

Robot Pengikut Objek Menggunakan Deteksi Warna Berbasis Smartphone Android dengan Kendali PID

Object Follower Robot Using Android Smartphone Based Color Detection with PID Control

Indra Kurniawan¹, Arif Fahmi²

Jurusan Bisnis dan Informatika, Politeknik Negeri Banyuwangi, Banyuwangi, Indonesia^{1,2}

Email: indrakurniawan@poliwangi.ac.id¹, ariffahmi@poliwangi.ac.id²

Abstract

Object-following robot is one of the popular research topics due to the various implementations of new technologies and their benefits in society. On the other hand, smartphones now have advantages as electronic devices that allow them to be used as a robot development media. This research attempted to develop an object-following robot that uses a smartphone as the primary sensor and data processor to generate PID (Proportional Integral Derivative) control signals in the robot. The research steps consisted of literature study, system design, identification of devices used, implementation, testing, and performance evaluation. The test results showed that the robot system with an Android smartphone has good performance. The program using a smartphone camera was able to detect colored objects with an average performance percentage of 84.36%. Motion testing consisting of translational motion, rotational motion, and combined motion (translation and rotation) produced maximum performance percentages of 100%, 84%, and 84%, respectively. This research showed that Android smartphones have adequate capabilities for use in robot systems, so that the implementation of robotic systems and algorithms might be developed and tested more easily without requiring difficult-to-obtain components.

Keywords: *Object Follower Robot, Color Detection, Android Smartphone, OpenCV, PID Control*

Abstrak

Robot pengikut objek merupakan salah satu topik penelitian populer karena berbagai implementasi teknologi baru dan manfaat-manfaatnya di masyarakat. Di sisi lain, *smartphone* sekarang ini memiliki keunggulan-keunggulan sebagai perangkat elektronik yang memungkinkannya digunakan sebagai media pengembangan robot. Penelitian ini berusaha mengembangkan robot pengikut objek yang menggunakan *smartphone* sebagai sensor dan pemrosesan data utama untuk menghasilkan isyarat pengendali PID (Proportional Integral Derivative) dalam robot. Tahap penelitian terdiri dari studi literatur, desain sistem, identifikasi perangkat yang digunakan, implementasi, pengujian, dan evaluasi kinerja. Hasil pengujian menunjukkan bahwa sistem robot dengan *smartphone* Android memiliki kinerja yang baik. Program yang menggunakan kamera *smartphone* dapat mendeteksi objek berwarna dengan persentase kinerja rata-rata sebesar 84,36%. Pengujian gerak yang terdiri dari gerak translasi, gerak rotasi, dan gerak gabungan (translasi dan rotasi) menghasilkan persentase kinerja maksimal masing-masing sebesar 100%, 84%, dan 84%. Penelitian ini menunjukkan bahwa *smartphone* Android memiliki kemampuan yang memadai untuk digunakan dalam sistem robot, sehingga implementasi sistem dan algoritma robotika dapat dikembangkan dan diuji dengan lebih mudah tanpa memerlukan komponen yang sulit didapat.

Kata Kunci: Robot Pengikut Objek, Deteksi Warna, *Smartphone* Android, OpenCV, Kendali PID

I. PENDAHULUAN

Robot pengikut objek merupakan salah satu topik penelitian yang populer di kalangan pendidik dan peneliti. Topik ini memberi peluang untuk implementasi berbagai teknologi baru baik di dalam ranah robotik maupun di luar ranah robotik. Contoh implementasi di ranah robotik antara lain teknologi sistem pengendali, mekanisme gerak robot, teknologi sensor gerak, dan sistem pengukuran gerak. Sedangkan di luar ranah robotik antara lain kecerdasan

buatan sebagai penentu perilaku robot. Robot pengikut objek sendiri juga memiliki manfaat-manfaat potensial di masyarakat. Antara lain untuk membantu tugas manusia membawakan barang-barang, atau menjaga keamanan orang-orang berkebutuhan khusus.

Terdapat macam-macam perangkat yang memungkinkan robot pengikut objek melakukan tugasnya. Beberapa di antaranya yakni pengukuran jarak dengan sensor ultrasonik [1], deteksi objek dengan kamera Pixy 2 [2], kamera stereo

ZED2i [3], kamera RGB dari Raspberry Pi [4], dan kamera *webcam* [5]. Dengan menggunakan perangkat yang sederhana, maka biaya pembuatan bisa ditekan dan proses perancangannya juga menjadi lebih sederhana. Namun, ini berarti perlu algoritma yang rumit untuk membuat robot menghasilkan kinerja yang baik. Sebaliknya, jika menggunakan perangkat yang canggih, maka algoritma bisa disederhanakan karena perangkat tersebut dapat memberikan informasi yang baik dan akurat. Namun, perlu biaya mahal untuk mendapatkannya juga proses pengaturannya lebih rumit.

Di sisi lain, ponsel pintar atau *smartphone* telah menjadi perangkat yang sangat dekat dengan masyarakat belakangan ini. Di Indonesia pada tahun 2023, jumlah pengguna *smartphone* adalah sebesar 190,03 juta orang [6]. Karena itu, perkembangan teknologi *smartphone* juga semakin maju. Sekarang ini, fitur-fitur yang ada di *smartphone* khususnya yang berbasis Android sudah sangat canggih. Beberapa di antaranya yaitu kamera beresolusi besar depan-belakang, konektivitas nirkabel melalui *wifi* dan *bluetooth* bahkan NFC, sistem GPS, serta berbagai sensor seperti *accelerometer* dan *gyroscope*. Kemudian, jika menilik pada spesifikasi perangkat keras, maka kebanyakan *smartphone* memiliki unit pemrosesan yang handal. Kemampuan tersebut dapat digunakan untuk menangani banyak tugas berat seperti kalkulasi intensif atau melakukan *multitasking*.

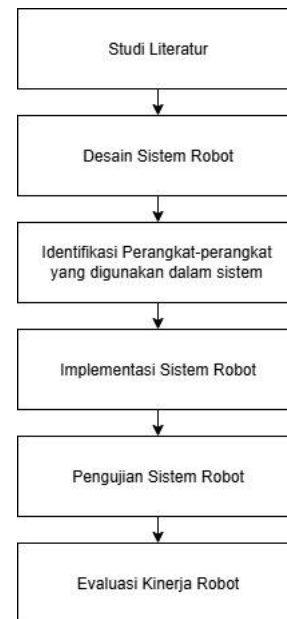
Karena keunggulan-keunggulan *smartphone* sebagai perangkat elektronik seperti yang disebut di atas, maka sangat mungkin menggunakan *smartphone* sebagai media pengembangan robot. *Smartphone* dapat difungsikan sebagai unit pengendali dan sensor. Beberapa penelitian sebelumnya telah menggunakan *smartphone* dalam pengembangan robot, khususnya pada media *input* [7], [8].

Pada penelitian ini, digunakan pengendali PID (*Proportional, Integral, Derivative*) yang populer digunakan dalam berbagai bidang penerapan, termasuk robot. PID merupakan sistem kendali tertutup (*closed-loop*) yang menggunakan umpan balik (*feedback*) yang didapat dari selisih nilai pembacaan sensor dengan nilai yang diinginkan. Beberapa penerapan PID sebagai pengendali robot antara lain pada robot *unmanned ground vehicle* (UGV) [9], robot penyeimbang (*balancing*) beroda dua [10], dan robot berkaki enam (*hexapod*) [11].

Penelitian ini bertujuan untuk merancang dan mengevaluasi sistem robot pengikut objek dengan menggunakan *smartphone* Android. Fokus utama dari penelitian ini adalah penggunaan *smartphone* Android yang secara bebas beredar di masyarakat sebagai media sensor dan pemrosesan data utama dalam sebuah robot. Pada penelitian ini, dihasilkan sebuah robot empat roda yang dapat diuji secara langsung untuk mengetahui kinerja kamera *smartphone* untuk mendeteksi objek, kemampuan pemrosesan data *smartphone* untuk mengolah isyarat pengendali robot, dan kemampuan komunikasi *smartphone* dengan komponen-komponen robotika.

II. METODE PENELITIAN

Penelitian ini terdiri dari beberapa tahap. Tahap pertama dimulai dari studi literatur mengenai sistem robot pengikut objek. Tahap selanjutnya adalah desain. Proses desain dilakukan dengan memperhatikan fungsi dari setiap perangkat, sehingga dapat menempatkannya dengan baik di badan robot. Hal yang paling diperhatikan di tahap ini adalah penggunaan *smartphone* sebagai perangkat yang menangkap citra, sehingga perlu diperhatikan penempatannya dengan baik. Tahap berikutnya adalah identifikasi perangkat-perangkat yang digunakan. Di tahap ini, setiap perangkat ditelaah untuk dapat dipergunakan secara dengan baik di sistem robot. Tahap selanjutnya adalah implementasi. Setiap bagian robot lalu dirangkai sesuai desain yang telah dibuat. Sistem deteksi objek, kelistrikan robot, pergerakan robot dan lain-lain diaktifkan dalam sistem robot. Tahap ini akan menghasilkan robot lengkap yang siap untuk melaksanakan tugasnya. Tahap selanjutnya adalah pengujian sistem robot yang khusus untuk menguji kerja deteksi objek dan kemampuan sistem pengendali PID. Tahap terakhir adalah evaluasi kinerja robot secara keseluruhan.



Gambar 1. Diagram Alur Penelitian Robot Pengikut Objek Menggunakan Deteksi Warna Berbasis Smartphone Android

A. Desain Sistem Robot

Konsep utama sistem yang digunakan robot di penelitian ini adalah menggunakan *smartphone* sebagai inti pengendalian robot. Tugas-tugas *smartphone* antara lain untuk menangkap citra di depan robot, mendeteksi objek berdasarkan warna, mengolah data input sehingga menghasilkan data pengendali, dan mengirimkan data pengendali tersebut melalui koneksi serial (USB).

Dari *smartphone*, pengendalian robot lalu diteruskan pada badan robot. Data pengendalian dari *smartphone* tersebut

lalu masuk ke unit *microcontroller* yang lalu mengolahnya untuk menghasilkan sinyal ke *motor driver*. Dari sana lalu didapat sinyal input ke keempat motor yang terpasang di setiap roda robot.

B. Identifikasi Perangkat yang Digunakan

Desain sistem di atas menjadi dasar untuk menentukan perangkat-perangkat yang digunakan untuk membangun robot. Dalam penelitian ini, telah ditentukan perangkat-perangkat yang digunakan sebagai berikut:

1) Smartphone

Dalam penelitian ini, *smartphone* yang digunakan adalah Oppo A1k. Spesifikasi umum dari *smartphone* ini antara lain:

- *Processor* yang digunakan adalah Mediatek MT6762 Helio P22 yang memiliki 8 *core* (Octa-core) dengan kecepatan masing-masing *core* 2.0 GHz yang berarsitektur Cortex-A53
- Sistem operasi menggunakan Android 9.0 (Pie)
- Kamera depan (*selfie*) beresolusi 5 Megapixel (MP) dengan kemampuan merekam video 720p.

Berdasarkan spesifikasi tersebut, *smartphone* ini memiliki kemampuan yang memadai untuk menjalankan tugas-tugas yang diperlukan robot dalam penelitian ini.

2) Arduino Uno

Arduino Uno adalah sebuah *microcontroller board* berbasis ATmega328P [12]. Perangkat ini memiliki 14 pin input/output digital (6 di antaranya dapat digunakan sebagai output PWM), 6 input analog, 16 MHz resonator keramik (CSTCE16M0V53-R0), satu buah koneksi USB, sebuah *power jack*, sebuah ICSP header dan sebuah tombol reset.

Perangkat ini dipilih karena kemampuannya yang beragam, kemudahan dalam memprogram dan penggunaan, serta kemudahan dalam mendapatkannya. ATmega328P sendiri merupakan *microcontroller* 8-bit [13] yang dalam penelitian ini cukup untuk mengatur jalannya motor DC yang terpasang pada robot. Memprogram perangkat Arduino secara umum dapat dilakukan dengan menyambungkan perangkat tersebut ke komputer melalui koneksi USB. Kemudian di dalam komputer, program dibuat dengan *software* khusus yang ditulis dengan bahasa pemrograman tingkat tinggi khusus untuk Arduino yang mirip dengan bahasa C. *Software* yang sering digunakan adalah Arduino IDE. Dalam implementasi di *hardware*, Arduino dapat dihubungkan ke perangkat-perangkat lain melalui pin-pin yang ia miliki. Selain itu, terdapat banyak perangkat *shield* yang dapat dipasang langsung pada Arduino Uno dengan fungsi-fungsi spesifik. Arduino Uno dapat dibeli dengan mudah di negeri ini. Selain itu, Arduino Uno merupakan perangkat yang umum digunakan dalam modul pembelajaran sekolah maupun perguruan tinggi terutama di bidang elektronika.

Dalam penelitian ini, Arduino Uno berfungsi sebagai *microcontroller* robot yang mengendalikan kerja motor DC

(melalui *shield*). Sehingga sinyal pengendali dari *smartphone* lalu diterjemahkan menjadi sinyal yang dapat mengaktifkan motor

3) Four Wheels Car Robot Kit

Dalam penelitian ini, robot yang digunakan dibangun berdasarkan kit robot empat roda. Robot ini memiliki motor DC yang menggerakkan setiap rodanya. Kemudian untuk sasis, kit ini menggunakan material berbahan akrilik yang disatukan dengan mur dan baut. Selain sasis, roda, dan motor, robot juga memerlukan sumber daya. Sumber daya yang digunakan adalah baterai 18650 sebanyak empat buah. Keempat baterai tersebut dipasang pada dua tempat baterai yang masing-masing dapat menampung dua baterai yang terhubung secara paralel.

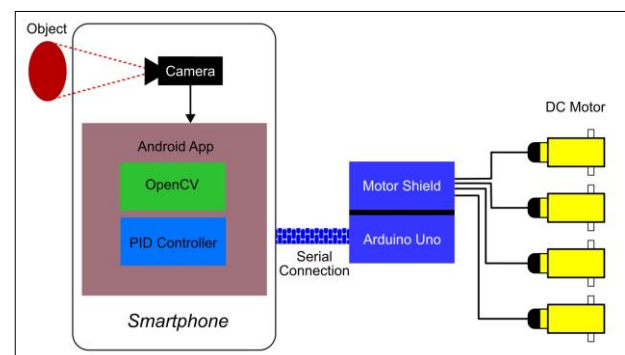
4) L293D Arduino Motor Driver Shield

L293D Arduino Motor Driver Shield adalah komponen yang serba guna dan populer untuk mengendalikan berbagai macam motor listrik dengan Arduino. Komponen ini berfungsi mengendalikan gerak motor listrik berdasarkan sinyal input yang diberikan oleh Arduino.

Komponen ini, sesuai namanya, menggunakan IC L293D *motor driver* untuk menggerakkan motor. *Motor driver* ini merupakan H-Bridge dua jalur yang dapat mengendalikan dua motor DC atau satu motor *stepper*. Di dalam *shield* yang dimaksud, terdapat dua buah *driver* ini sehingga kemampuan mengendalikannya menjadi dua kali lipatnya. Untuk dapat menghubungkan kedua *driver* ini, di dalam *shield* terpasang *shift register* 74HC595 sehingga dapat memperbanyak output sinyal dari empat pin Arduino menjadi delapan input sinyal di kedua *driver*.

Dalam penelitian ini, L293D Arduino Motor Driver Shield digunakan sebagai penggerak utama motor DC dari robot. Hal ini mengingat robot menggunakan *kit* yang memiliki empat roda dengan masing-masing roda digerakkan satu motor DC.

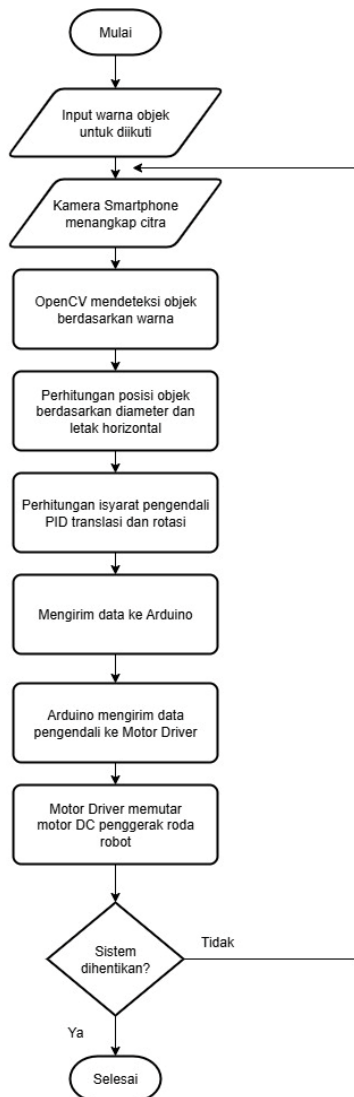
C. Implementasi



Gambar 2. Alur Kerja Sistem Robot

Untuk bisa mengolah citra gambar kamera dan mengolah data pengendali, *smartphone* menggunakan aplikasi yang dibuat dengan *software* Android Studio. Aplikasi ini menggunakan OpenCV, sebuah pustaka *cross-platform* di

bidang Visi Komputer untuk pemrosesan citra secara *real-time* [14]. OpenCV yang digunakan adalah versi 3.4.3.



Gambar 3. Flowchart Sistem Kerja Robot Pengikut Objek Menggunakan Deteksi Warna Berbasis Smartphone Android

Konsep deteksi objek yang digunakan dalam penelitian ini berdasarkan pada salah satu algoritma dalam OpenCV yaitu *Color Blob Detector*. Algoritma ini bertujuan untuk mencari daerah atau area dari suatu citra/gambar yang memiliki kesamaan properti, dalam hal ini adalah warna.

Implementasi dari algoritma ini diawali dengan mengubah *channel* gambar berwarna yang menggunakan ruang warna RGB (*Red, Green, Blue*) menjadi HSV (*Hue, Saturation, Value*). Penggunaan ruang warna HSV menyebabkan program dapat dengan lebih mudah mengabaikan perbedaan dalam cahaya atau kontras [15], yang memungkinkan program lebih mudah melakukan pengelompokan. Teknik pengelompokan area menerapkan pendeteksian yang mirip dengan metode *thresholding*. Metode *thresholding* biasa digunakan pada gambar *grayscale* (satu *channel*). Metode ini

bertujuan untuk mengelompokkan *pixel-pixel* gambar berdasarkan satu nilai *threshold*, baik itu *pixel* dengan nilai lebih kecil dari *threshold*, lebih besar dari *threshold*, atau sama dengan *threshold* [16].

Untuk gambar berwarna (tiga *channel*), maka nilai *threshold* ini menjadi satu set *threshold* dengan tiga nilai. Selain itu, teknik pengelompokan sendiri juga berubah. Di sini, terdapat dua set *threshold*, yang disebut nilai minimum dan maksimum. Suatu area di citra/gambar yang memiliki warna yang diinginkan, harus memiliki *pixel* dengan nilai-nilai *channel* di antara kedua set nilai tersebut. Area lain yang tidak memenuhi kriteria tersebut dianggap tidak memiliki warna yang diinginkan.

Setelah didapat pemisahan area dengan warna yang diinginkan, maka proses selanjutnya adalah mencari *contour*. *Contour* adalah garis yang mengelilingi area tepi semua titik *pixel* yang berada di kelompok yang sama. Berdasarkan hasil proses sebelumnya, *contour* ini akan memperlihatkan area-area di mana warna yang diinginkan terkumpul. Proses ini mungkin akan menghasilkan beberapa *contour* dalam setiap citra/gambar. Hal ini karena ada kemungkinan citra/gambar menangkap *error/noise* yang menghasilkan area dengan warna yang sama dengan objek. Bisa juga karena dalam citra/gambar tersebut memang terdapat benda dengan warna yang sama dengan objek.

Dari *contour* hasil *Color Blob Detection* tersebut, maka dapat dicari letak objek dalam citra/gambar. Dalam citra/gambar, objek yang ingin dideteksi akan menghasilkan *contour* dengan area paling luas. Lingkaran yang melingkupi *contour* lalu dibuat, sehingga didapat titik tengah dan radius lingkaran.

Titik tengah lingkaran pelingkup digunakan untuk mencari selisih orientasi robot dibandingkan objek. Jika robot sudah menghadap objek dengan sempurna, maka titik tengah tersebut harus berada di tengah-tengah citra yang ditangkap kamera. Area tengah citra ini merupakan nilai referensi. Jika titik tengah lingkaran objek berada di kanan citra, maka robot harus bergerak memutar sehingga titik tengah itu menjadi di tengah. Hal yang sama juga terjadi jika titik tengah berada di kiri citra, dengan arah putaran yang terbalik. Citra kamera yang ditangkap *smartphone* dalam penelitian ini akan menghasilkan gambar HD, yang berarti memiliki lebar 1280 *pixel*. Karena itu, titik tengahnya di *pixel* ke 640. Dengan menggunakan nilai toleransi sebesar 10 *pixel*, maka nilai referensi area tengah dalam program adalah 630-650 *pixel* diukur dari kiri gambar yang dihasilkan.

Radius lingkaran pelingkup digunakan untuk mencari seberapa jauh jarak objek terhadap robot. Terdapat nilai referensi tertentu yang menunjukkan jarak objek sudah sesuai. Jika radius lingkaran ini lebih kecil dari nilai referensi, maka menunjukkan jarak objek lebih jauh dari nilai yang diinginkan, sehingga robot perlu bergerak maju. Sebaliknya, jika radius lebih besar, maka objek berada terlalu dekat sehingga robot harus bergerak mundur. Nilai referensi radius lingkaran yang digunakan dalam program adalah 140-

160 pixel.

Kedua nilai ini menyebabkan terdapat dua derajat kebebasan (*degree of freedom*) pada robot. Yang pertama adalah gerak translasi posisi robot, dan yang kedua adalah gerak rotasi untuk orientasi robot. Masing-masing gerak tersebut memerlukan pengendali yang berbeda.

Pemrosesan data pengendali dalam aplikasi *smartphone* menggunakan konsep pengendali PID (*proportional, integral, derivative*). Pengendali ini merupakan jenis pengendali umpan balik yang menggabungkan tiga komponen (algoritma) pengendali, yaitu *proportional*, *integral*, dan *derivative* [17]. Umpan balik berasal dari pembacaan sensor, yang dicari selisihnya dengan nilai referensi atau nilai tujuan pengendalian untuk mendapatkan *error* (*e*). Selisih ini merupakan representasi besarnya kesalahan kondisi sistem dengan kondisi yang diinginkan. Persamaan pengendali PID dapat dituliskan sebagai berikut:

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

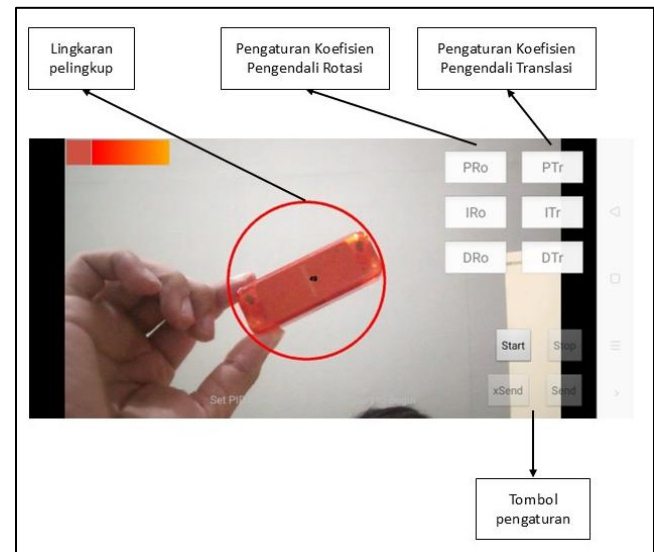
Komponen *proportional* akan mengalikan *error* tersebut dengan suatu nilai tertentu, yang biasa disebut konstanta K_p . Komponen *integral* dan *derivative* akan mengintegrasikan dan menurunkan nilai *error* secara kalkulus. Masing-masing dari kedua komponen tersebut kemudian dikalikan dengan suatu konstanta K_i dan K_d . Hasil pemrosesan dari ketiga komponen tersebut lalu dijumlahkan sehingga mendapatkan data pengendali (u). Terdapat dua data pengendali, yaitu data pengendali translasi dan data pengendali rotasi, sesuai dengan derajat kebebasan robot.

Data ini kemudian dikirimkan ke Arduino melalui komunikasi serial dengan perantara kabel USB. Di dalam Arduino, data pengendali translasi dan rotasi lalu digabungkan untuk menghasilkan nilai-nilai kecepatan setiap roda. Dengan menggunakan *library* khusus, nilai-nilai ini lalu mengatur motor melalui L293D Arduino *Motor Driver Shield*. Dengan demikian, proses perhitungan lebih dibebankan kepada *smartphone* dibandingkan Arduino. Hal ini mengingat kemampuan komputasi *smartphone* lebih tinggi dibandingkan Arduino yang bertujuan membangun sistem yang lebih stabil dan responsif.

III. HASIL DAN PEMBAHASAN

A. Pembuatan Program Android

Program dalam penelitian ini memiliki tampilan dan fitur sederhana. Program ini hanya punya satu tampilan yang memperlihatkan hasil tangkapan kamera dan pengaturan-pengaturan yang diperlukan. Gambar 2 memperlihatkan hasil tangkapan layar *smartphone* saat membuka program tersebut.



Gambar 4. Tampilan Program Android

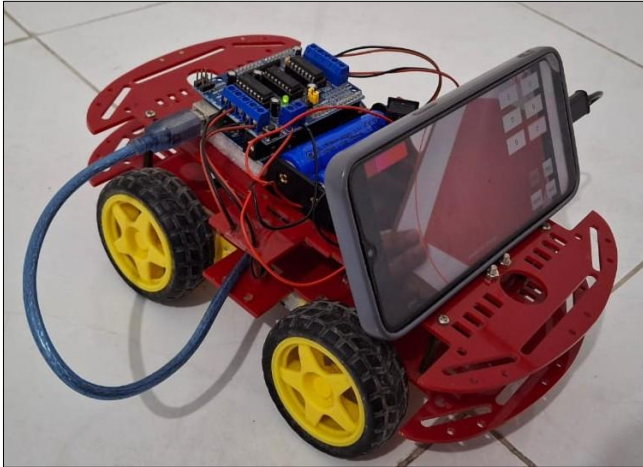
Program bekerja dengan memanfaatkan kamera depan *smartphone*. Saat program dibuka, layar akan menampilkan hasil tangkapan kamera depan secara *real-time*. Pemilihan warna objek yang ingin dideteksi dilakukan dengan menempatkan objek pada area yang dapat ditangkap kamera *smartphone*, lalu gambar objek yang ditampilkan di layar ditekan sehingga program menyimpan data warna yang dimaksud. Program kemudian menampilkan lingkaran di sekeliling objek untuk menunjukkan bahwa program tersebut sudah bisa melakukan proses deteksi.

Di sudut kanan atas tampilan program, terdapat kolom-kolom input yang dibuat dengan elemen 'EditText'. Kolom-kolom tersebut digunakan untuk melakukan pengaturan koefisien-koefisien pengendali. Terdapat total enam input karena terdapat dua jenis pergerakan yang dikendalikan, dan setiap jenis gerakan menggunakan tiga koefisien pengendali.

Di sudut kanan bawah terdapat tombol-tombol yang dibuat dengan elemen 'Button'. Tombol-tombol tersebut digunakan untuk mengendalikan jalannya pengiriman data dari *smartphone* ke Arduino. Tombol 'Start' digunakan untuk mengaktifkan kemampuan pengiriman data. Tombol 'Stop' digunakan untuk menonaktifkan kemampuan pengiriman data. Tombol 'Send' digunakan untuk memulai pengiriman data. Tombol 'xSend' digunakan untuk menghentikan pengiriman data.

B. Perakitan Robot

Perakitan robot dilakukan dengan menggabungkan semua komponen penyusun robot. Gambar di bawah ini menunjukkan robot yang sudah lengkap.



Gambar 5. Robot Pengikut Objek Lengkap dengan Semua Komponen

C. Pengujian Deteksi Objek

Pengujian deteksi objek dilakukan dengan *smartphone* Android yang terlepas dari robot.

Objek yang digunakan untuk menguji adalah kotak berwarna yang dicetak di lembar kertas. Ukuran kotak tersebut adalah lima belas kali lima belas sentimeter. Objek lalu ditempatkan di tempat-tempat yang berbeda untuk menguji bagaimana kemampuan program untuk mendeteksi keberadaan objek. Terdapat tiga warna objek yang digunakan dalam pengujian ini, yaitu merah, biru, dan hijau. Ketiga warna tersebut dipilih berdasarkan cara kerja kamera yang menangkap citra berdasarkan tiga *channel* warna. Dengan menggunakan ketiga warna tersebut, diharapkan kinerja sistem untuk setiap *channel* warna dapat diketahui. Proses pengujian dilakukan dengan merekam layar *smartphone* sehingga didapat periode waktu saat program berhasil memberi lingkaran yang tepat dan tidak. Periode keberhasilan ini kemudian dipersentasekan dengan total waktu pengujian sehingga menghasilkan persentase kinerja pendeteksian objek.



Gambar 6. Pengujian Deteksi Objek Berjarak 50 cm dengan Tiga Warna Berbeda

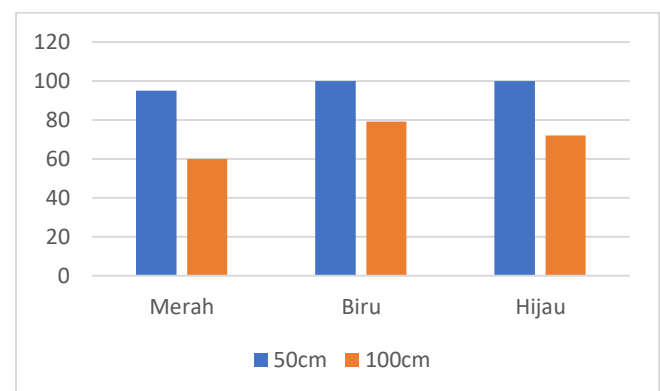
Gambar 4 menunjukkan hasil pengujian deteksi objek dengan jarak 50 sentimeter dari *smartphone*. Objek diubah-ubah posisinya, yaitu di kiri, tengah, dan kanan relatif terhadap *smartphone*. Program menghasilkan persentase

kinerja sebesar 95% untuk pendeteksian objek berwarna biru, kemudian 100% untuk pendeteksian objek berwarna merah dan hijau.



Gambar 7. Pengujian Deteksi Objek Berjarak 100 cm dengan Tiga Warna Berbeda

Gambar 5 menunjukkan hasil pengujian deteksi objek dengan jarak 100 sentimeter dari *smartphone*. Sama seperti pengujian sebelumnya, posisi objek diubah-ubah. Selama pengujian, program tetap mampu menampilkan lingkaran yang sesuai dengan letak objek, tapi konsistensi dan ketepatannya berkurang. Ada beberapa kesempatan lingkaran tidak melingkupi keseluruhan objek, atau lingkaran berada di area yang sama sekali tidak menunjukkan objek. Program menghasilkan persentase kinerja sebesar 60% untuk objek berwarna merah, 79,2% untuk objek berwarna biru, dan 72% untuk objek berwarna hijau. Penurunan kinerja ini terjadi karena permasalahan pengaturan otomatis yang ada di kamera *smartphone* saat terjadi perubahan pencahayaan. Citra objek yang ditangkap kamera menjadi tidak konsisten, menyesuaikan tingkat kecerahan kamera. Program secara keseluruhan memiliki rata-rata persentase kinerja sebesar 84,36%.



Gambar 8. Persentase Kinerja Pendeteksian Objek dari *Smartphone* Android

D. Pengujian Gerak Robot

Pengujian gerak robot menggunakan robot lengkap dengan semua komponennya. Pengujian ini terdiri dari tiga bagian yaitu pengujian gerak translasi, pengujian gerak rotasi, dan pengujian gabungan. Setiap pengujian menggunakan nilai koefisien pengendali yang berbeda-beda

untuk memperlihatkan respons kendali robot. Persentase kinerja merupakan penilaian atas kemampuan robot dalam bergerak sesuai dengan posisi objek. Penilaian ini dilakukan berdasarkan stabilitas dan kecepatan gerak robot. Nilai 0% berarti robot memiliki gerak yang tidak stabil dan lambat, sementara 100% berarti robot dapat bergerak dengan stabil dan cepat.

Pengujian gerak translasi dilakukan dengan mengeset koefisien pengendali rotasi menjadi nol agar kendali rotasi tidak aktif. Koefisien pengendali translasi lalu diubah-ubah nilainya sesuai skenario pengujian. Objek berwarna lalu ditempatkan di depan robot. Objek yang digunakan menggunakan salah satu dari objek warna pada pengujian deteksi objek, karena tidak ada perbedaan hasil di antara ketiga warna. Objek lalu didekatkan dan dijauhkan untuk melihat respons gerak robot. Saat objek didekatkan, maka respons yang diharapkan adalah robot bergerak mundur. Sebaliknya, saat objek dijauhkan maka respons yang diharapkan adalah robot bergerak maju. Sebelum pengujian, dilakukan pengukuran untuk mengetahui jarak objek yang memicu respons robot. Dari hasil pengukuran ini didapat jarak objek kurang dari 55 cm dari robot akan membuat robot bergerak mundur, sedangkan jarak lebih dari 128 cm akan membuat robot bergerak maju mendekat. Tabel 1 memperlihatkan hasil pengujian gerak translasi.

Tabel 1. Hasil Pengujian I Gerak Translasi dengan Berbagai Kombinasi Koefisien Pengendali Translasi

Kp	Ki	Kd	Persentase Kinerja (%)	Keterangan
8	0	1	50	stabil respon tidak cepat
8	1	1	50	osilasi tidak berhenti, respons cepat
8	1	2	50	osilasi tidak berhenti, respons cepat
8	1	3	67	osilasi berhenti setelah mundur, respons cepat
8	1	4	67	osilasi berhenti setelah mundur, respons cepat
8	1	5	67	osilasi berhenti setelah mundur/maju, respons cepat
8	1	6	67	osilasi berhenti setelah mundur/maju, respons cepat
8	1	7	67	osilasi berhenti setelah mundur/maju, respons cepat
8	1	8	67	osilasi berhenti setelah mundur/maju, respons cepat
10	0	0	50	stabil respon tidak cepat
10	0	1	50	stabil respon tidak cepat
10	1	1	50	osilasi tidak berhenti, respons cepat
10	1	2	50	osilasi tidak berhenti, respons cepat
10	1	3	50	osilasi tidak berhenti, respons cepat
10	1	4	67	osilasi berhenti setelah mundur, respons cepat
10	1	5	67	osilasi berhenti setelah mundur/maju, respons cepat

10	1	6	67	osilasi berhenti setelah mundur/maju, respons cepat
10	1	7	67	osilasi berhenti setelah mundur/maju, respons cepat
10	1	8	67	osilasi berhenti setelah mundur/maju, respons cepat

Pada hasil pengujian tersebut, penulis menggunakan nilai-nilai koefisien secara acak dengan nilai yang sekecil mungkin. Tujuan dari pengujian ini adalah untuk mengetahui pengaruh dari masing-masing komponen pengendali pada gerak robot. Penggunaan pengendali *Proportional* saja (dengan membuat K_i dan K_d sama dengan nol) atau *Proportional* dan *Derivative* saja (dengan membuat K_i sama dengan nol) memperlihatkan respons gerak robot yang stabil tapi tidak cepat. Dengan menambahkan komponen *Integral*, respons menjadi lebih cepat tapi terjadi osilasi sehingga robot kesulitan untuk berhenti pada posisi yang diharapkan. Dengan meningkatkan nilai koefisien K_d , osilasi berkurang sehingga robot bisa berhenti pada posisi yang tepat setelah maju atau mundur beberapa kali. Selain karena faktor pengendali, osilasi juga terjadi karena program Android terkadang gagal untuk melakukan deteksi objek. Permasalahan pencahayaan pada pengujian deteksi objek sebelumnya kembali terulang pada pengujian gerak ini. Saat hal tersebut terjadi, data pengendali menjadi tidak sesuai dengan keberadaan objek sehingga robot menjadi tidak stabil.

Pada pengujian di atas, motor kesulitan untuk menggerakkan robot dari keadaan diam. Terkadang perlu dorongan dari luar dulu agar robot mulai bisa berjalan. Persentase kinerja yang dihasilkan juga memiliki nilai yang buruk, yaitu di antara 50-67%. Karena itu, dilakukan pengujian kedua dengan meningkatkan nilai K_p dan K_d agar dorongan awal motor menjadi lebih besar.

Tabel 2. Hasil Pengujian II Gerak Translasi dengan Kombinasi Koefisien Pengendali Translasi yang Lebih Besar

Kp	Ki	Kd	Persentase Kinerja (%)	Keterangan
30	0	10	75	Stabil, respons kurang cepat
30	1	10	100	Stabil, respons cepat
20	1	10	42	Kurang stabil, respons kurang cepat

Dengan penggunaan K_p dan K_d yang besar, motor langsung bisa bergerak setiap objek dijauhkan atau didekatkan. Respons juga lebih cepat dibandingkan pada pengujian pertama. Robot juga mampu untuk berhenti pada posisi yang diharapkan. Dalam pengujian ini terkadang masih terdapat permasalahan deteksi objek karena pencahayaan. Jika hal ini terjadi, maka robot kembali tidak stabil dengan gerakan yang tidak menentu.

Terdapat tiga kombinasi koefisien pengendali translasi pada pengujian ini, dengan hasil kombinasi koefisien terbaik adalah $K_p = 30$, $K_i = 1$, dan $K_d = 10$. Kombinasi ini membuat robot bergerak dengan stabil dengan respons lebih cepat dibandingkan dua kombinasi yang lainnya. Nilai-nilai

koefisien ini memberikan persentase kinerja sebesar 100%, yang berarti robot sangat baik dalam merespons posisi objek.

Pengujian gerak rotasi hampir sama dengan pengujian translasi. Koefisien pengendali translasi diset nol dan koefisien pengendali rotasi diubah-ubah. Objek yang ditempatkan di depan robot lalu digeser ke kiri dan ke kanan untuk mengamati kemampuan robot dalam mengubah arah orientasi. Pada pengujian ini, digunakan nilai koefisien pengendali rotasi K_p dan K_d yang besar karena robot tidak bisa bergerak sama sekali saat nilai-nilai koefisien tersebut kecil seperti pada pengujian pertama gerak translasi. Tabel 3 memuat hasil pengujian gerak rotasi robot.

Tabel 3. Hasil Pengujian Gerak Rotasi dengan Berbagai Kombinasi Koefisien Pengendali Rotasi

Kp	Ki	Kd	Persentase Kinerja (%)	Keterangan
80	0	10	50	Stabil, respons tidak cepat
80	0	15	17	Kurang stabil, respons tidak cepat
80	1	10	59	Cukup stabil, respons cukup cepat
80	2	10	67	Kurang stabil, respons cepat
75	1	10	84	Cukup stabil, respons cepat
75	2	10	67	Kurang stabil, respons cepat

Pada pengujian rotasi ini, didapat kombinasi nilai koefisien terbaik adalah $K_p = 75$, $K_i = 1$, dan $K_d = 10$. Dengan menggunakan nilai-nilai ini, gerak robot cukup stabil mengikuti objek, dengan respons yang cepat. Dari hasil pengujian ini, juga terlihat bahwa peningkatan nilai K_i bisa meningkatkan respons gerak rotasi, tapi berakibat kestabilan berkurang. Hal ini ditunjukkan dengan gerak robot yang berputar ke kanan dan ke kiri serta sulit berhenti sesuai dengan posisi objek. Permasalahan deteksi objek karena pencahayaan masih terjadi pada pengujian ini. Jika permasalahan ini muncul, maka gerak robot menjadi tidak menentu dan tidak mengikuti objek (antara berputar-putar di tempat atau diam).

Pengujian gerak gabungan dilakukan dengan mengaktifkan pengendali translasi dan rotasi. Objek berwarna kembali ditempatkan di depan robot. Kemudian, objek tersebut digerakkan menjauh dari robot dengan jalur tertentu yang merupakan kombinasi jalur lurus dan satu belokan untuk mengetahui kemampuan robot untuk mengikuti objek. Hasil pengujian gerak gabungan termuat dalam Tabel 4.

Skenario pengujian awalnya menggunakan nilai-nilai koefisien terbaik dari dua pengujian sebelumnya. Namun, penggunaan nilai-nilai ini menimbulkan gerak yang tidak stabil dari robot. Gerak robot menjadi tidak terarah karena respons berbeloknya terlalu cepat. Karena itu, pada skenario selanjutnya, nilai K_p pada pengendali rotasi dikurangi hampir setengahnya. Dengan menggunakan nilai tersebut, gerak robot menjadi lebih stabil sehingga robot bisa mengikuti pergerakan objek sesuai jalur yang telah ditentukan. Nilai-nilai koefisien terbaik pada pengujian ini adalah $K_p = 20$, $K_i = 1$, dan $K_d = 10$ untuk pengendali

translasi serta $K_p = 40$, $K_i = 1$, dan $K_d = 10$ untuk pengendali rotasi. Nilai-nilai ini membuat pergerakan robot cukup stabil dengan respons maju dan mundur serta gerak membelok yang cukup cepat. Sifat pergerakan ini memungkinkan robot bisa mengikuti gerak objek dari awal sampai akhir jalur yang telah ditentukan. Pada pengujian ini, permasalahan pencahayaan tetap muncul yang menyebabkan hilangnya kemampuan deteksi objek pada program *smartphone* Android. Sama seperti pengujian sebelumnya, jika hal ini terjadi maka gerak robot menjadi tidak menentu dan tidak bisa mengikuti objek sama sekali.

Tabel 4. Hasil Pengujian Gerak Gabungan dengan Berbagai Kombinasi Koefisien Pengendali

Koefisien Translasi			Koefisien Rotasi			Persentase Kinerja (%)	Keterangan
Kp	Ki	Kd	Kp	Ki	Kd		
30	1	10	75	0	10	0	Tidak stabil, tidak terarah
30	1	10	40	1	10	68	Cukup stabil, respons membelok cepat hingga terlihat terpisah dengan gerak maju
30	1	10	40	1	10	59	Cukup stabil, respons membelok kurang cepat
20	1	10	40	1	10	84	Cukup stabil, respons membelok cepat

IV. KESIMPULAN

Pada penelitian ini, robot telah diuji secara menyeluruh mulai dari deteksi objek, gerak translasi, gerak rotasi, dan gerak gabungan. Program *smartphone* Android telah mampu melakukan deteksi objek berdasarkan warna serta mampu memproses isyarat pengendali berdasarkan konsep pengendali PID. Robot mampu bergerak sesuai isyarat yang diberikan, sesuai dengan jenis pengujian. Secara keseluruhan, sistem robot yang telah dikembangkan berhasil mengikuti objek berwarna dengan menggunakan nilai-nilai koefisien pengendali PID yang tepat.

Hasil penelitian ini menunjukkan bahwa *smartphone* Android memiliki kemampuan yang memadai untuk digunakan dalam sistem robot. Kamera dalam *smartphone* sudah memiliki kinerja yang baik dalam mendeteksi objek. Selanjutnya, unit pemrosesan dalam *smartphone* sudah dapat secara andal mengolah isyarat pengendali secara *real-time*. Karena itu, implementasi sistem dan algoritma robotika dapat dikembangkan dan diuji dengan lebih mudah dengan

sistem robot dengan *smartphone* tanpa memerlukan komponen robot yang sulit didapat.

Terdapat beberapa kekurangan robot pengikut objek di penelitian ini. Kekurangan utama dari sistem robot adalah kemampuan deteksi objek yang sangat dipengaruhi oleh pencahayaan. Jika kondisi pencahayaan berubah, maka program Android kehilangan kemampuan untuk mendeteksi objek. Hal ini sangat membatasi kemampuan robot dalam mengikuti objek. Kekurangan lain dari robot ini adalah perlunya mencari nilai-nilai koefisien pengendali yang tepat. Terdapat total enam nilai koefisien yang perlu diperhatikan agar gerak robot menjadi stabil dan responsif.

Perlu adanya penelitian lanjutan untuk mengatasi masalah pencahayaan dan sistem kendali yang rumit tersebut. Selain itu, penelitian ini juga belum menggunakan fitur-fitur yang potensial dalam *smartphone* Android seperti sensor akselerasi, sensor orientasi, sensor *proximity*, GPS, serta konektivitas *Bluetooth*. Diharapkan, terdapat penelitian-penelitian lanjutan untuk mengimplementasikan fitur-fitur tersebut, baik robot pengikut objek maupun jenis robot lain.

REFERENSI

- [1] D. R. Palupi, D. Syaury, and A. S. Budi, "Object Following Robot berbasis Pembacaan Jarak menggunakan Metode PID Controller," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 4, no. 7, Art. no. 7, Aug. 2020.
- [2] R. Wairini, "Development of Color-Based Object Follower Robot Using Pixy 2 Camera and Arduino to Support Robotics Practice Learning," *Jurnal Edukasi Elektro*, vol. 7, no. 2, Art. no. 2, Nov. 2023, doi: 10.21831/jee.v7i2.64413.
- [3] M. I. Ramadhan, D. Purwanto, H. Kusuma, and H. Kusuma, "Sistem Pengikut Manusia pada Robot Servis Menggunakan Model YOLO dan Kamera Stereo," *JTITS*, vol. 12, no. 3, pp. A194–A201, Dec. 2023, doi: 10.12962/j23373539.v12i3.118869.
- [4] A. Sadiyoko, A. Fahrnasa, and T. M. Rahayu, "Perancangan Robot Pengikut berbasis Visual dan Kontrol Logika Samar," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 12, no. 1, Art. no. 1, Jan. 2024, doi: 10.26760/elkomika.v12i1.106.
- [5] M. Amangesti, D. Puspita, S. Sulaeman, and M. C. Rijal, "Rancang Bangun Troli Pengikut Objek Otomatis," *SNTEI*, vol. 9, no. 1, Art. no. 1, Nov. 2023.
- [6] B. P. S. Indonesia, "Proporsi Individu yang Menguasai/Memiliki Telepon Genggam Menurut Provinsi - Tabel Statistik." Accessed: Aug. 06, 2024. [Online]. Available: <https://www.bps.go.id/id/statistics-table/2/MTIyMSMy/proporsi-individu-yang-menguasai-memiliki-telepon-genggam-menurut-provinsi.html>
- [7] W. A. Salah, A. A. Sneineh, and A. A. A. Shabaneh, "Smartphone Sensor-based Development and Implementation of a Remotely Controlled Robot Arm," *Journal of Robotics and Control (JRC)*, vol. 5, no. 4, Art. no. 4, Jun. 2024, doi: 10.18196/jrc.v5i4.21987.
- [8] Y. S. Handayani and Y. Mardiana, "KENDALI ROBOT BLUETOOTH DENGAN SMARTPHONE ANDROID BERBASIS ARDUINO UNO," *ILKOM Jurnal Ilmiah*, vol. 10, no. 3, Art. no. 3, Dec. 2018, doi: 10.33096/ilkom.v10i3.363.331-337.
- [9] W. Purbowaskito and C.-H. Hsu, "Sistem Kendali PID untuk Pengendalian Kecepatan Motor Penggerak Unmanned Ground Vehicle untuk Aplikasi Industri Pertanian," *JURNAL INFOTEL*, vol. 9, no. 4, Art. no. 4, Nov. 2017, doi: 10.20895/infotel.v9i4.253.
- [10] M. Lamatenggo, I. Wiranto, and W. Ridwan, "Perancangan Balancing Robot Beroda Dua Dengan Metode Pengendali PID Berbasis Arduino Nano," *Jambura Journal of Electrical and Electronics Engineering*, vol. 2, no. 2, Art. no. 2, Jul. 2020, doi: 10.37905/jjee.v2i2.6906.
- [11] F. F. W. Amaral, J. Subur, and S. Suryadhi, "Implementasi Metode PID dan Inverse Kinematics pada Pergerakan Robot Berkaki Enam (Hexapod)," *SinarFe7*, vol. 6, no. 1, Art. no. 1, Sep. 2024.
- [12] "Arduino Uno Rev3," Arduino Official Store. Accessed: Apr. 14, 2025. [Online]. Available: <https://store.arduino.cc/products/arduino-uno-rev3>
- [13] "ATMEGA328P." Accessed: Apr. 14, 2025. [Online]. Available: <https://www.microchip.com/en-us/product/atmega328p>
- [14] Bharat, "From Healthcare to Automotive: Exploring OpenCV Applications in 2025," OpenCV. Accessed: Apr. 15, 2025. [Online]. Available: <https://opencv.org/blog/opencv-applications-in-2023/>
- [15] A. A. M. Suradi, M. F. Rasyid, M. Mushaf, and M. Rizal, "Deteksi Tingkat Kematangan Buah Apel Menggunakan Segmentasi Ruang Warna HSV," *SISITI: Seminar Ilmiah Sistem Informasi dan Teknologi Informasi*, vol. 12, no. 1, Art. no. 1, Feb. 2023, doi: 10.36774/sisiti.v12i1.1163.
- [16] S. Jardim, J. António, and C. Mora, "Image thresholding approaches for medical image segmentation - short literature review," *Procedia Computer Science*, vol. 219, pp. 1485–1492, Jan. 2023, doi: 10.1016/j.procs.2023.01.439.
- [17] R. P. Borase, D. K. Maghade, S. Y. Sondkar, and S. N. Pawar, "A review of PID control, tuning methods and applications," *Int. J. Dynam. Control*, vol. 9, no. 2, pp. 818–827, Jun. 2021, doi: 10.1007/s40435-020-00665-4.