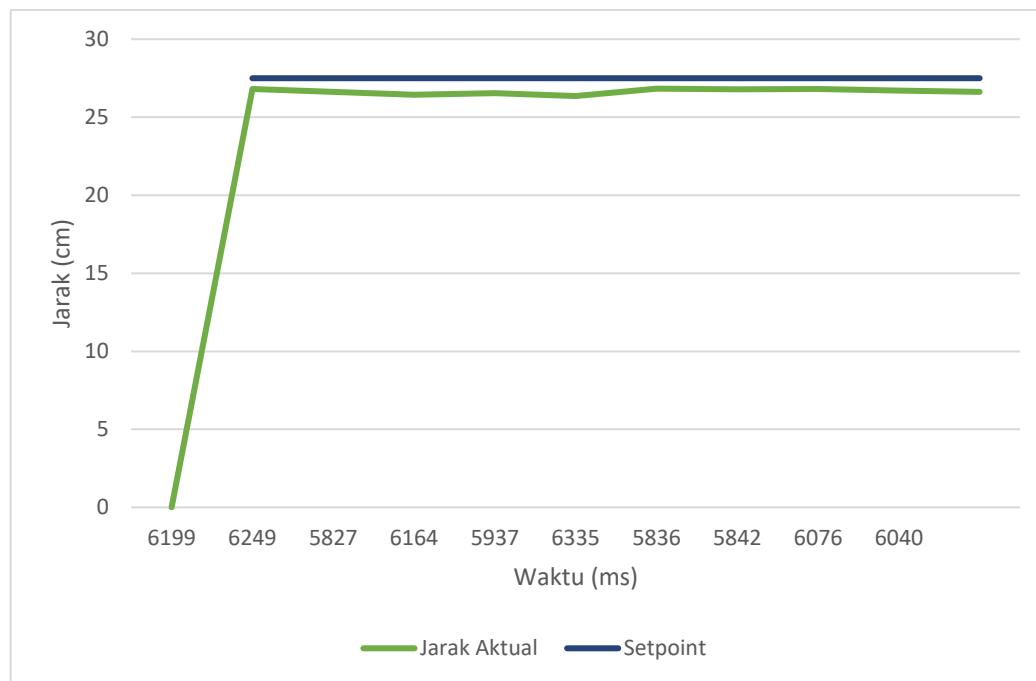
4.2.2 Pengujian Menggunakan Parameter Kp=3.6, Ki=0, Kd=3.0

Konfigurasi ini menggunakan parameter Kp dan Kd atau kontrol proporsional dan kontrol derivatif untuk membantu memprediksi arah dan perubahan *error* serta meningkatkan stabilitas sistem.

- *Setpoint* 27,5

Tabel 4. 11 Pengujian Parameter Kp dan Kd pada *Setpoint* 27,5 cm

Percobaan	Jarak Aktual (cm)	Error (cm)	Waktu (ms)	Persentase <i>error</i> (%)
1	26,81	0,69	6199	2,51 %
2	26,63	0,87	6249	3,16 %
3	26,44	1,06	5827	3,85 %
4	26,55	0,95	6164	3,45 %
5	26,36	1,14	5937	4,15 %
6	26,83	0,67	6335	2,44 %
7	26,78	0,72	5836	2,62 %
8	26,81	0,69	5842	2,51 %
9	26,70	0,80	6076	2,91 %
10	26,62	0,88	6040	3,20 %
TOTAL				3,08%

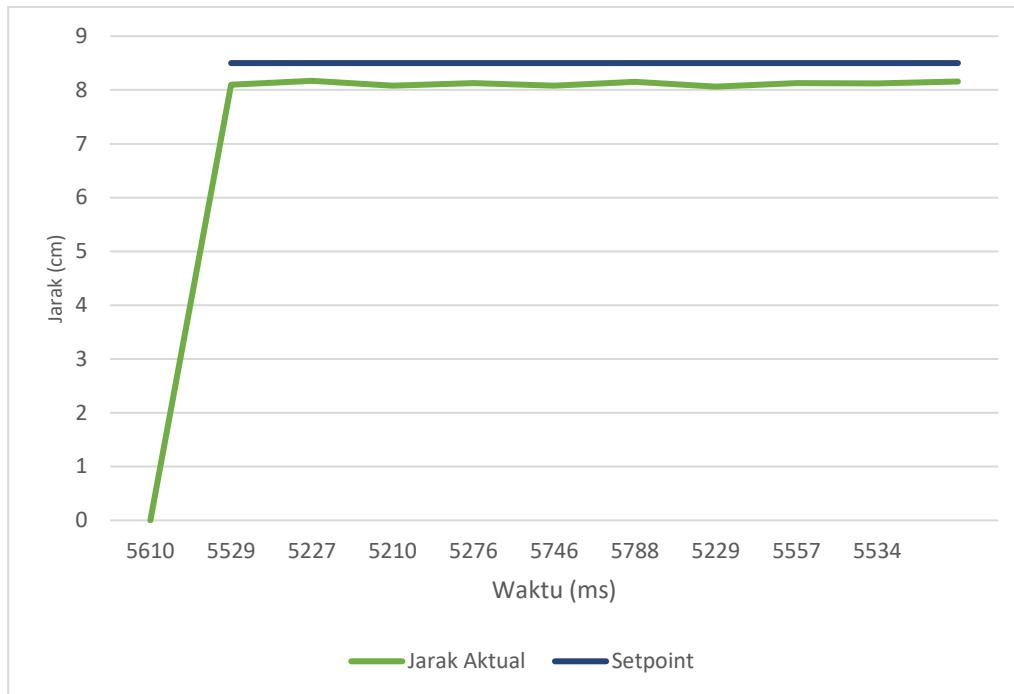


Gambar 4. 5 Grafik Respon Transien PID Kp dan Kd *setpoint* 27,5 cm

- *Setpoint* 8,5

Tabel 4. 12 Pengujian Parameter Kp dan Kd pada *Setpoint* 8,5 cm

Percobaan	Jarak Aktual (cm)	Error (cm)	Waktu (ms)	Persentase error (%)
1	8,10	0,40	5610	4,71 %
2	8,17	0,33	5529	3,88 %
3	8,08	0,42	5227	4,94 %
4	8,13	0,37	5210	4,35 %
5	8,08	0,42	5276	4,94 %
6	8,15	0,35	5746	4,12 %
7	8,06	0,44	5788	5,18 %
8	8,13	0,37	5229	4,35 %
9	8,12	0,38	5557	4,47 %
10	8,16	0,34	5534	4,00 %
TOTAL				4,49%



Gambar 4. 6 Grafik Respon Transien PID Kp dan Kd setpoint 8,5 cm

Pada pengujian ini, sistem dilengkapi dengan kontrol derivatif (Kd), yang bertujuan untuk meningkatkan stabilitas. Hasil pengujian menunjukkan penurunan rata-rata persentase *error* pada *setpoint* 27.5 cm menjadi 3.08%, sedangkan pada *setpoint* 8.5 cm turun menjadi 4.49%. Penurunan *error* ini membuktikan bahwa penambahan kontrol derivatif membantu meningkatkan stabilitas sistem. Penambahan Kd membantu meredam osilasi dan mempercepat respon dengan mendeteksi perubahan *error* lebih awal, sehingga *tray* mampu berhenti lebih presisi mendekati target.

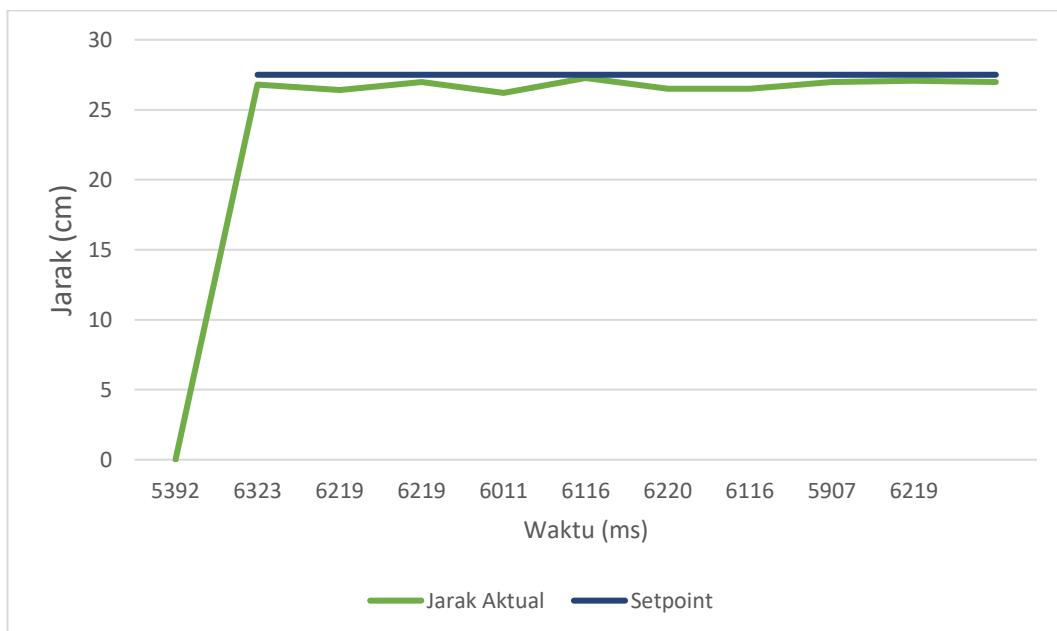
4.2.3 Pengujian Menggunakan Parameter Kp=3.6, Ki=0.5, Kd=3.0

Konfigurasi ini menggunakan semua parameter yakni Kp, Ki, dan Kd untuk membantu memprediksi perubahan *error* yang terjadi dan menahan putaran motor DC saat mengalami *overshoot*.

- Setpoint 27,5

Tabel 4. 13 Pengujian Semua Parameter pada Setpoint 27,5 cm

Percobaan	Jarak Aktual (cm)	Error (cm)	Waktu (ms)	Persentase error (%)
1	26,79	0,71	5392	2,58 %
2	26,40	1,10	6323	4,00 %
3	26,98	0,52	6219	1,89 %
4	26,21	1,29	6219	4,69 %
5	27,27	0,23	6011	0,84 %
6	26,50	1,00	6116	3,64 %
7	26,50	1,00	6220	3,64 %
8	26,98	0,52	6116	1,89 %
9	27,08	0,42	5907	1,53 %
10	26,98	0,52	6219	1,89 %
TOTAL				2,66 %

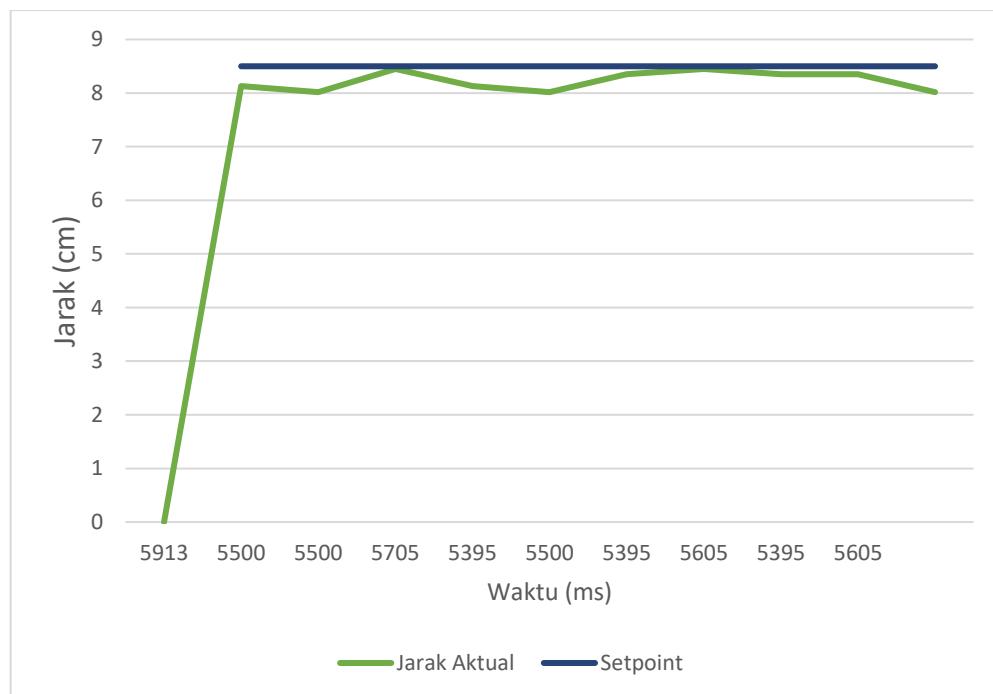


Gambar 4. 7 Grafik Respon Transien PID Lengkap setpoint 27,5 cm

- *Setpoint 8.5*

Tabel 4. 14 Pengujian Semua Parameter pada *Setpoint* 8,5 cm

Percobaan	Jarak Aktual (cm)	Error (cm)	Waktu (ms)	Persentase error (%)
1	8,13	0,37	5913	4,35 %
2	8,02	0,48	5500	5,65 %
3	8,45	0,05	5500	0,59 %
4	8,13	0,37	5705	4,35 %
5	8,02	0,48	5395	5,65 %
6	8,35	0,15	5500	1,76 %
7	8,45	0,05	5395	0,59 %
8	8,35	0,15	5605	1,76 %
9	8,35	0,15	5395	1,76 %
10	8,02	0,48	5605	5,65 %
TOTAL				3,21 %



Gambar 4. 8 Grafik Respon Transien PID Lengkap *setpoint* 8,5 cm

Pada percobaan ini, sistem sudah menerapkan kontrol PID penuh dengan penambahan parameter integral (Ki). Rata-rata persentase *error* yang dihasilkan adalah 2,66% pada setpoint 27.5 cm dan 3.21% pada *setpoint* 8.5 cm. Hasil ini menunjukkan performa yang sangat baik di antara ketiga parameter, di mana *error* dapat dikurangi sedikit mungkin. Penambahan Kd membantu mempercepat respon sistem serta meredam osilasi atau *overshoot* yang sering muncul pada sistem tanpa derivatif. *Error steady-state* menjadi lebih kecil dan *tray* mampu mencapai *setpoint* dengan ketepatan yang sangat tinggi. Secara keseluruhan, dengan menggunakan kontrol PID penuh terbukti efektif dalam menjaga akurasi dan kestabilan pergerakan *tray*, baik pada jarak panjang maupun jarak pendek.

4.3. Analisis

Bagian analisis ini disusun untuk menjelaskan keterkaitan antara data dan hasil temuan yang diperoleh selama seluruh rangkaian pengujian dan validasi. Melalui analisis terhadap hasil yang diperoleh, dilakukan evaluasi terhadap tingkat pencapaian masing-masing tujuan penelitian, identifikasi faktor-faktor yang memengaruhinya, serta implementasi dari kinerja sistem berdasarkan hasil uji yang telah dilakukan.

4.3.1 Analisis Performa Sensor IR FC-51 Dalam Deteksi Makanan

Sensor infrared FC-51 dimanfaatkan untuk mendeteksi ada atau tidaknya makanan di atas *tray*. Berdasarkan hasil pengujian, sensor ini mampu menghasilkan sinyal digital “1” hingga jarak sekitar 20 cm, walaupun pada *datasheet* tercantum jangkauan hingga 30 cm. Namun, dalam konteks sistem ini, jangkauan tersebut sudah memadai, karena objek yang perlu dideteksi umumnya berada dalam jarak 5–10 cm dari sensor, yaitu langsung di atas *tray*.

Salah satu aspek penting dari performa sensor ini adalah kestabilan sinyal *output*-nya. Dari hasil pengujian, sinyal digital yang dihasilkan cukup konsisten, sehingga sistem bisa menentukan dengan tepat kapan *tray* seharusnya kembali masuk. Selama sensor masih mendeteksi keberadaan makanan, *tray* akan tetap berada di luar. Namun jika semua sensor IR menunjukkan bahwa *tray* kosong, sistem secara otomatis menggerakkan *tray* untuk masuk kembali.

Kemampuan deteksi yang andal ini sangat penting dalam mendukung proses penyajian makanan yang bersifat otomatis, higienis, dan efisien. Dengan kata lain, sensor IR berperan penting dalam pengambilan keputusan sistem *tray* tanpa perlu adanya intervensi langsung dari pengguna.

4.3.2 Analisis Fungsi *Limit Switch* KW10-Z1P sebagai Pengaman dan Pengontrol Posisi

Limit switch dipasang di ujung lintasan *tray* sebagai penanda batas gerak mekanis. Berdasarkan hasil pengujian, *switch* ini mampu memberikan respon yang cukup baik. Ketika ditekan, motor akan berhenti dan saat dilepas, motor kembali bergerak. Pola kerja ini memastikan bahwa motor tidak melampaui batas gerak yang telah dirancang, sehingga setiap pergerakan tetap terkendali.

Fungsi utama *Limit switch* adalah untuk mencegah pergerakan berlebih yang berpotensi merusak bagian mekanik sistem. Selain itu, keberadaannya juga berperan dalam menyinkronkan tahapan kerja otomatis. *Tray* hanya akan keluar jika pintu benar-benar terbuka, dan pintu hanya akan menutup setelah *tray* kembali ke posisi semula.

Dengan memiliki dua fungsi sebagai penentu batas dan pengatur urutan gerak, *limit switch* menjadi komponen yang penting dalam menjaga keamanan sistem serta memastikan seluruh proses berjalan secara baik, presisi, dan sesuai urutan yang diharapkan.

4.3.3 Analisis Integrasi Sistem *Monitoring* dan Kontrol IoT

Sistem ini merupakan hasil integrasi dari beberapa mikrokontroler dan sensor yang saling terhubung dan berkomunikasi satu sama lain. ESP32 bertugas mengirim status sistem ke Raspberry PI, Arduino Mega mengontrol *tray* dan mengirim data informasi ke EPS32, Raspberry Pi sebagai otak dari navigasi robot dan mengirimkan data ke database yakni Firebase, sementara Kodular digunakan sebagai UI *monitoring* berbasis *smartphone*. Seluruh komponen ini bekerja secara bersamaan dan *real-time*.

Dengan dukungan Firebase, pengguna bisa memantau kondisi pintu, *tray*, dan makanan secara langsung dari *smartphone*. Selain itu, sistem juga bisa

merespon perintah yang dikirim dari Raspberry Pi untuk menjalankan fungsi secara otomatis, tanpa perlu kontrol manual.

Keberhasilan dalam menggabungkan semua perangkat ini membuktikan bahwa sistem IoT yang dibangun tidak hanya mampu menjalankan kontrol otomatis, tetapi juga menawarkan fitur pemantauan yang responsif, fleksibel dan *real-time*. Dari pengujian *delay* terlihat jika pada saat pengiriman data kepada *database* mengalami rata-rata *delay* 0,25 ms ini membuktikan jika *delay* yang muncul relatif kecil dan cenderung responsif karena dibawah 1 detik dan untuk melakukan *monitoring* dapat langsung dipantau dengan baik. Hal ini tentunya sangat meningkatkan efisiensi dan kenyamanan dalam penggunaan robot pengantar makanan.

4.3.4 Analisis Penggunaan Kontrol PID Pada Pergerakan *Tray*

Berdasarkan hasil pengujian dan analisis terhadap tiga konfigurasi kontrol PID, dapat disimpulkan bahwa penggunaan ketiga parameter PID sekaligus dengan nilai $K_p = 3.6$, $K_i = 0.5$, dan $K_d = 3.0$ merupakan konfigurasi paling optimal dan memberikan performa terbaik. Kombinasi ini menghasilkan rata-rata persentase *error* terendah, yaitu 2.66% pada setpoint 27.5 cm dan 3.21% pada setpoint 8.5 cm.

Penambahan kontrol derivatif (K_d) pada pengujian kedua terbukti mampu meningkatkan kecepatan respon sistem dan meredam osilasi (overshoot) dibandingkan dengan konfigurasi pertama yang hanya menggunakan kontrol proporsional. Dengan adanya K_d , sistem menjadi lebih stabil dalam pergerakan *tray*, meskipun masih terdapat sedikit *error steady-state* karena tidak ada komponen integral (K_i) yang mengoreksi *error* akumulatif.

Di sisi lain, penerapan kontrol PID yang lengkap dengan ketiga parameter (K_p , K_i , K_d) pada pengujian ketiga menunjukkan peningkatan yang signifikan dalam hal stabilitas, akurasi, dan ketepatan pergerakan *tray*. Konfigurasi ini tidak hanya berhasil menurunkan *error steady-state*, tetapi juga mempercepat respon sistem serta meningkatkan kinerja sistem secara menyeluruh.

Dengan tingkat *error* yang lebih rendah dan kestabilan sistem yang lebih tinggi, konfigurasi lengkap PID ini terbukti paling efektif dalam menjaga presisi

posisi *tray*, baik pada jarak pendek maupun panjang. Hasil ini menandakan pentingnya penerapan kontrol PID secara penuh untuk mencapai performa sistem yang maksimal.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini berhasil merancang dan mengimplementasikan sistem *tray* otomatis pada robot pengantar makanan dengan memanfaatkan motor DC yang dikontrol menggunakan algoritma PID serta sensor ultrasonik HC-SR04 sebagai sensor jarak. Hasil pengujian menunjukkan bahwa pada kontrol proporsional (K_p) saja, *tray* masih mengalami *error steady-state* yang cukup signifikan, yaitu 4,42% pada jarak 27,5 cm dan 5,38% pada jarak 8,5 cm. Penambahan kontrol derivatif (K_d) kemudian meningkatkan respon sistem dan meredam osilasi, meskipun *error* akumulatif belum sepenuhnya terkoreksi. Performa terbaik diperoleh ketika parameter PID lengkap diterapkan ($K_p = 3.6$, $K_i = 0.5$, $K_d = 3.0$), dengan rata-rata *error* hanya 2,66% pada jarak 27,5 cm dan 3,21% pada jarak 8,5 cm. Hal ini membuktikan bahwa sistem memiliki tingkat akurasi tinggi, pergerakan *tray* yang lebih stabil, serta minim osilasi dan *overshoot*.

Selain akurasi jarak, keberhasilan sistem juga didukung oleh limit switch sebagai pengaman batas gerak *tray* dan sensor IR yang efektif mendeteksi keberadaan makanan pada jarak yang sudah di uji yakni maksimal 20 cm dengan kondisi pencahayaan yang baik. Integrasi sistem *monitoring IoT* menggunakan Kodular dan Firebase berjalan dengan baik, memungkinkan pemantauan status *tray* melalui *smartphone* secara *real-time* dengan rata-rata *delay* 0,25 ms yang cenderung kecil karena dibawah 1 detik.

Secara keseluruhan, sistem ini mampu mendukung pelayanan otomatis di bidang pengantaran makanan, dan mampu menghidangkan makanan yang aman kepada pelanggan, mengurangi ketergantungan tenaga manusia, meningkatkan higienitas, serta meningkatkan efisiensi proses pelayanan penyerahan makanan kepada pelanggan. Hasil pengujian menunjukkan sistem memiliki potensi besar untuk dikembangkan lebih lanjut sesuai kebutuhan lapangan.

5.2 Saran

1. Penggunaan sensor yang lebih presisi dan adaptif

Meskipun sensor IR FC-51 telah bekerja dengan baik, untuk depannya disarankan untuk menggunakan sensor jarak yang lebih baik atau sensor optik lain yang lebih stabil terhadap variasi pencahayaan dan warna objek. Hal ini bertujuan untuk meningkatkan keakuratan sistem deteksi makanan dalam berbagai kondisi lingkungan.

2. Penambahan fitur pengaman

Sistem saat ini belum dilengkapi dengan proteksi aktif terhadap kegagalan. Disarankan untuk menambahkan sensor suhu untuk memantau kondisi *real-time* perangkat, sehingga sistem dapat dipantau dan dapat langsung mematikan sistem secara otomatis saat terdeteksi kondisi yang tidak normal.

3. Peningkatan *User Interface* dan kontrol jarak jauh

User Interface Kodular saat ini hanya berfungsi untuk *monitoring* status. Untuk pengembangan berikutnya, akan lebih baik jika ditambahkan opsi fitur kontrol manual jarak jauh. Ini akan memberi fitur tambahan bagi operator atau pemilik robot.

4. Desain produk yang mendukung perawatan (maintenance) yang mudah

Untuk penggunaan jangka panjang, sangat penting untuk memastikan bahwa komponen seperti *tray*, motor, sensor, dan *wiring* dapat diakses dan diganti dengan mudah jika terjadi kerusakan. Oleh karena itu, disarankan untuk merancang struktur sistem dengan fleksibilitas yang baik sehingga proses *troubleshooting* dan perawatan tidak menyulitkan pengguna awam.

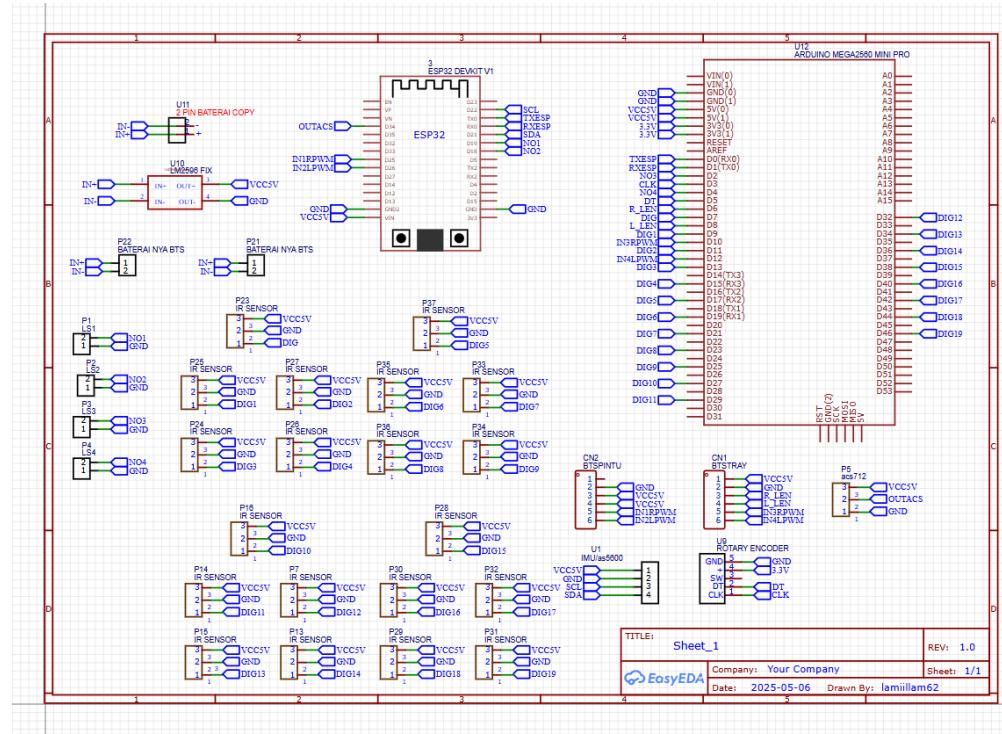
DAFTAR PUSTAKA

- [1] R. Thaker, “Service Robots in Hospitality and Tourism: Adoption, Challenges, and Long-Term Interaction Article in,” *International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences*, 2021, doi: 10.5281/zenodo.14001439.
- [2] L. Khakim, Sunarno, Sugiyanto, S. Siahaan, J. Simanjuntak, and D. Lumbantoruan, “‘Modern Control Systems’ by Richard C. Dorf and Robert H. Bishop - Пошук Google,” *Jurnal MIPA Unnes*, vol. 35, no. 2, pp. 38–49, 2022.
- [3] C. Spencer, “Restaurant hires \$1000-a-month robot waiter, leading to tip surge – The Hill.” Accessed: Dec. 22, 2024. [Online]. Available: <https://thehill.com/changing-america/resilience/smart-cities/577452-restaurant-hires-1000-a-month-robot-waiter-and-tips/>
- [4] C. Hu, “How robots are actually helping in restaurants | Popular Science.” Accessed: Dec. 22, 2024. [Online]. Available: <https://www.popsci.com/technology/robots-in-restaurants/>
- [5] F. T. Elektro and U. Telkom, *BUKU TUGAS AKHIR CAPSTONE DESIGN Mobilisasi Robot Pengantar Makanan Berbasis Odometry dan QR Detection*. 2023.
- [6] S. P. V, “Introduction To Robotics.,” *Springs*, vol. 4, no. 3, 2021, doi: 10.48175/ijarsct-v4-i3-007.
- [7] E. S. Eriana and Drs. A. Zein, “Artificial Intelligence,” *Angewandte Chemie International Edition*, vol. 6(11), p. 1, 2023.
- [8] HumasTGD, “STMIK TRIGUNA DHARMA - Ayo Mengenal Apa itu Robotik di Kampus Robotik STMIK Triguna Dharma.” Accessed: Dec. 22, 2024. [Online]. Available: <https://fresh.trigunadharma.ac.id/detail/ayomengenal-apitu-robotik-di-kampus-robotik-stmik-triguna-dharma>
- [9] E. A. Prastyo, “Mengenal Detail tentang Robotika.” Accessed: Dec. 22, 2024. [Online]. Available: <https://www.arduino.biz.id/2023/02/mengenal-detail-tentang-robotika.html>
- [10] A. Malewar, “PuduTech introduced BellaBot, a cute meowing food delivery robot.” Accessed: Dec. 22, 2024. [Online]. Available: <https://www.inceptivemind.com/pudutech-bellabot-holabot-cute-meowing-food-delivery-robot/11193/>

- [11] “Apa itu Internet of Things? Pengertian, Cara Kerja, dan Contohnya.” Accessed: Aug. 19, 2025. [Online]. Available: <https://docif.telkomuniversity.ac.id/apa-itu-iot/>
- [12] A. Dubey and S. K. Yadav, “Basics of Internet of Things,” pp. 1–5, 2024, doi: 10.55041/IJSREM37970.
- [13] Y. A. Furima, J. P. P. Naibaho, and C. D. Suhendra, “Aplikasi Belajar Dan Bermain Untuk Anak Usia Dini Menggunakan Kodular,” *JISTECH: Journal of Information Science and Technology*, vol. 11, no. 1, pp. 47–58, 2023, doi: 10.30862/jistech.v11i1.63.
- [14] N. Z.A, Y. P. Roja, and N. Sylvia, “Aplikasi Kontrol PID pada Reaktor Pabrik Asam Formiat dengan Kapasitas 100.000 Ton/Tahun,” *Jurnal Teknologi Kimia Unimal*, vol. 7, no. 2, p. 135, 2019, doi: 10.29103/jtku.v7i2.1253.
- [15] Z. F. Emzain and I. Mashudi, “Konsep Dasar Konsep Dasar Konsep Dasar,” vol. 14090, no. 1989, 2018, [Online]. Available: <https://adoc.pub/queue/bab-ii-konsep-dasar-langsung-schult-videbeck-1998-langsung-d.html>
- [16] “Pengontrol PID dan Penjelasan Teorinya - NI.” Accessed: Dec. 22, 2024. [Online]. Available: <https://www.ni.com/en/shop/labview/pid-theory-explained.html?srsltid=AfmBOoq0TNanKKQk7NxDUJEhPjmN98oWpGJDjpPDQNZdlkerZgzpNKma>
- [17] W. H. ITB, “Digital PID Controller with Arduino | by Workshop HME ITB | Medium.” Accessed: Jan. 04, 2025. [Online]. Available: <https://medium.com/@workshopitb/digital-pid-controller-with-arduino-9f5d7a06acc7>
- [18] S. Zhuge, “PID ‘Proportional, Integral, and Derivative’ Control Theory.” Accessed: Dec. 22, 2024. [Online]. Available: <https://www.crystalinstruments.com/blog/2020/8/23/pid-control-theory>
- [19] P. Saka Gilap Asa and S. Priyambodo, “Sistem Pembelajaran Kontrol Pid (Proporsional Integral Derivatif) Pada Pengatur Kecepatan Motor Dc Pid(Proportional Integral Derivative) Control Learning System on Dc Motor Speed Controller,” *Jurnal Elektrikal*, vol. 3, no. 1, pp. 72–77, 2016.

LAMPIRAN

1. Pinout PCB



2. Robot Tampak Depan dan Belakang





3. Source Code kontrol tray

Screenshot of a GitHub repository page showing the source code for the food delivery robot's control system.

The repository is named "Tugas-Akhir---Robot-Pengantar-Makanan". The file being viewed is "kode_arduino_pid.ino".

The code is as follows:

```

1  #include <PID_v1.h>
2
3  // PIN
4  #define LIMIT_KELUAR 2
5  #define LIMIT_MASUK 4
6  #define RPM 10
7  #define LPM 12
8  #define REN 6
9  #define LEN 8
10 #define IRL 7
11 #define IRL2 9
12 #define IRS 11
13 #define IRS2 19
14 #define IRS5 21
15 #define TRIG_PIN 3
16 #define ECHO_PIN 5
17
18 // Variabel
19 char trayState;
20 bool commandActive = false;
21 bool trayOut = false;
22 bool trayIn = false;
23 char tray_status = '0';
24 bool detect = false;
25 bool manualMasuk = false;
26
27 // Delay

```

The right sidebar shows symbols defined in the code, including functions like setup, loop, semuaRtidakMendeteksi, kontrolMotor, bacaJarak, motorKeluarPWM, motorMasukPWM, kirimKeESP, resetStatus, and stopMotor.

File Source code yang dibuat, diunggah di Github pada tautan :
<https://github.com/Gvnnisalim/Tugas-Akhir---Robot-Pengantar-Makanan>