

**SISTEM KENDALI AUTONOMOUS
COLOR DETECTION BERBASIS COMPUTER VISION
(STUDI KASUS KONTES ROBOT ABU INDONESIA 2024)**

TUGAS AKHIR

Diajukan guna Memenuhi Persyaratan untuk Mencapai Derajat Strata-1

Program Studi Teknik Elektro Fakultas Teknik

Universitas Muhammadiyah Yogyakarta



Disusun oleh:

SATRIA GIRI KUSUMA

20210120024

**PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH YOGYAKARTA
2025**

HALAMAN JUDUL
SISTEM KENDALI AUTONOMOUS
COLOR DETECTION BERBASIS COMPUTER VISION
(STUDI KASUS KONTES ROBOT ABU INDONESIA 2024)

Diajukan guna Memenuhi Persyaratan untuk Mencapai Derajat Strata-1
Program Studi Teknik Elektro Fakultas Teknik
Universitas Muhammadiyah Yogyakarta



PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH YOGYAKARTA
2025

HALAMAN PENGESAHAN I
SISTEM KENDALI AUTONOMOUS
COLOR DETECTION BERBASIS COMPUTER VISION
(STUDI KASUS KONTES ROBOT ABU INDONESIA 2024)

Disusun oleh:

Satria Giri Kusuma

NIM: 20210120024

Telah diperiksa dan disetujui:

Dosen Pembimbing



Muhamad Yusvin Mustar, S.T., M.Eng.

NIK. 19880508201504 123 073

Mengetahui,

Ketua Program Studi Teknik Elektro

Universitas Muhammadiyah Yogyakarta



Ir. Karisma Trinanda Putra, S.ST., M.T., Ph.D.

NIK. 19900619201604 123 092

HALAMAN PENGESAHAN II
SISTEM KENDALI AUTONOMOUS
COLOR DETECTION BERBASIS COMPUTER VISION
(STUDI KASUS KONTES ROBOT ABU INDONESIA 2024)

Disusun oleh:

Satria Giri Kusuma

NIM: 20210120024

Telah dipertahankan didepan dewan penguji
pada tanggal 24 Mei 2025

Dosen Pembimbing



Muhamad Yusvin Mustar, S.T., M.Eng.

NIK. 19880508201504 123 073

Dosen Penguji,



Kunnu Purwanto, S.T., M.Eng.

NIK. 19830919201710 123 103

Mengesahkan,

Ketua Program Studi Teknik Elektro

Universitas Muhammadiyah Yogyakarta



Ir. Karisma Rinanda Putra, S.ST., M.T., Ph.D.

NIK. 19900619201604 123 092

HALAMAN PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : Satria Giri Kusuma

NIM : 20210120024

Program Studi : Teknik Elektro

Fakultas : Teknik

Universitas : Universitas Muhammadiyah Yogyakarta

Menyatakan bahwa sesungguhnya Tugas Akhir dengan judul “Sistem Kendali *Autonomous Color Detection* Berbasis *Computer Vision* (Studi Kasus Kontes Robot ABU Indonesia 2024)” merupakan benar hasil karya saya sendiri dan belum diajukan dalam bentuk apapun kepada perguruan tinggi manapun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian Tugas Akhir ini.

Yogyakarta, 24 Mei 2025

Yang menyatakan,

Satria Giri Kusuma

20210120024

HALAMAN PERSEMBAHAN

Puji dan syukur saya panjatkan kepada Allah SWT, Tuhan Yang Maha Pengasih lagi Maha Penyayang. Berkat rahmat, hidayah, dan karunia-Nya, saya dapat menyelesaikan Tugas Akhir ini dengan baik. Semoga pencapaian ini menjadi awal dari kontribusi nyata saya dalam menerapkan ilmu yang telah saya pelajari kepada dunia kerja dan masyarakat luas demi mewujudkan cita-cita yang telah saya impikan.

Karya tulis ini saya persembahkan kepada Ibu dan Bapak saya sebagai rasa terimakasih atas semua dukungan dan doa yang diberikan untuk saya dalam menjalani pendidikan. Semoga karya ini dapat bermanfaat bagi semuanya.

Saya mengucapkan terimakasih kepada dosen pembimbing saya Bapak Muhamad Yusvin Mustar, S.T., M.Eng., yang telah bersedia membimbing saya selama proses penulisan karya ini. Semoga ilmu dan bimbingan yang diberikan kepada saya menjadi buah baik untuk bapak.

MOTTO

"Barangsiapa yang menempuh jalan untuk menuntut ilmu,
Allah akan memudahkan baginya jalan menuju surga."

(HR. Muslim)

"Sesungguhnya bersama kesulitan ada kemudahan."

(QS. Al-Insyirah: 6)

“Teruslah melangkah untuk menjadi lebih baik
dalam berproses mencapai semua yang dicita-citakan”

“Sesungguhnya Allah tidak akan mengubah keadaan suatu kaum
sebelum mereka mengubah keadaan diri mereka sendiri”

(QS. Ar Ra’d: 11)

KATA PENGANTAR

Alhamdulillah rabbi *‘aalamiin*, puji dan syukur kehadiran Allah SWT atas berkat Rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul “Sistem Kendali *Autonomous Color Detection* Berbasis *Computer Vision* (Studi Kasus Kontes Robot ABU Indonesia 2024)”. Penulisan tugas akhir ini sebagai salah satu syarat untuk meraih gelar Sarjana Teknik di Universitas Muhammadiyah Yogyakarta.

Dalam Menyelesaikan Tugas Akhir ini penulis banyak memperoleh dukungan, bimbingan dan bantuan dari berbagai pihak. Oleh karena itu, penulisan menyampaikan ucapan terimakasih yang sebesar-besarnya kepada:

1. Orang tua saya, Ibu Diah Ratnaningsih dan Bapak Suropto, serta Indra Wicaksono sebagai kakak dan Halim Bagus Al Hakim sebagai adik penulis, yang selalu memanjatkan doa terbaik, memberikan dukungan, nasehat, perhatian, dan semangat yang tidak pernah berhenti kepada penulis sehingga penulis bisa menyelesaikan tugas akhir ini.
2. Bapak Karisma Trinanda Putra, S.ST., M.T., Ph.D., selaku Ketua Program Studi Teknik Elektro Universitas Muhammadiyah Yogyakarta.
3. Bapak Muhammad Yusvin Mustar, S.T., M.Eng., selaku Dosen Pembimbing yang telah banyak memberikan arahan dan masukkan kepada penulis dalam penyusunan tugas akhir ini.
4. Teman-teman seperjuangan, Muhammad Afif Fawwaz Rizqullah, Muhammad Hisyam, dan Mohammad Rizqy Arransyah, yang seringkali mengingatkan semasa kuliah dan membantu dalam pengambilan data.
5. Seluruh teman-teman Tim MR DART dan mentor Mas Bintang yang telah membantu saya membangun tim dari awal hingga membantu dalam membangun robot.
6. Teman-teman Tim MR LEBA yang telah membantu dan mendukung penulis dalam menulis tugas akhir ini.
7. Seluruh teman-teman anggota Microcontroller and Robotic Club serta pihak yang tidak dapat penulis sebutkan satu persatu yang telah memberikan

dukungan, referensi, masukan, saran, pengalaman, dan ilmu pengetahuannya sehingga penulis dapat melakukan penelitian pada tugas akhir ini.

Berbagai upaya dalam menyelesaikan tugas akhir ini telah dilakukan oleh penulis. Apabila terdapat kekurangan pada penulisan tugas ini, penulis memohon maaf.atas kesalahan-kesalahan tersebut. Penulis berharap tugas akhir ini dapat memberikan manfaat bagi penulis dan pembaca.

Yogyakarta, 24 Mei 2025

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	ii
HALAMAN PENGESAHAN I	iii
HALAMAN PENGESAHAN II	iv
HALAMAN PERNYATAAN	v
HALAMAN PERSEMBAHAN	vi
MOTTO.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xiv
INTISARI.....	xv
ABSTRACT.....	xvi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	2
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Penelitian	3
1.6. Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI	5
2.1. Tinjauan Pustaka	5
2.2. Dasar Teori	11
2.2.1. <i>Image Processing</i> (Pengolahan Citra).....	11
2.2.2. Warna BGR dan HSV Citra Digital	12

2.2.3.	<i>Thresholding</i>	13
2.2.4.	Kontrol PID.....	14
2.2.5.	Bahasa Pemrograman Python	15
2.2.6.	Library OpenCV.....	16
2.2.7.	Webcam.....	17
2.2.8.	Arduino Mega	18
2.2.9.	Arduino IDE.....	19
2.2.10.	Module Driver Motor BTS 7960	20
2.2.11.	Motor PG45.....	20
2.2.12.	Baterai LiPo	21
BAB III METODE PENELITIAN.....		23
3.1.	Metode Perancangan	23
3.2.	Waktu dan Tempat Penelitian.....	25
3.3.	Objek Penelitian	25
3.4.	Analisis Kebutuhan	26
3.4.1.	Kebutuhan Rancangan Sistem.....	26
3.4.2.	Kebutuhan Perangkat Keras	26
3.4.3.	Kebutuhan Perangkat Lunak	28
3.5.	Perancangan Sistem	29
3.5.1.	Perancangan Perangkat Keras	30
3.5.2.	Perancangan Perangkat Lunak	31
3.6.	Pengujian Sistem.....	34
3.6.1.	Pengujian Sistem Deteksi Bola Berwarna.....	34
3.6.2.	Pengujian Simulasi Pembacaan Posisi Objek Terdeteksi.....	35
3.6.3.	Pengujian Akurasi Deteksi Bola Berwarna	36

3.6.4. Pengujian Dua Sistem Terintegrasi	36
BAB IV HASIL DAN ANALISIS	38
4.1. Hasil Rancangan Sistem pada Robot ABU	38
4.2. Hasil <i>Graphic User Interface</i>	38
4.3. Hasil Pengujian Sistem Deteksi Bola Berwarna	39
4.4. Hasil Pengujian Simulasi Pembacaan Posisi Objek Bola Biru	40
4.5. Hasil Pengujian Akurasi Sistem Deteksi	40
4.6. Hasil Pengujian Dua Sistem Terintegrasi	41
BAB V KESIMPULAN DAN SARAN	45
5.1. Kesimpulan	45
5.2. Saran	46
DAFTAR PUSTAKA	47
LAMPIRAN	50

DAFTAR GAMBAR

Gambar 2. 1. Pengolahan Citra	12
Gambar 2. 2. HSV Warna.....	13
Gambar 2. 3. Thresholding.....	14
Gambar 2. 4. Diagram Block PID	15
Gambar 2. 5. Python	16
Gambar 2. 6. OpenCV.....	17
Gambar 2. 7. Webcam Logitech C922	18
Gambar 2. 8. Arduino Mega 2560.....	18
Gambar 2. 9. Arduino IDE.	19
Gambar 2. 10. Modul BTS7960.....	20
Gambar 2. 11. Motor PG45	21
Gambar 2. 12. Baterai Lipo.....	22
Gambar 3. 1. Diagram Alir Penelitian.....	23
Gambar 3. 2. Diagram Block Full Sistem.....	29
Gambar 3. 3. Rangkaian Sistem Perangkat Keras.....	31
Gambar 3. 4. Diagram Alir Sistem Deteksi Bola Berwarna	32
Gambar 3. 5. Diagram Alir Sistem Kontrol PID.....	33
Gambar 3. 6. Skema Pengujian Deteksi Bola Berwarna.....	35
Gambar 3. 7. Skema Pengujian Pembacaan Nilai Error	35
Gambar 3. 8. Skema Pengujian Akurasi Deteksi Bola Berwarna	36
Gambar 3. 9. Skema Dua Sistem Terintegrasi	37
Gambar 4. 1. Rancangan Robot ABU	38
Gambar 4. 2. Peletakan Laptop Pada Robot ABU	38
Gambar 4. 3. Hasil Graphic User Interface.....	39
Gambar 4. 4. Hasil Pengujian Sistem Deteksi Bola Berwarna	39
Gambar 4. 5. Hasil Pengujian Simulasi Pembacaan Nilai Error.....	40
Gambar 4. 9. Grafik Pengujian Konstanta Proposional	43

DAFTAR TABEL

Tabel 2. 1. Tinjauan Pustaka	8
Tabel 4. 2. Hasil Pengujian Akurasi Sistem Deteksi	41
Tabel 4. 3. Pengujian Konstanta Proposional.....	42

INTISARI

Penelitian ini membahas perancangan sistem kendali *autonomous* berbasis deteksi warna menggunakan *computer vision*, yang diterapkan pada studi kasus Kontes Robot ABU Indonesia (KRAI) 2024. Tujuannya adalah merancang sistem yang memungkinkan robot bergerak secara autonom menuju objek bola berwarna yang terdeteksi oleh sistem komputer. Sistem terdiri dari dua metode, yaitu *color tracking* dan kontrol PID. Keluaran dari proses color tracking berupa posisi koordinat objek dalam satuan pixel citra, yang digunakan dalam perhitungan kontrol PID untuk menghasilkan nilai *PWM* sebagai pengendali kecepatan dan arah putaran motor. Hasil pengujian menunjukkan bahwa sistem color tracking berhasil mendeteksi objek bola berwarna dengan akurasi hingga 100% pada kondisi pencahayaan tanpa ada gangguan *backlight* dan pada jarak dekat 1 meter hingga jarak maksimum arena kontes 6 meter. Pada pengujian kontrol PID, konstanta $K_p = 0,6$; $K_i = 0,0$; dan $K_d = 0,0$ menghasilkan pergerakan robot yang responsif. Nilai K_p yang terlalu kecil menyebabkan respon lambat, sementara nilai K_p yang terlalu besar menyebabkan *overshoot*. Penelitian ini membuktikan bahwa integrasi deteksi warna berbasis *computer vision* dengan kendali PID dapat menghasilkan sistem kendali *autonomous* yang andal dan responsif untuk aplikasi robotika.

Kata kunci: KRAI, computer vision, color tracking, kontrol PID, otonom.

ABSTRACT

This research discusses the design of an autonomous control system based on color detection using computer vision, applied to the case study of the 2024 ABU Robocon Indonesia Contest (KRAI). The goal is to design a system that allows a robot to autonomously move toward a colored ball detected by the computer system. The system consists of two main methods: color tracking and PID control. The output of the color tracking process is the coordinate position of the object in image pixels, which is used in PID control calculations to generate PWM values that control the speed and direction of the motor's rotation. Test results show that the color tracking system successfully detects colored ball objects with up to 100% accuracy under lighting conditions without backlight interference and within a range of 1 meter to the contest arena's maximum distance of 6 meters. In PID control testing, the constants $K_p = 0.6$, $K_i = 0.0$, and $K_d = 0.0$ resulted in responsive robot movements. A K_p value that is too low causes slow response, while a value that is too high leads to overshooting. This study demonstrates that the integration of color detection based on computer vision with PID control can produce a reliable and responsive autonomous control system for robotics applications.

Keywords: KRAI, computer vision, color tracking, PID control, autonomous.

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kontes Robot ABU Indonesia (KRAI) merupakan kontes robot yang diadakan oleh Pusat Prestasi Nasional (Puspresnas) dibawah naungan Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi (Kemendikbudristek) pada ajang Kontes Robot Indonesia. Kontes ini mengacu pada tema dan pedoman dari ajang *ABU Asia-Pacific Robot Contest* (ABUROBOCON) yang setiap tahun temanya berbeda-beda. Contoh tema kontes beberapa tahun terakhir yaitu “robot pemanah” pada tahun 2021, *lagori* (permainan tradisional India) pada tahun 2022, dan “Melempar Bunga di Atas Angkor Wat” pada tahun 2023. Mengacu pada tema dan pedoman *ABU Asia Pacific Robot Contest 2024* yang akan dilaksanakan di *Dai Yen Multi-Purpose Gymnasium, Quảng Ninh, Vietnam*, Kontes Robot ABU Indonesia (KRAI) tahun 2024 bertemakan “Hari Panen” (Benyamin Kusumoputro, 2023).

Tugas dari tema ini yaitu membangun dua buah robot yang dapat melakukan simulasi pertanian dimulai dengan menanam bibit berupa tanaman padi hingga memasukan hasil panen kedalam keranjang. Sistem pengendali dari robot dalam menjalankan misi dibedakan sesuai tugas yang dikerjakan. Sistem pengendali pada robot satu (robot pertama) dapat secara manual, *semi-auto*, ataupun *full-auto*. Sedangkan sistem pengendali pada robot dua dikendalikan secara *autonomous* atau berjalan sendiri sesuai dengan algoritma dan pembacaan sensor yang digunakan. Tugas dari robot dua atau robot *autonomous* yaitu mencari properti simulasi berupa bola berwarna sebagai pengganti padi hasil panen yang harus dimasukan kedalam keranjang.

Tugas robot dua mengharuskan untuk membangun sistem kendali *otonomous* yang dapat melakukan deteksi bola yang letaknya tidak beraturan. Oleh karena itu, perlu dirancang sistem yang dapat membuat robot mengetahui posisi bola terhadap robot. Selain itu, robot harus dapat mengambil bola menggunakan pencapit yang ada didepan robot. Untuk mengatasi masalah tersebut, sistem pembacaan

posisi bola perlu diintegrasikan dengan sistem yang berfungsi untuk menggerakkan robot kearah bola.

Dalam menyelesaikan permasalahan tersebut, penelitian ini akan membuat sistem kendali yang dapat mengendalikan robot secara *autonomous* untuk mencari dan mengarahkan robot mendekati bola. Salah satu metode untuk melakukan deteksi objek yaitu dengan memanfaatkan warna dari objek dengan menggunakan deteksi warna RGB (Red Green Blue) sebagai warna dasar dalam deteksi objek berwarna (Ahadi, 2024). Metode deteksi objek ini dapat dilakukan dengan pengolahan citra menggunakan bahasa pemrograman Python dan menggunakan *library* OpenCV. Sedangkan untuk metode pengendalian robot diperlukan sebuah kendali untuk dapat mengatur kecepatan setiap motor sesuai, sehingga dapat mengarahkan robot kearah dan posisi yang diinginkan. Kendali kecepatan motor pada robot roda omni dapat dirancang menggunakan metode kontrol Proporsional-Integral-Derivatif (PID) (Wahab et al., 2021). Diangkatnya dua metode tersebut pada penelitian ini bertujuan untuk membangun sistem kendali *autonomous* robot dengan mengintegrasikan kedua metode tersebut. Melalui penelitian ini, diharapkan dapat terbangun sistem kendali yang dapat diimplementasikan pada robot ABU ataupun robot yang bertugas dalam mencari dan mengambil objek.

1.2. Rumusan Masalah

Berikut merupakan rumusan masalah yang didasarkan dari latar belakang:

1. Apakah melakukan pembacaan warna bola menggunakan *computer vision* dapat mengenali bola.
2. Apakah sistem deteksi bola yang dibuat dapat mengetahui posisi bola terhadap robot.
3. Bagaimana responsivitas pergerakan robot terhadap bola dengan penggunaan kontrol PID yang terintegrasi dengan sistem deteksi bola.

1.3. Batasan Masalah

Agar terhindar dari pembahasan yang melebar dan tidak terfokus, maka berikut batasan masalah dalam penelitian ini:

1. Pada penelitian ini hanya membahas mengenai robot *autonomous* pada robot ABU MR DART dalam melakukan pencarian bola.
2. Penelitian ini hanya membahas integrasi sistem mencari bola dan sistem kontrol PID serta hasil responsivitas pergerakan robot terhadap posisi bola yang dideteksi.
3. Sistem deteksi bola dibuat dengan menggunakan bahasa pemrograman Python dan menggunakan *library* OpenCV dalam mengolah citra.
4. Sistem deteksi bola hanya dengan metode pembacaan warna dalam RGB (Red Green Blue) dan HSV (Hue Saturation Value) pada pembacaan *wabcam*.
5. Jarak bola terjauh yang dideteksi tidak lebih jauh dari lebar arena kontes, yaitu sejauh 6 meter dan berada pada jangkauan pembacaan webcam.

1.4. Tujuan Penelitian

Adapun tujuan dari penelitian ini antara lain:

1. Mengetahui apakah penggunaan *computer vision* terhadap warna bola dapat mengenali bola.
2. Mengetahui sistem deteksi bola dapat mengetahui posisi bola terhadap robot.
3. Mengetahui seberapa responsif penggunaan kontrol PID yang terintegrasi dengan sistem deteksi bola terhadap pergerakan robot.

1.5. Manfaat Penelitian

Manfaat dari hasil penelitian ini antara lain:

1. Pengembangan teknologi *autonomous* pada robot MR DART.
2. Sistem dapat diimplementasikan pada robot MR DART dalam mengikuti Kontes Robot ABU Indonesia.

1.6. Sistematika Penulisan

Penulisan tugas akhir ini terdiri dari lima bab yang masing-masing dijelaskan sebagai berikut:

1. BAB I PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang dilakukan penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat, serta sistem penulisan.

2. BAB II TINJAUAN PUSTAKA DAN DASAR TEORI

Dalam bab ini berisi mengenai sumber materi dan referensi yang menjadi landasan dalam pembahasan dari tugas akhir yang berkaitan tentang sistem, perhitungan, dan analisis.

3. BAB III METODE PENELITIAN

Bab ini menjelaskan tentang metode yang akan digunakan untuk mendapatkan data pada penelitian untuk digunakan pada analisis dan pembahasan.

4. BAB IV ANALISIS DAN PEMBAHASAN

Bab ini berisi hasil dari penelitian berupa hasil kemampuan deteksi objek, hingga responsivitas pergerakan robot terhadap posisi bola.

5. BAB V PENUTUP

Bab penutup berisikan kesimpulan dan saran dari seluruh pembahasan dan analisis yang dilakukan.

6. DAFTAR PUSTAKA

Bagian ini berisi mengenai sumber referesnsi yang digunakan pada penulisan tugas akhir.

7. LAMPIRAN

Pada bagian terakhir ini mencantumkan informasi tambahan yang mendukung tugas akhir seperti hasil pengujian, tabel data, foto, skema, maupun dokumen lainnya.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

Dalam perancangan tugas akhir ini, agar dapat berhasil dan mendapatkan hasil yang baik maka diperlukan beberapa perbandingan dan referensi yang digunakan pada penelitian ini. Sumber referensi yang digunakan pada penelitian ini menggunakan beberapa penelitian terdahulu yang memiliki kaitan dengan topik yang dibahas pada penelitian ini.

Fahmi Erza, Hurriyatul Fitriyah, dan Eko Setiawan (2022) melakukan penelitian dengan judul “Sistem *Object Tracking* pada *Quadcopter* Menggunakan Segmentasi Citra dengan Deteksi Warna HSV dan Metode *Regresi Linier* Berbasis Raspberry Pi”. Penelitian ini membahas pengembangan sistem pengawasan menggunakan deteksi objek yang diimplementasikan pada sebuah teknologi *Quadcopter*. Tujuan dari penelitian ini yaitu membuat sistem yang dapat mendeteksi objek dan mengikutinya. Sistem yang dikembangkan memanfaatkan atribut berwarna untuk menjadikan acuan *threshold* segmentasi warna untuk dideteksi. Penerapan sistem ini pada drone mempengaruhi pergerakan drone dengan hasil 27 pengujian didapatkan akurasi pergerakan *pitch* sebesar 84%, pergerakan *yaw* sebesar 94%, dan pergerakan *up/down* sebesar 91,5%, serta waktu komputasinya adalah 0.175829662 detik per frame (Fahmi Erza, 2022).

Indah Sulistiyowati, Hafidz Maulana Ichsan, dan Izza Anshory (2024) melakukan penelitian dengan judul “Konveyor Penyortir Objek dengan Deteksi Warna Menggunakan Kamera ESP-32 Berbasis Open-CV Python”. Penelitian yang dibahas mengenai pengembangan alat penyortir benda yang diaplikasikan pada konveyor. Tujuan pengembangan ini adalah untuk mempermudah penyortiran benda dengan memanfaatkan pendeteksian warna. Pengembangan alat penyortir benda ini menggunakan metode *library* OpenCV dalam mengolah citra dengan langkah awal menangkap citra dalam RGB (Red, Green, dan Blue). Penelitian ini menghasilkan kemampuan untuk membedakan objek dengan berdasarkan warna

RGB dengan mempertimbangkan nilai HSV (Hue, Saturation, dan Value) pada citra (Sulistiyowati et al., 2024).

Andy Hartono, Hurriyatul Fitriyah, Edita Rosana Widasari (2023) melakukan penelitian dengan judul “Sistem Deteksi Tomat Matang Hidroponik Berdasarkan Warna Hue, Saturation, dan Value Menggunakan Metode *Threshold* Berbasis *Graphical User Interface*”. Penelitian ini melakukan pengembangan dalam mendeteksi tanaman tomat hidroponik dengan menggunakan fitur warna HSV (Hue, Saturatuin, dan Value) dan menampilkan hasil deteksi melalui *GUI* (*Graphical User Interface*) yang dibuat menggunakan aplikasi *Qt Designer* berbasis python sebagai bahasa pemrograman untuk memproses citra digital menggunakan *library* PyQt5 dan OpenCV. Selain itu, sistem yang dikembangkan untuk dapat mengukur dan menentukan jarak dan tinggi tanaman pada webcam. Hasil pengujian yang dilakukan pada penelitian ini dengan melakukan 20 kali pengujian terhadap 10 tanaman tomat matang dan 10 tanaman tomat tidak matang, diperoleh keseluruhan tanaman tomat dapat terdeteksi dengan sesuai kategori matan dan tidak matang hingga presentase 100% dengan rata-rata waktu komputasi sebesar 3,893 detik (Hartono et al., 2023).

Ahmad Ersam Nur Ramadhan, Widyadi Setiawan, dan Duman Care Khrisne (2023) melakukan penelitian dengan judul “Rancang Bangun Deteksi Objek dengan Metode Filter Warna HSV pada Sistem Klasifikasi Kualitas Biji Kopi Berbasis NVIDIA Jetson Nano”. Penelitian yang dilakukan yaitu merancang sistem deteksi objek yang penerapannya sebagai penunjang pirwarupa sistem klasifikasi kualitas biji kopi dengan berbasis NVIDIA Jetson Nano. Sistem yang dirancang menggunakan metode filter warna Hue Saturation value (HSV) untuk mendeteksi adanya objek pada frame citra. Sistem deteksi objek direalisasikan menggunakan webcam menghasilkan identifikasi adanya objek biji kopi dengan efektif, tetapi terdapat keterbatasan berupa kurang efisien dalam mendeteksi biji kopi diluar ambang batas warna HSV yang telah ditentukan, seperti warna hitam dari biji kopi dan ukuran biji kopi yang terlalu kecil, serta adanya *jello effect* yang diakibatkan

spesifikasi webcam kurang cocok dalam menangkap objek bergerak (Ramadhan et al., 2023).

Mukhammad Bayu Riswan Maulana, Denda Dewatama, dan Mila Fauziyah (2024) melakukan penelitian dengan judul “Kontrol Kecepatan Motor Induksi 3 Fasa Berbasis Arduino Mega 2560 pada Lift 4 Lantai”. Penelitian yang dilakukan yaitu melakukan pengujian sistem untuk mempertahankan posisi dan kestabilan motor lift dengan menggunakan metode PID (Proportional Integral Derivative). Komponen lainnya yang digunakan pada penelitian ini yaitu sensor *rotary encoder* untuk melakukan pembacaan putaran motor, serta komponen rangkaian PWM *analog converter* untuk mengubah sinyal PWM menjadi tegangan analog yang mengatur perubahan frekuensi VSD (*Variable Speed Drive*). Pengujian dilakukan dengan mengukur besarnya kecepatan pergerakan lift yang menggunakan motor induksi 3 fasa ketika bergerak dari semua lantai menuju lantai lainnya. Setelah dilakukan pengujian didapat hasil putaran motor induksi 3 fasa terkendali dengan nilai parameter $K_p = 0,3$; $K_i = 0,3$; dan $K_d = 0,9$ (Maulana et al., 2024).

Fadhil Ekadipa, Beauty Anggraheny Ikawanty, dan Mohammad Luqman (2024) melakukan penelitian dengan judul “Sistem Kendali Kecepatan Motor DC untuk Alat Pemeras Madu Menggunakan Metode PID”. Penelitian yang dilakukan yaitu merancang alat ekstraktor madu otomatis yang cara kerjanya dengan memutar sarang madu pada sebuah keranjang menggunakan motor DC sebagai pemutar keranjang. Tujuan dari alat ini untuk mengembangkan metode ekstraktor madu manual yang diputar dengan menggunakan tangan menjadi otomatis menggunakan motor, sehingga kestabilan putaran dapat terjaga agar tidak terjadi kerusakan sarang madu. Alat yang dirancang memanfaatkan sistem kendali PID untuk mengontrol kecepatan motor pada *setpoint* 50 rpm. Hasil pengujian pada penelitian ini didapatkan bahwa sistem dapat mempertahankan kecepatan dengan baik dan menghasilkan *steady-state error* sebesar 1,4% dan *maximum overshoot* sebesar 8% (Ekadipa et al., 2024).

Muhamad Yusvin Mustar, Rudy Hartanto, dan Paulus Insap Santosa (2024) melakukan penelitian dengan judul “*Exploring Attentive User Interface Input via*

Raspberry Pi, based on Face Landmark Detection, Eye Open-Closed Detection and Head Movements Detection”. Penelitian yang dilakukan berfokus ada pengembangan *Attention User Interface* berbasis *vision based* atau *computer vision* yang diterapkan pada platform Raspberry Pi. Tujuan utama penelitian adalah mengeksplorasi dan merancang AUI menggunakan pendeteksian *landmark*, deteksi mata terbuka dan tertutup serta deteksi pergerakan kepala melalui Raspberry Pi. Penelitian ini membahas tiga skenario pemodelan input AUI menggunakan berbagai metode untuk mengidentifikasi desain yang paling efektif melalui Raspberry Pi 4. Berdasarkan hasil eksperimen dan serangkaian tugas yang diuji, memberikan hasil terbaik dengan nilai FPS rata-rata 10 dan akurasi 100% untuk deteksi *landmark* wajah, deteksi kondisi mata terbuka, serta deteksi pandangan ke depan, atas, bawah, kanan, dan kiri pada berbagai sudut deteksi. Pengujian tingkat akurasi yang dilakukan dengan membagi jumlah frame terdeteksi dengan total jumlah frame dan dikalikan dengan 100 persen (Mustar et al., 2024).

Tabel 2. 1. Tinjauan Pustaka

Peneliti (Tahun)	Judul Penelitian	Metode	Hasil
Fahmi Erza, Hurriyatul Fitriyah, dan Eko Setiawan (2022)	Sistem <i>Object Tracking</i> pada <i>Quadcopter</i> Menggunakan Segmentasi Citra dengan Deteksi Warna HSV dan Metode <i>Regresi Linier</i> Berbasis Raspberry Pi	<i>Regresi linear thresholding</i>	Hasil 27 pengujian didapatkan akurasi pergerakan <i>pitch</i> sebesar 84%, pergerakan <i>yaw</i> sebesar 94%, dan pergerakan <i>up/down</i> sebesar 91,5%, serta waktu komputasinya adalah 0.175829662 detik per frame
Indah Sulistiyowati, Hafidz Maulana	Konveyor Penyortir Objek dengan Deteksi Warna	OpenCV	Kemampuan untuk membedakan objek dengan berdasarkan

Ichsan, dan Izza Anshory (2024)	Menggunakan Kamera ESP-32 Berbasis Open-CV Python		warna RGB dengan mempertimbangkan nilai HSV (Hue, Saturation, dan Value) pada citra
Andy Hartono, Hurriyatul Fitriyah, Edita Rosana Widasari (2023)	Sistem Deteksi Tomat Matang Hidroponik Berdasarkan Warna Hue, Saturation, dan Value Menggunakan Metode Threshold Berbasis Graphical User Interface	HSV	Tanaman tomat dapat terdeteksi dengan sesuai kategori matang dan tidak matang hingga presentase 100% dengan rata-rata waktu komputasi sebesar 3,893 detik
Ahmad Ersam Nur Ramadhan, Widyadi Setiawan, dan Duman Care Khrisne (2023)	Rancang Bangun Deteksi Objek dengan Metode Filter Warna HSV pada Sistem Klasifikasi Kualitas Biji Kopi Berbasis NVIDIA Jetson Nano	Filter HSV	Identifikasi adanya objek biji kopi dengan efektif, tetapi terdapat keterbatasan berupa kurang efisien dalam mendeteksi biji kopi diluar ambang batas warna HSV yang sudah ditentukan, seperti warna biji kopi berwarna kehitaman dan berukuran kecil, serta terdapat <i>jello effect</i> akibat spesifikasi webcam <i>roller shutter</i> tidak

			cocok untuk objek yang bergerak.
Mukhammad Bayu Riswan Maulana, Denda Dewatama, dan Mila Fauziyah (2024)	Rancang Bangun Deteksi Objek dengan Metode Filter Warna HSV pada Sistem Klasifikasi Kualitas Biji Kopi Berbasis NVIDIA Jetson Nano	Proporsional Integral Derivative (PID)	putaran motor induksi 3 fasa terkendali dengan nilai parameter $K_p = 0,3$; $K_i = 0,3$; dan $K_d = 0,9$
Fadhil Ekadipa, Beauty Anggraheny Ikawanty, dan Mohammad Luqman (2024)	Sistem Kendali Kecepatan Motor DC untuk Alat Pemeras Madu Menggunakan Metode PID	PID	Hasil pengujian pada penelitian ini didapatkan bahwa sistem dapat mempertahankan kecepatan dengan baik dan menghasilkan <i>steady-state error</i> sebesar 1,4% dan <i>maximum overshoot</i> sebesar 8%
Muhamad Yusvin Mustar, Rudy Hartanto, dan Paulus Insap Santosa (2024)	<i>Exploring Attentive User Interface Input via Raspberry Pi, based on Face Landmark Detection, Eye Open-Closed Detection and Head Movements Detection</i>	AUI	Memberikan hasil terbaik dengan nilai FPS rata-rata 10 dan akurasi 100% untuk deteksi <i>landmark</i> wajah, deteksi kondisi mata terbuka, serta deteksi pandangan ke depan, atas, bawah,

			kanan, dan kiri pada berbagai sudut deteksi.
--	--	--	--

2.2. Dasar Teori

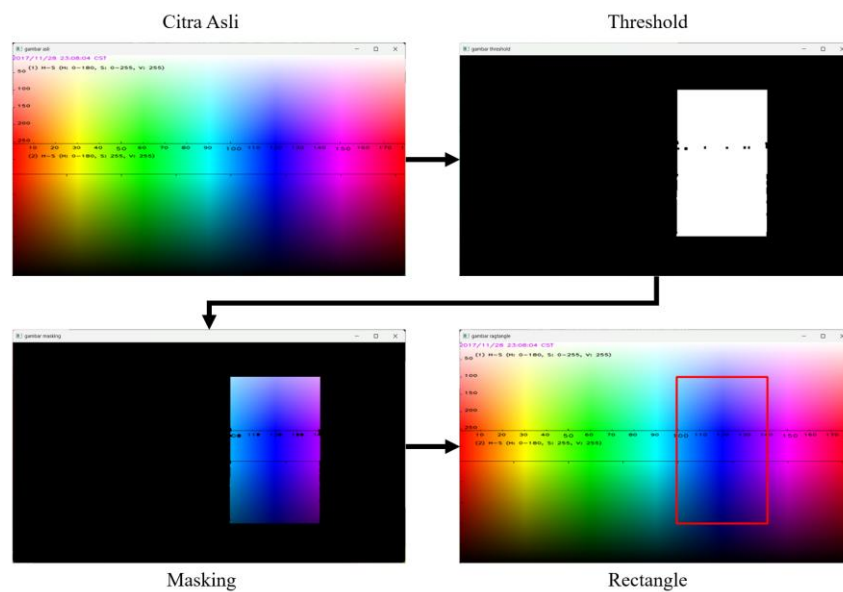
Sub bab ini merupakan bagian penjelasan mengenai berbagai macam komponen, *software*, maupun teori yang akan digunakan dalam merancang dan melakukan penelitian ini. Berbagai komponen dan *software* yang dimaksud sebagai berikut.

2.2.1. *Image Processing* (Pengolahan Citra)

Image processing adalah metode dalam menangkap dan mengolah citra gambar yang tidak bersifat merusak yang mana proses pengambilan data citra dilakukan menggunakan alat bantu penangkap citra. Hasil tangkapan citra yang telah didapat dilakukan pengolahan menggunakan algoritma *image processing* dalam perangkat lunak. Pengolahan citra ini dilakukan dari proses input dalam bentuk gambar yang kemudian dikonversi kedalam bentuk matriks (Mawarni et al., 2023). Pada awal perkembangannya, pengolahan citra dilakukan hanya untuk memperbaiki kualitas dari citra. Namun, sejalan dengan semakin berkembangnya dunia komputasi seperti meningkatnya kecepatan dan kapasitas komputer, kini memungkinkan manusia dapat memperoleh informasi dari suatu citra digital (Apridiansyah & Gumiri, 2021).

Pada penelitian ini, *image processing* digunakan untuk mendapatkan informasi keberadaan suatu objek melalui warna objek yang tertangkap didalam citra. Tahap mendeteksi objek berwarna menggunakan metode ini dimulai dari menginput *raw* data citra dari webcam yang kemudian akan dilakukan pengolahan menggunakan *library* OpenCV pada bahasa pemrograman Python. proses awal dilakukan dengan mengubah format *raw* data citra dari BGR (Blue Green Red) menjadi HSV (Hue Saturation Value), selanjutnya dilakukan *threshold* sesuai batas nilai atas dan nilai bawah dari HSV (Ramadhan et al., 2023). Hasil dari *threshold* tersebut digunakan untuk proses *masking* dan *bounding rectangle* yang ditandai

dengan garis membentuk segi empat yang pada dasarnya terbentuk dari nilai koordinat batas objek dalam panjang lebar pixel citra. Nilai koordinat tersebut terdiri dari x dan y sebagai titik koordinat salah satu sudut kotak, serta *w (weight)* dan *h (high)* sebagai jarak kearah *horizontal* dan *vertical* (Fahmi Erza, 2022). Nilai tersebutlah yang akan digunakan dalam kontrol robot pada penelitian ini. Berikut merupakan gambaran dari proses tahap pengolahan citra yang akan dilakukan.



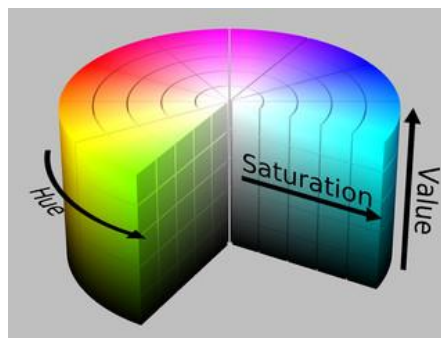
Gambar 2. 1. Pengolahan Citra

2.2.2. Warna BGR dan HSV Citra Digital

HSV merupakan mode warna dari Hue, Saturation, Value yang mempresentasikan warna dari merah, jingga, kuning, biru, ungu, dan hijau dengan tingkat kemurniannya dan nilai Cahaya yang diterima oleh mata, Dimana Hue mempresentasikan warna, Saturation mempresentasikan tingkat kemurnian warna seberapa kuat ataupun pudar suatu warna, dan Value mempresentasikan banyaknya cahaya yang diterima oleh mata atau seberapa cerah maupun gelap warna yang dihasilkan (Zakariya et al., 2020). Pada pengolahan citra dengan *library* OpenCV, nilai warna HSV memiliki skala Hue: 0-179, *Saturation*: 0-255, *Value*: 0-179.

Penggunaan model HSV dalam pengolahan citra memiliki beberapa keuntungan dibandingkan model RGB. Salah satu keuntungan utama adalah

kemampuannya untuk memisahkan informasi warna dari informasi kecerahan. Hal ini sangat berguna dalam aplikasi seperti deteksi objek dan segmentasi citra. Konversi dari model BGR ke HSV memungkinkan pemisahan luma (intensitas gambar) dari jenis warna (informasi warna), yang membuat ruang warna HSV lebih luas dan efektif dalam menghilangkan bayangan dari citra yang ditangkap. Konversi dari model BGR ke HSV merupakan langkah penting dalam pengolahan citra digital. Citra yang ditangkap oleh perangkat seperti webcam biasanya menggunakan format BGR. Oleh karena itu, untuk analisis lebih lanjut menggunakan model HSV, diperlukan proses konversi ini.



Gambar 2. 2. HSV Warna

(Sumber: <https://skillplus.web.id/memahami-color-space/>)

2.2.3. *Thresholding*

Thresholding merupakan teknik pengolahan citra untuk mengubah citra *grayscale* menjadi citra dengan nilai biner. Dalam pengolahan citra untuk melakukan *thresholding* perlu melakukan penentuan batas untuk nilai hue, saturation, dan value untuk memisahkan warna-warna tertentu (Wibowo et al., 2024). Proses *thresholding* meliputi:

- Mengubah citra RGB menjadi HSV: langkah awal ini adalah untuk mengkonversi matriks RGB menjadi nilai matriks HSV.
- Menentukan batas nilai: tahap ini untuk menentukan batasan bawah dan atas dari nilai hue, saturation, dan value sesuai nilai yang ingin ditargetkan.

- c. Menerapkan *threshold*: terakhir menggunakan nilai batas yang sudah ditentukan untuk membuat citra biner, dimana nilai yang ada diluar batasan akan diatur menjadi nilai 1 (putih) dan pixel lainnya menjadi 0 (hitam).



Gambar 2. 3. Thresholding

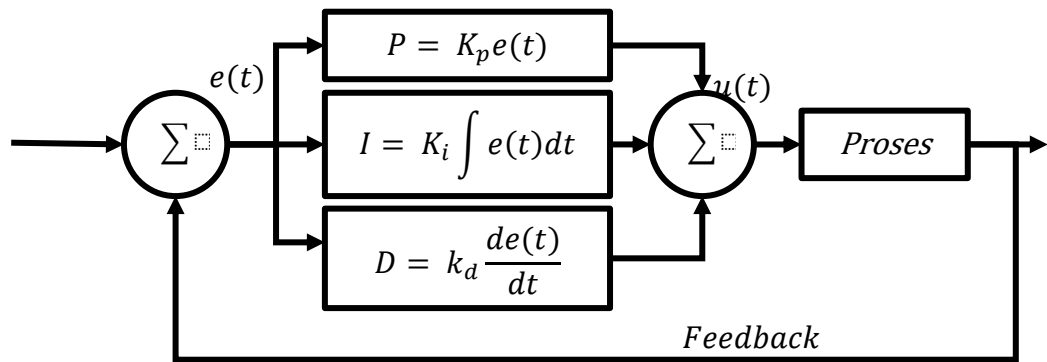
(Sumber: <https://cvexplained.wordpress.com/2020/04/28/color-detection-hsv/>)

Thresholding merupakan metode yang sederhana dan efektif, tetapi ada beberapa permasalahan yang harus dihadapi, seperti perubahan pencahayaan dan bayangan dapat mempengaruhi hasil pengolahan citra ini. Untuk mengatasi masalah ini teknik *processing* seperti penggunaan filter dapat diterapkan sebelum masuk ke pemrosesan *thresholding*. Secara keseluruhan, *thresholding* tetap menjadi salah satu teknik utama dalam pengolahan citra digital karena kesederhanaannya dan efektivitasnya dalam berbagai aplikasi.

2.2.4. Kontrol PID

Kontrol PID merupakan sistem yang digunakan untuk mengatur output dari hasil pembacaan nilai inputnya. Sistem kontrol ini dapat dikatakan sebagai sistem *close loop* karena dalam pengaplikasiannya memerlukan nilai umpan balik (Prasetyo et al., 2022). Sistem kontrol PID banyak digunakan dalam berbagai aplikasi industri, termasuk pengaturan suhu, tekanan, aliran, dan level cairan. Selain itu, sistem ini sering diaplikasikan pada robot yang memerlukan pengendalian roda dengan nilai error yang sekecil mungkin seperti robot *line follower*.

Proses pengendalian dengan sistem ini yaitu dengan melakukan pembacaan nilai error, kemudian dilakukan perhitungan proportional, integral, dan derivative untuk menghasilkan output yang akan digunakan untuk mengubah keadaan robot dalam memperkecil error. Diagram blok dan rumusan matematis dari perhitungan tersebut sebagai berikut.



Gambar 2. 4. Diagram Block PID

$$u(t) = k_p e(t) + K_i \int e(t) dt + k_d \frac{de(t)}{dt}$$

Keterangan:

- k_p : Konstanta Propotional
- k_i : Konstanta Integral
- k_d : Konstanta Derivatife
- e : error
- $u(t)$: Nilai PID terhadap waktu

2.2.5. Bahasa Pemrograman Python

Python merupakan bahasa pemrograman yang mudah dipelajari dan sudah banyak digunakan dalam pengembangan berbagai hal seperti website, analisis data, hingga kecerdasan buatan dan mesin learning (Ua et al., 2023). Python dikenalkan oleh Guido van Rossum pada tahun 1991, Python dirancang untuk memudahkan pengembangan perangkat lunak dengan sintaks yang jelas dan mudah dibaca, sehingga cocok untuk pemula maupun pengembang berpengalaman. Salah satu fitur utama dari Python adalah kemampuannya untuk mendukung berbagai paradigma pemrograman, termasuk pemrograman prosedural, berorientasi objek, dan fungsional.

Salah satu alasan mengapa Python sangat diminati adalah kemudahan pembelajaran. Sintaksisnya yang menyerupai bahasa Inggris membuatnya lebih mudah dipahami dibandingkan dengan banyak bahasa pemrograman lainnya.

Python memungkinkan pengembang untuk menulis program dengan lebih sedikit baris kode dibandingkan dengan bahasa lain, yang meningkatkan produktivitas dan efisiensi pengembangan perangkat. Bahasa pemrograman python menyiapkan berbagai macam *Library* yang dapat menunjang penelitian tugas akhir ini. Python pada penelitian ini berfungsi untuk memprogram sistem deteksi objek berbasis warna dengan pengolahan citra. Proses pengolahan citra menggunakan pada python memanfaatkan *library* OpenCV.



Gambar 2. 5. Python

(Sumber: <https://legacy.python.org/community/logos/>)

2.2.6. Library OpenCV

OpenCV adalah *library* yang berbasis *open source* pada perangkat lunak. OpenCV dirancang untuk melakukan efisiensi komputasi kompleks yang berjalan secara *real time* seperti pengolahan citra yang memungkinkan perangkat komputer dapat melihat dan memahami suatu objek seperti manusia. OpenCV banyak diterapkan pada robot yang memiliki tugas dalam mendeteksi benda secara *real time*. Dalam penerapannya, *library* ini sering digunakan untuk mengenali suatu objek, *face detection*, *face recognition*, *object tracking*, *road tracking* dan lain sebagainya. *Library* ini memiliki kelebihan dalam pengolahan citra yang lengkap dan juga *support* dijalankan diberbagai platform seperti bahasa pemrograman C, java, dan python, serta memiliki fitur untuk mendukung platform kecerdasan buatan. (Arifin, 2022).



Gambar 2. 6. OpenCV

(Sumber: <https://opencv.org/media-kit>)

2.2.7. Webcam

Kamera Webcam, atau "Web Camera," adalah perangkat yang dapat terhubung ke komputer. Webcam Fungsi utama sebagai perangkat yang memungkinkan kita menampilkan video secara *real-time* di komputer. Pada umumnya, semua laptop sudah dilengkapi dengan Webcam untuk mengambil gambar atau digunakan sebagai alat komunikasi langsung dengan video. Cara kerja Webcam mirip dengan kamera digital, yaitu menangkap gambar dengan memanfaatkan cahaya yang masuk melalui lensa. Cahaya tersebut ditangkap oleh detektor mikroskopis yang ada di dalam Webcam. Teknologi yang digunakan dalam pengambilan gambar pada Webcam adalah sensor gambar *Charge-Coupled Device* (CCD) atau CMOS. Hasil tangkapan gambar dari Webcam bisa diolah secara digital atau digunakan untuk penelitian (Nugroho et al., 2019).

Dalam melakukan proses pengolahan citra pada penelitian ini, diperlukan webcam untuk menangkap citra sebagai input data yang digunakan untuk sistem deteksi objek berwarna. Webcam yang akan digunakan yaitu kamera webcam Logitech tipe C922. Penggunaan webcam ini memiliki beberapa alasan, seperti video yang jelas dengan kualitas HD dan jernih dengan resolusi 1080p/30fps. Selain itu, kamera ini memiliki fitur autofocus cepat (hingga 7 cm dari lensa kamera) dan dilengkapi dengan koreksi cahaya. Dengan begitu, hasil tangkapan citra dapat menyesuaikan dengan jarak objek dengan kamera tanpa terjadi buram dan kecerahannya bisa lebih stabil (Dzil et al., 2024).



Gambar 2. 7. Webcam Logitech C922

(Sumber: <https://www.logitech.com/id-id/products/webcams/c922-pro-stream-webcam.960-001090.html>)

2.2.8. Arduino Mega

Arduino mega 2560 merupakan sebuah board yang menggunakan IC mikrokontroler Atmega2560. Board ini dirancang untuk digunakan pada sistem yang memerlukan banyak pin input/output dan memiliki kapasitas penyimpanan yang cukup besar dibandingkan Arduino jenis lain seperti Arduino UNO, Arduino Nano. Penggunaan IC ATmega2560 yang diproduksi oleh Atmel menawarkan berbagai fitur yang membuatnya ideal untuk berbagai aplikasi, termasuk robotika dan otomasi.

Board arduino ini memiliki 54 pin digital I/O yang diantaranya terdiri dari 15 buah pin dapat digunakan sebagai output PWM. Terdapat juga 16 pin analog input, 4 Universal Asynchronous Receiver Transmitter (UART), koneksi USB, osilator kristal 16 MHz, soket ICSP (In-Circuit System Programming), serta tombol reset. Arduino mega berjalan dengan tegangan kerja 5 volt, tetapi memiliki pin tegangan 3,3volt untuk memberi daya sensor (Rahman et al., 2020).



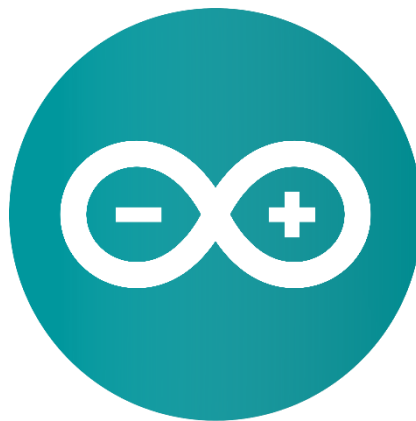
Gambar 2. 8. Arduino Mega 2560

(Sumber: <https://blog.famosastudio.com/2013/09/produk/arduino-mega-2560/531/>)

2.2.9. Arduino IDE

Arduino IDE (*Integrated Development Environment*) adalah platform yang banyak dikenal di kalangan pengembang untuk mengembangkan dan pemrograman proyek berbasis Arduino. IDE ini menyediakan tampilan yang ramah pengguna, sehingga memungkinkan pengguna dari berbagai tingkat keahlian untuk menulis atau memprogram, mengedit, dan mengunggah kode ke papan mikrokontroler Arduino dengan mudah. Salah satu fitur utama dari Arduino IDE adalah *text editor* yang dilengkapi dengan penyorotan sintaksis (*syntax highlighting*) dan *auto-formatting*, yang memudahkan pengguna dalam menulis dan memahami kode.

Salah satu keunggulan dari Arduino IDE adalah kemampuannya untuk mengelola library dengan mudah. Dengan menggunakan menu "Library Manager" yang terintegrasi, pengguna dapat mencari, mengunduh, dan menginstal library tambahan yang diperlukan dalam membuat proyek. Selain itu, Arduino IDE juga menyediakan alat *debugging* seperti *Serial Monitor*, yang memungkinkan pengguna untuk memantau data yang dikirim dan diterima melalui port serial secara *real-time*. Fitur ini sangat membantu dalam proses *debugging* dan pemantauan proyek. Arduino IDE ini dibuat menggunakan bahasa pemrograman java, yang dilengkapi *library* C/C++ (Tyas & Buckhari, 2023).



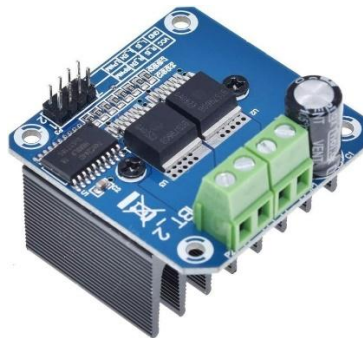
Gambar 2. 9. Arduino IDE.

(Sumber: <https://www.pngwing.com/en/free-png-azewd>)

2.2.10. Module Driver Motor BTS 7960

Driver BTS 7960 adalah driver motor yang sering digunakan dalam mengatur kecepatan dan arah pergerakan dari motor DC (Jayadi & Kurniawan, 2022). Pengendalian arah putaran dan kecepatan motor DC pada driver ini menggunakan IC BTS7960 H-Bridge dengan input sinyal PWM dari mikrokontroler. Penggunaan IC BTS 7960 yang terhubung secara H-bridge membuat module ini dapat mengatur arah putaran motor *secara counter clock wise* ataupun *clock wise* (Sujono et al., 2023).

Driver BTS 7960 memiliki keunggulan utama dalam menangani arus tinggi, dengan kapasitas arus kontinu hingga 43A sehingga dapat digunakan untuk aplikasi daya besar seperti robotika, kendaraan listrik, dan otomasi industri. Modul ini mendukung tegangan kerja 5V hingga 27V dan dilengkapi fitur proteksi seperti *over-current*, *over-temperature*, dan *short-circuit* untuk memastikan keandalan dan keselamatan (Wardana, 2024). Dengan efisiensi tinggi serta *connector* yang mudah digunakan, driver ini kompatibel dengan berbagai mikrokontroler seperti Arduino, STM32, dan ESP32.



Gambar 2. 10. Modul BTS7960

(Sumber: <https://www.ovaga.com/blog/transistor/bts7960-motor-driver-datasheet-and-circuit-dia>)

2.2.11. Motor PG45

Motor PG 45 merupakan jenis motor DC yang sering digunakan dalam robotika maupun otomatisasi sebagai aktuator. Motor DC adalah alat yang berfungsi merubah energi listrik menjadi energi putaran atau gerakan (Jayadi &

Kurniawan, 2022). Motor ini banyak digunakan karena memiliki torsi yang tinggi dengan ukuran motor yang relatif kecil. Motor ini dilengkapi dengan gearbox *planetari* yang meningkatkan torsi keluaran dan memungkinkan pengoperasian pada kecepatan rendah dengan torsi tinggi. *Gearbox* ini juga membantu dalam mengurangi kecepatan putaran motor, sehingga lebih sesuai untuk penggunaan yang memerlukan kontrol presisi.

Salah satu keunggulan dari motor PG 45 adalah kemampuannya untuk beroperasi pada tegangan rendah sambil tetap memberikan torsi yang cukup tinggi. Hal ini memungkinkan penggunaan motor dalam pengaplikasian dengan ruang komponen yang terbatas dan memerlukan efisiensi energi yang baik. Selain itu, dengan adanya encoder, motor ini mampu memberikan umpan balik posisi yang akurat, sehingga meningkatkan kontrol sistem secara keseluruhan. Motor PG 45 juga memiliki beberapa macam spesifikasi yang berbeda-beda sesuai dengan kebutuhan penggunaan, seperti terdapat model dengan *gearbox* dan *encoder* yang berbeda.



Gambar 2. 11. Motor PG45

2.2.12. Baterai LiPo

Baterai LiPo atau Lithium Polymer merupakan jenis baterai yang berupa bentuk lain dari baterai lithium-ion. LiPo tidak menggunakan elektrolit cair seperti baterai Li-ion, tetapi menggunakan elektrolit polimer kering berbentuk lapisan film tipis. Baterai LiPo bekerja dengan memanfaatkan pertukaran ion yang ada pada polimer kering sebagai elektrolit. Lapisan film dari baterai ini disusun berlapis-lapis antara anoda dan katoda sehingga terjadi pertukaran ion. Dengan menggunakan

metode ini, baterai LiPo memiliki keuntungan dapat diproduksi dalam bermacam bentuk dan ukuran (Florianus Lamablawa, 2022).

Namun, selain kelebihan dari arsitektur dari baterai LiPo, terdapat kekurangan berupa aliran pertukaran ion yang lemah saat melalui elektrolit polimer kering. Permasalahan ini dapat diatasi dengan memanaskannya, sehingga terjadi pertukaran ion lebih cepat, tetapi metode ini dirasa tidak dapat diaplikasikan pada keadaan sehari-hari. Tidak hanya itu, memanaskan baterai dapat menyebabkan resiko keamanan baterai akan berkurang.



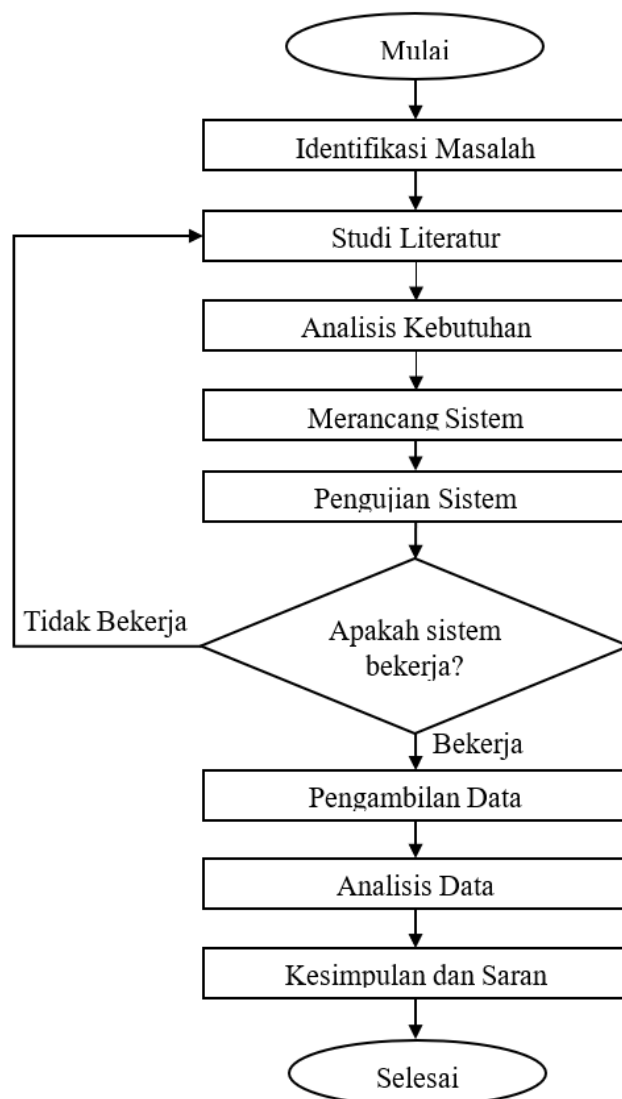
Gambar 2. 12. Baterai Lipo

BAB III

METODE PENELITIAN

3.1. Metode Perancangan

Dalam melaksanakan penelitian diperlukan susunan perancangan untuk menuntuk setiap langkah demi menjalankan penelitian dengan teratur dan mencapai hasil yang diharapkan. Susunan langkah-langkah penelitian ditunjukkan dalam seperti gambar diagram di bawah ini.



Gambar 3. 1. Diagram Alir Penelitian

Diagram tersebut digunakan sebagai acuan untuk merancang sistem pada penelitian ini dari awal hingga selesai penelitian. Adapun maksud dari setiap langkah dari diagram diatas sebagai berikut.

1. Identifikasi Masalah

Penelitian dimulai dengan mengidentifikasi permasalahan yang ada. Identifikasi dilakukan terhadap permasalahan apa yang muncul dan ingin diteliti topik penelitian.

2. Studi Literatur

Pada tahap kedua mencari literatur, informasi, dan referensi yang memiliki kaitan dengan topik yang akan dilakukan penelitian. Contoh literatur yang dapat dijadikan sebagai sumber informasi dan referensi yaitu seperti buku, jurnal, maupun karya ilmiah.

3. Analisis Kebutuhan

Tahap analisis kebutuhan yaitu tahapan untuk melakukan analisis terhadap apa yang diperlukan dalam menunjang penelitian seperti apa misi yang harus dijalankan sesuai peraturan pada kontes sehingga perlu dilakukan penelitian ini untuk membangun sistem robot, menentukan komponen-komponen yang sesuai spesifikasi yang ditentukan pada aturan kontes, dan menentukan *software* yang diperlukan untuk melakukan pengoprasian sistem ataupun pemrograman sistem.

4. Merancang Sistem

Pada tahap ini dilakukan perancangan keseluruhan sistem yang akan dibuat mulai dari perancangan hardware, alur kerja sistem, hingga penulisan kode program sistem.

5. Pengujian Sistem

Pengujian dilakukan untuk mengecek apakah sistem sudah dapat bekerja atau belum. Apabila terjadi permasalahan dalam sistem yang dikarenakan kesalahan pada perancangan sistem, maka mengulangi tahap penelitian dari studi literatur untuk mengecek kesalahan dan kekurangan dari sistem yang sudah dibuat. Oleh karena itu dapat dilakukan perbaikan sistem yang diteliti.

6. Pengambilan Data

Setelah sistem yang dirancang dapat bekerja dengan baik dan sesuai dengan rencana, maka dilakukan pengambilan data dari hasil pengujian. Data yang didapatkan digunakan untuk dianalisis pada tahap selanjutnya.

7. Analisis Data

Analisis data merupakan tahap tentang meneliti, memodelkan, dan mengidentifikasi data untuk mendapatkan informasi yang berguna. Tahapan ini membantu penelitian dalam memahami sistem yang sudah dirancang dan diujikan.

8. Kesimpulan dan Saran

Setelah dilakukan analisis didapatkan informasi terkait sistem, sehingga tahap terakhir ini akan dilakukan pengambilan kesimpulan mengenai apakah penelitian ini sudah sesuai dan dapat menjawab rumusan masalah yang ada pada penelitian ini. Selain itu, diberikan beberapa saran dengan tujuan untuk meminimalisir gangguan dan meningkatkan kinerja sistem yang telah dirancang.

3.2. Waktu dan Tempat Penelitian

Penelitian tugas akhir ini dilakukan ujicoba terhadap pergerakan dan perpindahan robot hingga beberapa meter, sehingga diperlukan area pengujian yang luas untuk robot dapat bergerak dengan aman. Oleh karena itu, penelitian ini dilakukan di daerah kampus yang memiliki area yang cukup luas seperti laboratorium MRC (Microcontroller and Robotic Club) dan lobby gedung G5.

3.3. Objek Penelitian

Tugas akhir ini melakukan penelitian mengenai rancang bangun sistem pada robot ABU MR DART tahun 2024 yang berfokus pada sistem pengendali robot secara *autonomous* dalam mendeteksi mendekati bola berwarna. Dalam mendeteksi bola dirancang sistem yang dapat mengetahui posisi bola berwarna menggunakan kamera untuk dilakukan pengolahan citra menggunakan bahasa pemrograman Python. Sedangkan pengendalian robot menggunakan sistem kendali PID agar dapat mengendalikan robot sesuai dengan nilai yang didapat dari hasil pengolahan

citra dalam mendeteksi bola berwarna. Oleh karena itu, penelitian ini dilakukan untuk mengintegrasikan dua sistem, deteksi bola dan kontrol PID, pada robot ABU MR DART tahun 2024.

3.4. Analisis Kebutuhan

Pada penelitian tugas akhir sebelum melakukan perancangan sistem perlu untuk melakukan analisis terhadap kebutuhan yang diperlukan saat merancang ataupun menjalankan sistem. Dalam menganalisis, kebutuhan penelitian dibagi menjadi tiga yaitu kebutuhan rancangan sistem, kebutuhan perangkat keras, dan kebutuhan perangkat lunak. Berikut merupakan perangkat yang akan digunakan pada penelitian ini.

3.4.1. Kebutuhan Rancangan Sistem

Berdasarkan buku panduan Kontes Robot Indonesia 2024 pada bagian Kontes Robot ABU Indonesia, perlombaan ini melibatkan dua buah robot, di mana salah satunya dikendalikan secara *autonomous*. Robot *autonomous* tersebut memiliki tugas untuk bergerak dari area awal menuju zona tiga guna menjalankan misi utama, yaitu mendeteksi dan mengambil bola berwarna untuk kemudian dimasukkan ke dalam keranjang. Oleh karena itu, penelitian ini difokuskan pada perancangan sistem yang mampu melakukan pembacaan posisi bola secara otomatis, serta mengendalikan pergerakan robot agar dapat menyelaraskan posisinya dengan posisi bola. Sistem yang dikembangkan terdiri atas tiga bagian utama, yaitu input, proses, dan output. Bagian input berupa tangkapan citra dari kamera, proses mencakup pengolahan citra digital untuk mendeteksi dan menentukan posisi bola, sedangkan output berupa perintah kendali yang diberikan kepada aktuator untuk mengarahkan robot menuju bola yang telah terdeteksi.

3.4.2. Kebutuhan Perangkat Keras

Perangkat keras atau hardware dibutuhkan sebagai pendukung dalam merancang sistem pada penelitian. Perangkat keras yang akan digunakan memiliki beberapa tujuan dan fungsi nya masing-masing dengan beberapa spesifikasinya. Spesifikasi setiap perangkat keras dan fungsi perangkat diantaranya adalah:

a. Spesifikasi Laptop

Laptop digunakan sebagai tempat pemrograman sistem dalam pengolahan citra maupun kontrol PID. Selain itu, laptop juga sebagai tempat menjalankan program Python dalam mendeteksi objek bola berwarna. Berikut spesifikasi laptop yang digunakan.

Processor	: AMD Ryzen 5 5600H
VGA	: Radeon Graphics
Sistem Operasi	: Windows 11 Home (64-bit)
RAM	: 16384MB

b. Spesifikasi Arduino Mega

Arduino Mega berfungsi sebagai tempat untuk menjalankan program kontrol PID yang sudah diprogram menggunakan Arduino IDE melalui laptop. Arduino Mega juga sebagai pengendali pergerakan motor menggunakan pin PWM yang terhubung melalui driver motor BTS 7960. Spesifikasi Arduino mega yang akan digunakan sebagai berikut.

Tegangan Operasional	: 5 Volt
Pin Input/Output Digital	: 54
Pin PWM	: 15
Pin Input Analog	: 16
Memori Flash	: 256 KB (8 KB untuk bootloader)
SRAM	: 8 KB
Clock Speed	: 16 MHz
Komunikasi	: UART, I2C, SPI

c. Spesifikasi Webcam

Webcam pada penelitian ini digunakan sebagai alat menangkap citra. Citra yang ditangkap akan diterima laptop untuk dilakukan pengolahan citra dalam mendeteksi bola berwarna. Webcam yang digunakan bermerek logitech dengan spesifikasi webcam sebagai berikut.

Resolusi Maksimal	: 1080p 30 fps, 720p 60 fps
Field of View	: 78° diagonal

Autofocus	: Ya
Low-Light Correction	: Ya
Kesesuaian OS	: Windows 7 atau lebih baru, macOS 10.10 atau lebih baru, ChromeOS

d. Spesifikasi Driver Motor BTS 7960

Driver motor ini berfungsi untuk pengendali arah pergerakan dan kecepatan motor DC. Hal ini dikarenakan driver memiliki rangkaian H-bridge dengan menggunakan mosfet. Berikut spesifikasi yang dimiliki driver motor BTS 7960.

Tegangan Operasional	: 6 – 27 Volt
Arus Maksimum	: 43 Ampere
Tegangan Logika	: 5 Volt
Metode Kontrol	: PWM (Pulse Width Modulation)

e. Spesifikasi Motor PG 45

Seperti halnya motor DC, pada penelitian ini motor PG 45 berfungsi sebagai penggerak. Akan tetapi, motor jenis ini memiliki *gearbox* yang dapat menghasilkan torsi tinggi dan kecepatan rendah. Oleh karena itu, motor PG 45 ideal untuk digunakan sebagai penggerak robot yang cukup berat seperti pada penelitian ini. Spesifikasi motor PG 45 sebagai berikut.

Diameter Motor	: 45 mm
As Motor	: 10 mm
Tegangan	: 24 Volt DC
Arus	: 2.5 – 5 Ampere
Rasio Gearbox	: 1:19.2

3.4.3. Kebutuhan Perangkat Lunak

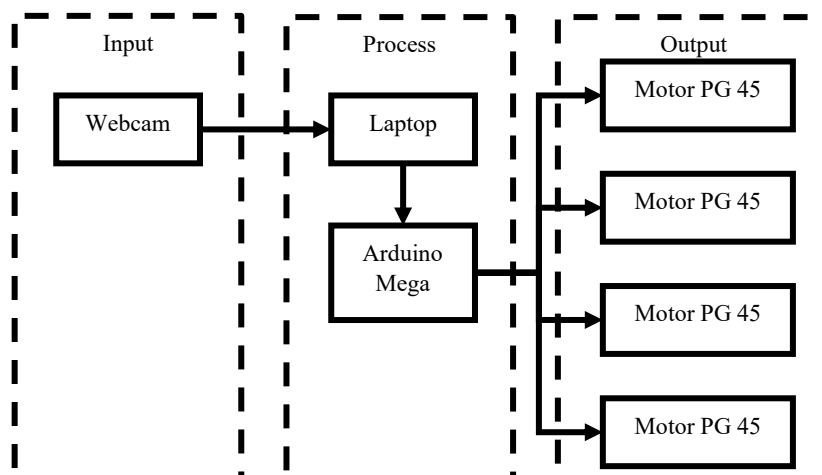
Perangkat lunak pada penelitian ini ditujukan untuk melakukan pemrograman dan melakukan proses pengolahan informasi sehingga robot dapat

bekerja secara *autonomous*. Perangkat lunak yang digunakan dalam menunjang penelitian ini ditunjukkan pada tabel berikut.

Sistem Operasi	: Windows 11 (64-bit)
Bahasa	: 1. Python
Pemrograman	2. Arduino (berbasis C++)
Text Editor	: 1. Visual Studio Code
	2. Arduino IDE
Library	: 1. PySimpleGUI (python)
	2. OpenCV (python)
	3. Serial (python)
	4. Csv (python)
	5. Matplotlib (python)
	6. MPU6050 (Arduino)

3.5. Perancangan Sistem

Sistem yang akan dirancang pada penelitian ini berfokus pada kendali robot secara otonom dengan mengintegrasikan deteksi objek berwarna berupa bola sebagai target arah pergerakan robot. Oleh karena itu diperlukan rancangan mulai dari proses pembacaan bola berwarna hingga menggerakkan robot sesuai dengan parameter yang didapatkan dari hasil deteksi bola. Berikut gambar diagram blok rancangan dari sistem robot yang akan diteliti.



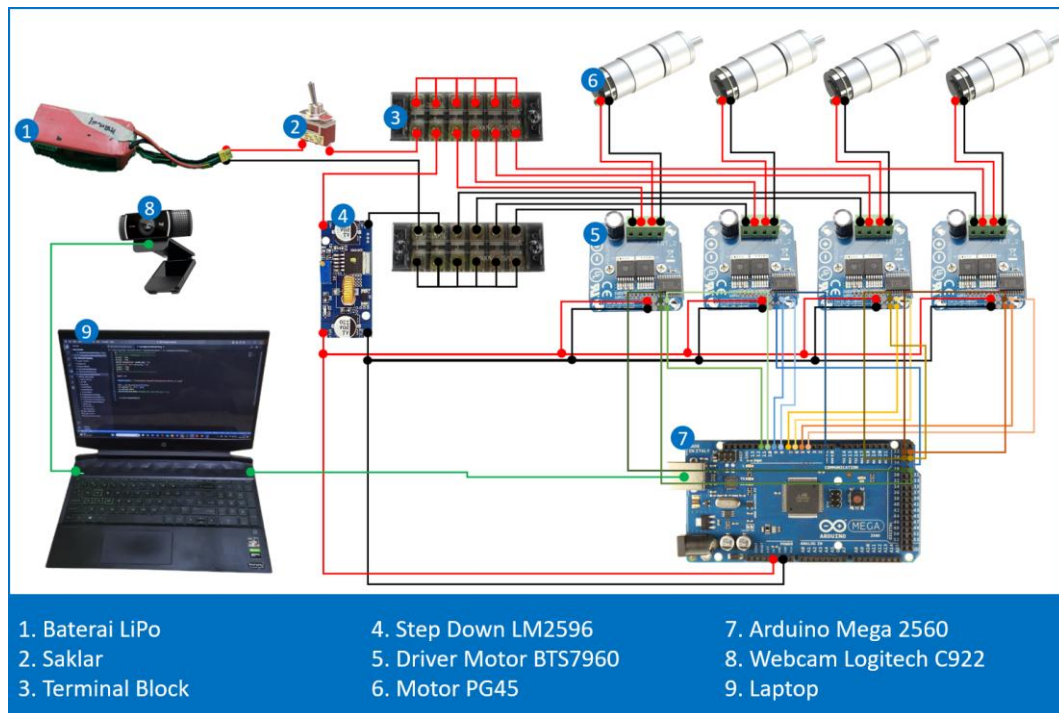
Gambar 3. 2. Diagram Block Full Sistem

Rancangan sistem pada penelitian ini dibagi menjadi tiga tahap, yaitu input, process, dan output. Sistem bekerja diawali dengan mendapat input berupa citra dari tangkapan webcam yang terhubung ke laptop. Kemudian, citra yang didapatkan masuk ke tahap process untuk dilakukan pengolahan citra di perangkat laptop. Pemrosesan yang dilakukan pada tahap ini bertujuan untuk mendapatkan informasi letak dan titik tengah objek bola yang dideteksi berupa nilai koordinat pixel citra. Nilai yang didapatkan tersebut dikirimkan ke Arduino Mega untuk dilakukan pemrosesan dalam mendapatkan nilai PWM dari kontrol PID. PWM yang didapatkan akan digunakan sebagai nilai output dalam mengatur motor PG 45 menggunakan driver motor BTS 7960.

Sistem yang dirancang tersebut tidak terlepas dari perangkat keras maupun lunak. Beberapa perangkat keras yang digunakan meliputi webcam, laptop, arduino mega, driver motor BTS 7960 dan motor PG 45. Sedangkan perangkat lunak terdiri dari algoritma pengolahan citra, dan pemrosesan kontrol PID. Berikut penjelasan lebih detail mengenai perancangan dari perangkat keras dan perangkat lunak.

3.5.1. Perancangan Perangkat Keras

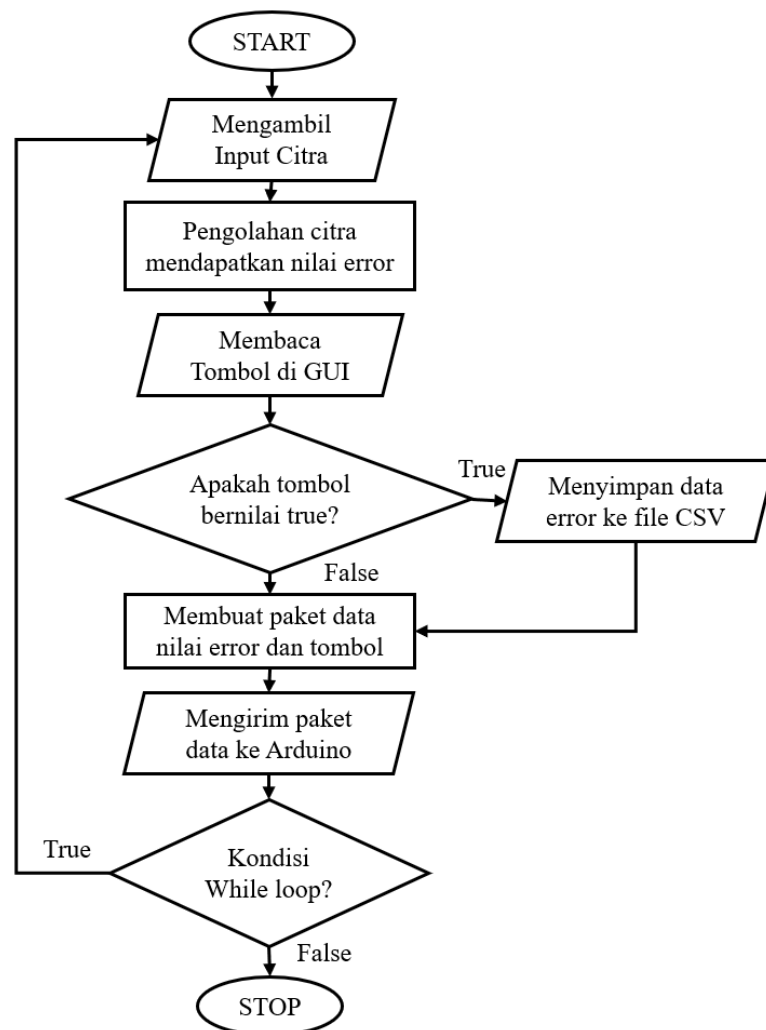
Dalam melakukan perancangan perangkat keras perlu untuk memperhatikan kebutuhan yang diperlukan untuk dapat menunjang berjalannya sistem yang dirancang. Sesuai seperti yang digambarkan pada gambar 3.3 sistem memerlukan perangkat seperti webcam, laptop, Arduino mega, driver motor BTS 7960, dan motor. Namun, terdapat komponen lain yang diperlukan agar perangkat keras dapat berjalan. Perangkat tersebut diantaranya yaitu baterai sebagai sumber daya dalam menyalakan sistem, regulator step down untuk menurunkan tegangan baterai sesuai tegangan kerja perangkat, dan driver motor BTS mengendalikan arah dan kecepatan motor PG 45. Berikut merupakan pengkabelan dari keseluruhan perangkat keras yang digunakan.



Gambar 3. 3. Rangkaian Sistem Perangkat Keras

3.5.2. Perancangan Perangkat Lunak

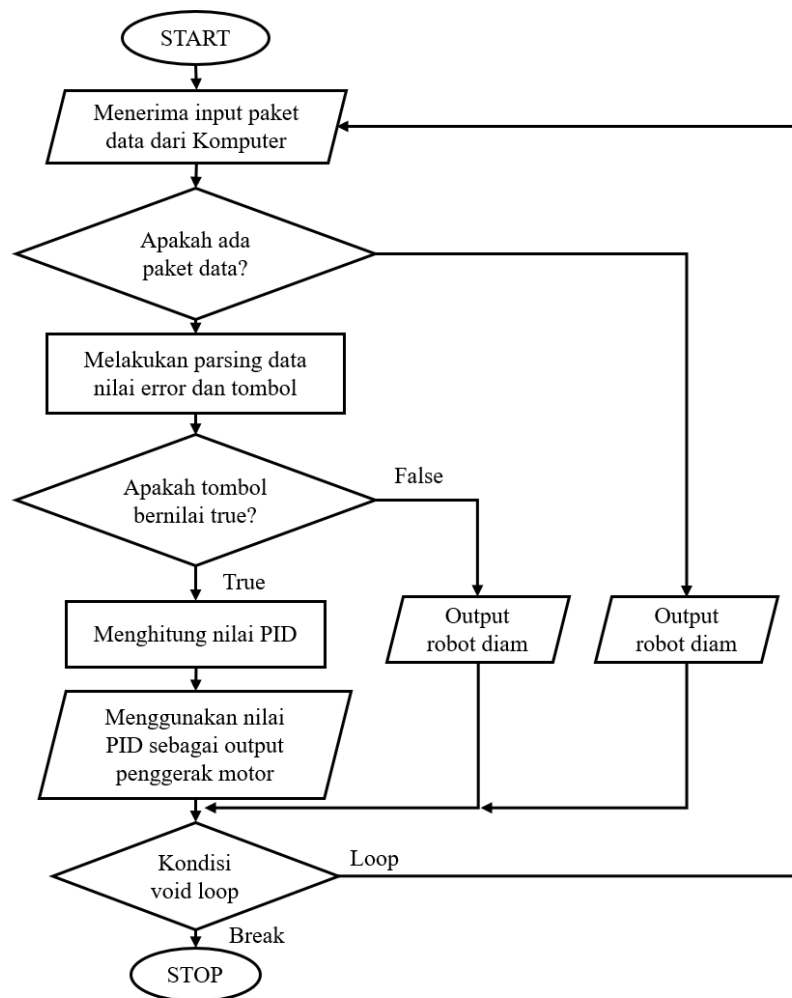
Pada penelitian ini terdapat dua bagian rancangan perangkat lunak sesuai kebutuhan sistem yang memerlukan pemrosesan citra dan kontrol PID. Pemrosesan pengolahan citra memerlukan perangkat lunak berupa program yang ditulis dalam bahasa Python dan dijalankan di laptop. Sedangkan kontrol PID ditulis menggunakan Arduino IDE yang dijalankan di Arduino mega. Berikut ini merupakan algoritma perangkat lunak dalam bentuk *flowchart* dari sistem yang akan dirancang pada penelitian ini.



Gambar 3. 4. Diagram Alir Sistem Deteksi Bola Berwarna

Flowchart diatas merupakan algoritma dari sistem dalam mendapatkan posisi objek bola berwarna hingga mengirimkan data posisi bola ke Arduino. Sistem dimulai dengan mengambil citra yang didapatkan dari perangkat webcam. Selanjutnya, citra tersebut diolah hingga mendapatkan besar nya error posisi bola dalam nilai koordinat pixel citra terhadap titik tengah citra. Kemudian sistem membaca kondisi tombol yang ada di GUI bernilai true atau false. Jika tombol pada GUI bernilai false, maka sistem akan langsung melanjutkan untuk membuat paket data yang berisi kondisi tombol berupa nilai 0 (nol) dan nilai error posisi bola. Sedangkan tombol bernilai true, akan melanjutkan untuk menyimpan data error dan

waktu kedalam file csv agar dapat dianalisis dan diolah menjadi grafik. Selanjutnya, dilanjutkan membuat paket data yang berisi kondisi tombol berupa nilai 1 (satu) dan nilai error. Setelah paket data sudah dibuat, sistem akan mengirim data paket tersebut ke Arduino melalui komunikasi serial. Proses tersebut akan terus berulang hingga program memerintahkan break looping dengan menutup jendela GUI.



Gambar 3. 5. Diagram Alir Sistem Kontrol PID

Flowchart kedua ini merupakan algoritma sistem dalam melakukan kontrol robot dengan menggunakan nilai data error posisi bola sebagai salah satu parameter yang digunakan pada kontrol PID. Algoritma ini digunakan untuk sistem yang diprogramkan didalam Arduino Mega. Sistem dimulai dengan menerima input dan mengecek apakah terdapat paket data yang diterima dari komputer berupa nilai

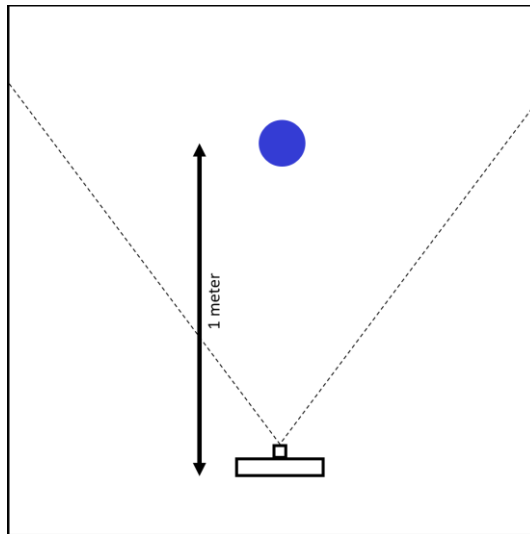
kondisi tombol (1 atau 0) serta nilai error posisi bola. Bila tidak terdapat data yang diterima maka sistem akan memerintahkan robot untuk diam. Sedangkan ketika terdapat data yang diterima, robot akan melanjutkan melakukan proses parsing data untuk memisahkan serta menyimpan data nilai kondisi tombol dan nilai error posisi bola kedalam variabel penyimpanan yang berbeda. Selanjutnya, dilakukan pengecekan nilai kondisi tombol, jika tombol bernilai false atau nol, maka sistem memerintahkan robot untuk diam. Sedangkan ketika bernilai true atau satu, maka robot akan melanjutkan melakukan pemrosesan data error posisi bola untuk mendapatkan nilai PID. Kemudian, nilai PID tersebut digunakan sebagai pengatur arah dan kecepatan pergerakan robot sebagai output. Proses ini dilakukan secara berulang atau looping hingga sistem dimatikan.

3.6. Pengujian Sistem

Pengujian dalam penelitian ini dilakukan untuk mengetahui apakah sistem yang dirancang sudah dapat bekerja dengan baik. Pengujian yang dilakukan melalui beberapa tahap pengujian sebelum melakukan pengujian dari keseluruhan sistem yang sudah terintegrasi. Hal ini akan mempermudah dalam pengecekan permasalahan yang mungkin saja dapat terjadi karena kesalahan dalam merancang sistem. Berikut beberapa tahap pengujian yang dari penelitian ini.

3.6.1. Pengujian Sistem Deteksi Bola Berwarna

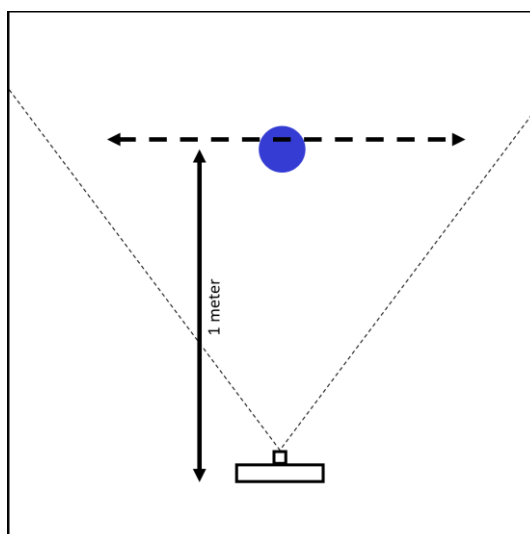
Pengujian sistem ini dimulai dengan meletakkan objek bola berwarna biru dan ungu, serta gambar berwarna didepan webcam dengan jarak 1 meter. Kemudian, pengujian dilakukan dengan mengubah nilai batas Hue, Saturation, dan Value yang ada pada GUI. Pengujian ini bertujuan untuk mendapatkan nilai batas Hue, Saturation, dan Value yang sesuai dengan warna objek bola yang akan dideteksi (warna biru), serta untuk mengetahui apakah sistem dapat berfungsi mendeteksi objek bola berwarna biru dengan benar. Keberhasilan sistem deteksi dalam mendeteksi bola berwarna biru akan ditandai dengan gambar bola berwarna biru yang diberi garis kotak mengelilinginya.



Gambar 3. 6. Skema Pengujian Deteksi Bola Berwarna

3.6.2. Pengujian Simulasi Pembacaan Posisi Objek Terdeteksi

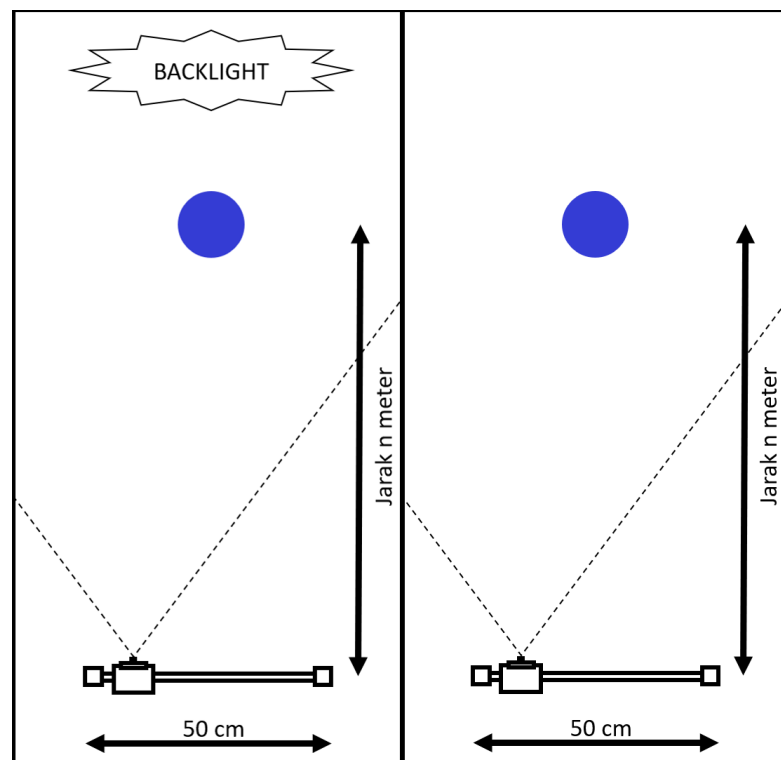
Pengujian simulasi pembacaan posisi objek terdeteksi dilakukan untuk mengetahui apakah sistem sudah dapat mengeluarkan output berupa nilai posisi dalam sumbu x pixel gambar. Pengujian dilakukan dengan menjalankan sistem deteksi warna dan memberikan objek bola berwarna dengan jarak 1 meter didepan kamera dan memberi pergerakan kepada bola. Dari pengujian ini akan dihasilkan grafik perubahan nilai posisi objek bola sesuai dengan pergerakan bola yang diberikan. Berikut ilustrasi untuk menggambarkan proses pengujian yang dilakukan.



Gambar 3. 7. Skema Pengujian Pembacaan Nilai Error

3.6.3. Pengujian Akurasi Deteksi Bola Berwarna

Pengujian tingkat akurasi sistem deteksi bola berwarna dilakukan untuk mengetahui akurasi pendeteksian objek bola berwarna dengan berbagai kondisi jarak dan pencahayaan. Pengujian dilakukan dengan melakukan penangkapan frame gambar sebanyak 1000 frame menghadap kearah bola berwarna yang dideteksi untuk dilakukan pendeteksian bola berwarna pada setiap framenya. Hasil akurasi didapatkan menghitung presentase akurasi dengan membagi jumlah frame yang berhasil mendeteksi bola dengan jumlah total frame yang ditangkap dan dikalikan dengan 100%. Berikut gambaran pengujian yang akan dilakukan.

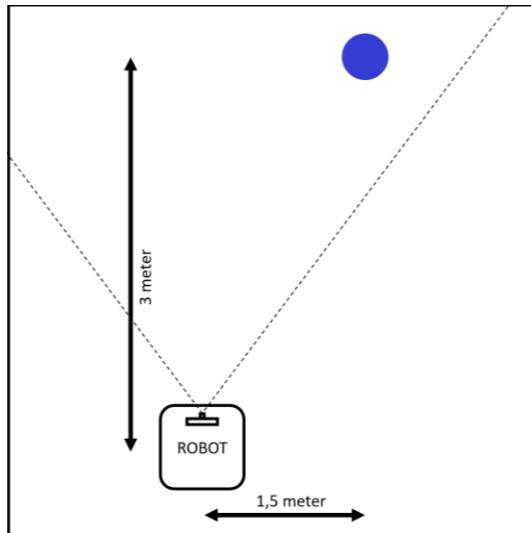


Gambar 3. 8. Skema Pengujian Akurasi Deteksi Bola Berwarna

3.6.4. Pengujian Dua Sistem Terintegrasi

Pengujian ini merupakan pengujian sistem secara menyeluruh yang sudah saling terhubung. Setelah sistem deteksi bola berwarna telah berhasil dilakukan pengujian dan didapatkan output berupa nilai posisi objek bola biru dalam sumbu x pixel gambar, maka dilakukan pengujian terhadap kontrol PID menggunakan input nilai posisi tersebut sebagai *setpoint*. Pengujian dilakukan dengan mengubah nilai

konstanta PID untuk mendapatkan nilai konstanta PID dengan hasil pergerakan robot yang responsif dan ketepatan pergerakan robot terhadap bola yang terdeteksi sebagai titik *setpoint*-nya. Berikut gambaran pengujian yang akan dilakukan.



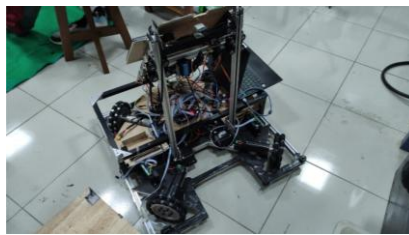
Gambar 3. 9. Skema Dua Sistem Terintegrasi

BAB IV

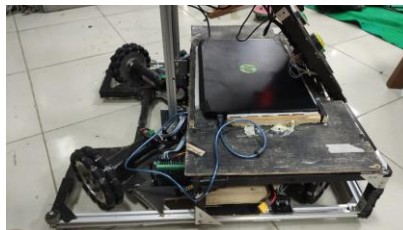
HASIL DAN ANALISIS

4.1. Hasil Rancangan Sistem pada Robot ABU

Pada bagian ini ditampilkan hasil dari perancangan penelitian ini yang sudah diaplikasikan pada robot ABU 2024. Meskipun komponen yang terpasang tidak hanya komponen yang ada di perancangan perangkat keras, tetapi semua komponen yang sudah dirancang sudah sesuai dengan perancangan yang sudah dirancang. Selain itu, software dari sistem ini telah dapat berfungsi baik dalam sistem deteksi objek bola berwarna maupun sistem kontrol PID. Dalam mempermudah melakukan pengawasan, hasil deteksi bola ditampilkan menggunakan GUI (*Graphical User Interface*). Berikut gambar dari hasil rancangan sistem yang sudah diaplikasikan pada robot ABU 2024.



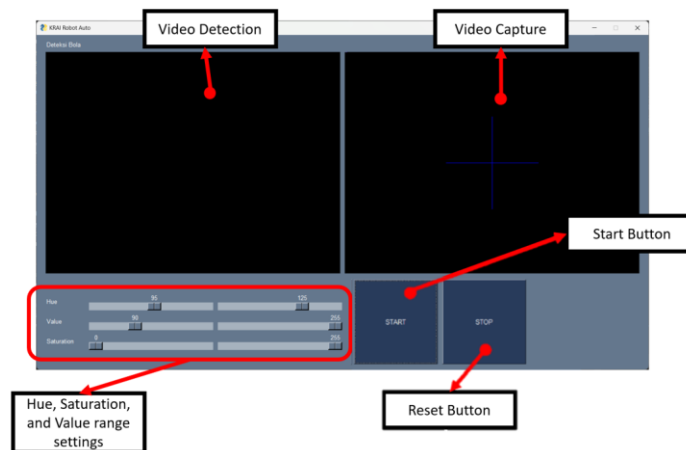
Gambar 4. 1. Rancangan Robot ABU



Gambar 4. 2. Peletakan Laptop Pada Robot ABU

4.2. Hasil *Graphic User Interface*

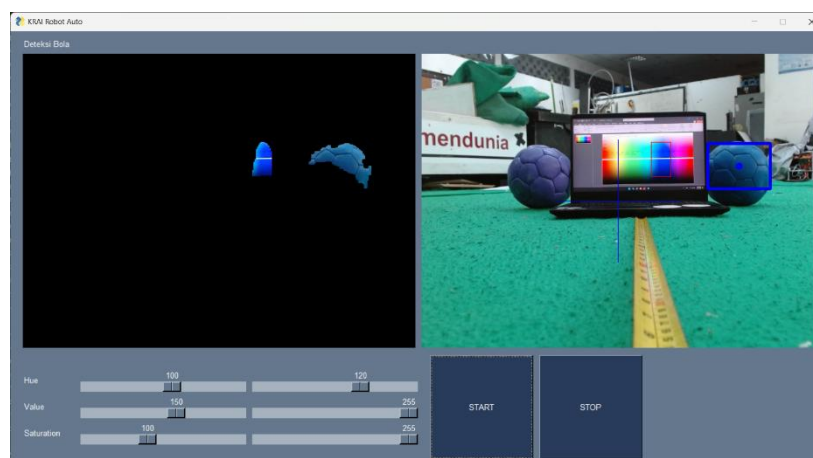
Graphic user interface pada sistem yang dirancang menampilkan beberapa hal, antarlain tangkapan video dari wabcam, video deteksi warna objek, tombol Start, tombol Reset, dan *Trackbar* setelan *threshold* HSV. Berikut merupakan tampilan dari *graphic user interface*.



Gambar 4. 3. Hasil Graphic User Interface

4.3. Hasil Pengujian Sistem Deteksi Bola Berwarna

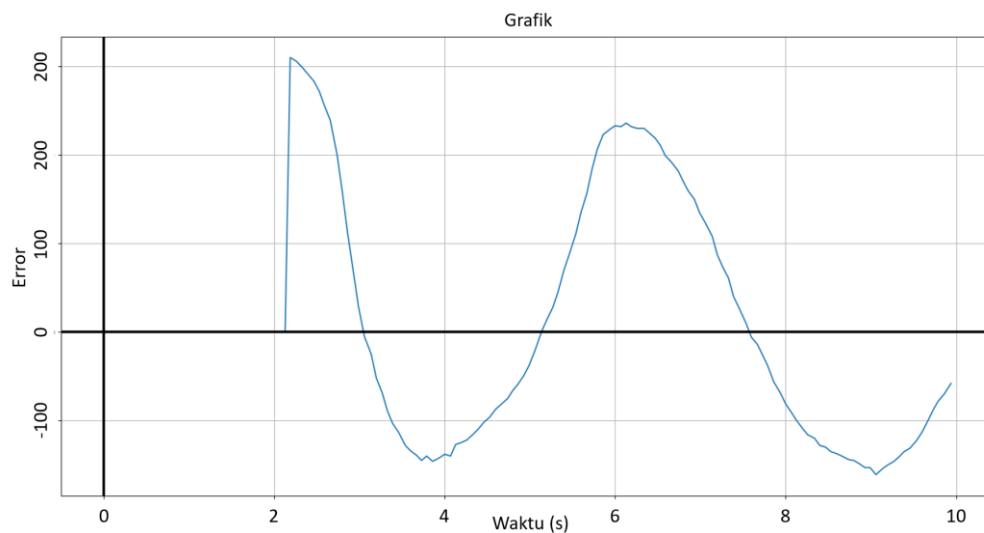
Setelah dilakukan pengujian ini dengan mengubah nilai batas bawah dan atas dari Hue, Saturation, dan Value, didapatkan nilai batas yang sesuai untuk mendeteksi bola berwarna biru. Nilai batas yang didapatkan untuk mendeteksi bola berwarna biru yaitu Hue (100 – 120), Saturation (150 – 255), dan Value (100 – 255). Penggunaan nilai batas tersebut menghasilkan perubahan berupa tanda kotak yang mengelilingi bola berwarna biru pada gambar tangkapan webcam sebelah kanan GUI. Selain itu, pada gambar sebelah kiri menampilkan gambar hasil masking yang hanya menampilkan objek objek berwarna biru. Ini menandakan bahwa sistem dapat membedakan warna objek dan menandai objek yang berwarna biru. Berikut ini merupakan hasil pengujian yang ditampilkan dalam GUI.



Gambar 4. 4. Hasil Pengujian Sistem Deteksi Bola Berwarna

4.4. Hasil Pengujian Simulasi Pembacaan Posisi Objek Bola Biru

Setelah pengujian deteksi objek bola berwarna biru, pengujian dilanjutkan mensimulasikan pembacaan posisi objek bola biru. Pengujian ini bertujuan untuk mengetahui apakah sistem deteksi objek sudah dapat memberikan output berupa nilai posisi objek yang dideteksi. Setelah dilakukan pengujian, didapatkan nilai nilai posisi objek yang terdeteksi setiap waktu nya. Untuk mempermudah pembacaan perubahan nilai posisi objek terhadap waktu, maka ditampilkan dalam bentuk grafik. Berikut ini merupakan hasil dari pengujian simulasi pembacaan posisi objek bola biru.



Gambar 4. 5. Hasil Pengujian Simulasi Pembacaan Nilai Error

Hasil data dan grafik yang dihasilkan menggambarkan kondisi objek bola biru yang diberi pergerakan kearah kanan dan kiri. Hal ini menandakan sistem sudah berhasil untuk mendapatkan data posisi bola. Oleh karena itu, sistem deteksi objek ini sudah dapat digunakan untuk mendapatkan data posisi objek bola biru yang akan digunakan sebagai *setpoint* pada sistem kontrol PID.

4.5. Hasil Pengujian Akurasi Sistem Deteksi

Tujuan pengujian akurasi sistem deteksi ini untuk mengevaluasi apakah sistem sudah dapat melakukan deteksi objek secara *real time* dengan akurat pada kondisi kondisi pencahayaan tertentu. Proses pengujian dilakukan dengan menjalankan

menekan tombol Start pada GUI untuk memulai mengambil frame citra hingga sebanyak 1000 dalam kondisi kamera menghadap kearah bola dan digerakan kearah kanan serta kiri secara bolak balik sebagai gangguan. Pada GUI akan menampilkan hasil perhitungan akurasi dengan membagi jumlah frame yang berhasil mendeteksi bola dengan total jumlah frame yang ditangkap yaitu sebanyak 1000 frame, lalu dikali dengan 100%. Berikut ini merupakan data hasil pengujian akurasi sistem deteksi yang dilakukan pada beberapa kondisi pencahayaan dan jarak objek.

Tabel 4. 1. Hasil Pengujian Akurasi Sistem Deteksi

Jarak Bola (m)	Intensitas Cahaya (lux)			
	Tanpa Gangguan Backlight		Dengan Gangguan Backlight	
	5 lux	90 lux	17 lux	110 lux
1 m	100.00 %	100.00 %	100.00 %	100.00 %
2 m	100.00 %	100.00 %	100.00 %	100.00 %
3 m	100.00 %	100.00 %	46.90 %	100.00 %
4 m	100.00 %	100.00 %	0.00 %	100.00 %
5 m	100.00 %	100.00 %	0.00 %	98.60 %
6 m	100.00 %	100.00 %	0.00 %	26.00 %

Dari data yang didapatkan, menghasilkan akurasi deteksi bola biru dengan akurat dari semua pengujian jarak dan intensitas cahaya pada kondisi tanpa gangguan *backlight*. Akan tetapi, terjadi penurunan akurasi pada kondisi dengan adanya gangguan *backlight*. Akurasi mulai menurun ketika intensitas cahaya 17 lux dengan gangguan *backlight* dan jarak objek lebih dari 2 meter. Selain itu, akurasi mulai menurun ketika kondisi pencahayaan menurun ketika intensitas cahaya 110 lux dengan gangguan *backlight* dan jarak objek lebih dari 4 meter. Oleh karena itu, sistem deteksi ini cocok untuk digunakan pada kondisi pencahayaan tanpa adanya *backlight* pada objek yang akan dideteksi.

4.6. Hasil Pengujian Dua Sistem Terintegrasi

Setelah dilakukan pengujian sistem deteksi, selanjutnya dilakukan pengujian sistem deteksi objek yang terintegrasi dengan sistem kontrol PID. Hal ini perlu dilakukan untuk mengetahui apakah keseluruhan sistem sudah dapat berjalan

dengan sesuai perancangan. Selain itu, pengujian ini dilakukan untuk mendapatkan nilai konstanta PID yang sesuai dengan robot, sehingga robot dapat melakukan pergerakan menghadap bola dengan akurat dan responsif. Proses pengujian tahap ini dilakukan dengan beberapa kali pengujian nilai konstanta PID. Hasil pengujian dijelaskan sebagai berikut.

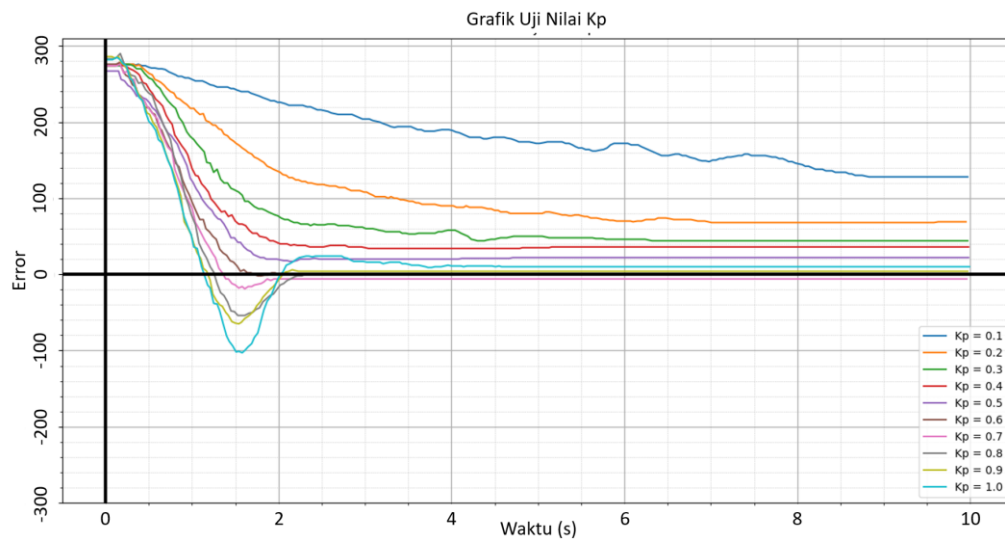
Pengujian nilai konstanta proposional dilakukan dengan beberapa pengujian terhadap 10 nilai konstanta proposional yang berbeda beda. Perbedaan nilai konstanta setiap pengujian sebesar 0,1. Berikut ini merupakan tabel nilai konstanta yang akan diujikan dan ditampilkan hasil pengujian berupa grafik perubahan nilai error posisi robot dari bola terhadap waktu.

Tabel 4. 2. Pengujian Konstanta Proposional

No	Konstanta PID		
	Kp	Ki	Kd
1	0.1	0	0
2	0.2	0	0
3	0.3	0	0
4	0.4	0	0
5	0.5	0	0
6	0.6	0	0
7	0.7	0	0
8	0.8	0	0
9	0.9	0	0
10	1.0	0	0

Pengujian ini dilakukan di lobi gedung G5 Universitas Muhammadiyah Yogyakarta dengan kondisi pencahayaan sedang dengan kisaran intensitas cahaya sebesar 270 lux. Pengujian dilakukan dengan mengubah nilai parameter konstanta proposional pada program Arduino secara bergantian. Hasil pengujian didapat dari perubahan error posisi bola terhadap titik tengah kamera yang disimpan dengan menggunakan program Python dalam bentuk file CSV. Kemudian data-data

percobaan diolah menjadi grafik dalam satu buah diagram. Data hasil pengujian yang dilakukan sebagai berikut.



Gambar 4. 6. Grafik Pengujian Konstanta Proposional

Percobaan dilakukan dengan memposisikan robot pada kondisi nilai error sekitar 280 (posisi bola berada disebelah kanan sebanyak 280 pixel dari titik tengah gambar). Nilai error yang berubah mendekati nilai 0 menandakan bahwa robot bergerak memposisikan bola ke tengah pandangan webcam. Dari grafik yang dihasilkan terdapat perbedaan perubahan error yang terjadi dengan perbedaan nilai konstanta yang di berikan. Semua grafik menunjukkan berubah nilai error untuk mendekati nilai 0 (nol), tetapi kecepatan perubahan nilai error setiap grafik berbeda-beda. Hal ini terjadi karena penggunaan nilai Kp yang berbeda-beda.

Ketika menggunakan $K_p = 0,1$, perubahan error nya selama 9 detik hanya dapat mendekati hingga nilai error kurang lebih sebesar 125. Penggunaan $K_p = 0,2$; $K_p = 0,3$; $K_p = 0,4$; $K_p = 0,5$ hasilnya lebih baik dari $K_p = 0,1$. Akan tetapi, perubahan nilai error masih kurang sempurna karena nilai error nya masih cukup besar. Dimana nilai $K_p = 0,2$ menghasilkan error menjadi sekitar 75, $K_p = 0,3$ sebesar 40, $K_p = 0,4$ sekitar 28, dan $K_p = 0,5$ sekitar 22.

Pada penggunaan nilai $K_p = 0.6$ menghasilkan perubahan nilai error yang signifikan dengan hanya membutuhkan waktu selama 1,8 detik nilai error sudah

bernilai 0. Ketika dilakukan pengujian kembali menggunakan nilai K_p yang lebih tinggi seperti $K_p = 0,7$; $K_p = 0,8$; $K_p = 0,9$; $K_p = 1,0$ Menghasilkan perubahan yang lebih signifikan tetapi tidak stabil. Hal ini dikarenakan adanya *overshoot* yang disebabkan kecepatan pergerakan robot yang berlebih sehingga robot tidak sanggup untuk berhenti lebih cepat. Dengan demikian, sistem ini dapat menghasilkan perubahan nilai error yang signifikan dan pergerakan robot yang responsif ketika menggunakan konstanta proposional sebesar 0,5.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil dari tahap perancangan sistem, pengujian sistem, pengambilan data pengujian dan analisis hasil pengujian sistem deteksi objek bola berwarna serta kontrol PID yang telah dilaksanakan, maka penulis mendapatkan Kesimpulan bahwa:

1. Perancangan sistem deteksi objek bola berwarna berbasis computer vision menggunakan library OpenCV pada bahasa pemrograman Python sudah dapat bekerja dengan baik. Sistem dapat mencapai tingkat akurasi 100 % dalam mendeteksi bola berwarna biru pada jarak dekat 1 meter maupun jarak maksimal arena lomba dari KRAI 2024 yaitu 6 meter. Tingkat akurasi ini terjadi ketika tidak adanya gangguan pencahayaan berupa *backlight* pada objek bola biru yang dideteksi.
2. Berdasarkan hasil pengujian sistem deteksi terhadap pembacaan nilai error (posisi objek terdeteksi dalam sumbu x pixel tangkapan gambar) yang telah dilakukan, didapatkan hasil grafik yang menggambarkan posisi bola terhadap sudut pandang kamera ataupun robot kearah kanan atau kiri robot. Oleh karena itu, Sistem ini sudah bekerja sesuai rancangan untuk dapat mengetahui posisi bola berwarna terhadap robot.
3. Dari hasil pengujian dua sistem terintegrasi didapatkan 10 grafik dalam satu gambar yang menggambarkan kondisi perubahan nilai error (posisi objek terdeteksi dalam sumbu x pixel tangkapan gambar) terhadap waktu dengan pengujian terhadap penggunaan 10 nilai konstanta propotional yang berbeda-beda. Didapatkan hasil perubahan nilai error yang signifikan dan pergerakan robot yang responsif dengan perubahan nilai error dari 280 menjadi 0 dalam waktu 1,8 detik pada penggunaan nilai konstanta propotional sebesar 0,6. Oleh karena itu, penggunaan konstanta propotional yang terbaik dalam sistem kontrol PID untuk digunakan pada robot ini yaitu sebesar 0,6.

5.2. Saran

Dalam perancangan sistem kendali autonomous dengan terintegrasi dengan color detection berbasis *computer vision* untuk robot terdapat beberapa kekurangan dan dapat dikembangkan dengan melakukan:

1. Dalam meningkatkan tingkat akurasi sistem deteksi objek pada sistem ini dapat dilakukan dengan memperhatikan beberapa kondisi seperti kondisi pencahayaan yang stabil dan tidak ada perbedaan kondisi cahaya pada saat menyetel nilai batas bawah dan atas HSV dengan kondisi cahaya pada tempat penerapan robot. Selain itu, dapat dengan mengganti metode deteksi objek berwarna dengan metode yang lebih akurat.
2. Penggantian webcam dengan spesifikasi yang lebih baik dalam menangkap citra gambar. Hal ini agar sistem dapat menangkap citra gambar tanpa mengalami perubahan atau perbedaan warna dari warna objek sebenarnya.
3. Menambahkan konstanta integral dan konstanta derivative pada sistem kendali PID agar dapat meningkatkan tingkat responsif pergerakan robot.

DAFTAR PUSTAKA

- Ahadi, A. H. (2024). Implementasi Sistem Pendeteksia Warna Objek dengan OpenCV-pyhton. *SENTRI: Jurnal Riset Ilmiah*, 3, 5.
- Apridiansyah, Y., & Gumiri, J. R. (2021). Penerapan Metode Background Subtraction Untuk Deteksi Gerak Pada Kendaraan. *JUKOMIKA (Jurnal Ilmu Komputer dan Informatika)*, 4(1), 47-56.
- Arifin, S. (2022). Sistem Pendeteksi Bola Menggunakan Open CV pada Mobile Robot. *Jurnal Inovasi Teknologi dan Edukasi Teknik*, 2(3), 131-137.
- Benyamin Kusumoputro, M. H. P., Heru Santoso Budi Rochardjo, Gigih Prabowo. (2023). *Pedoman Kontes Robot Indonesia 2024*. Balai Pengembangan Talenta Indonesia Pusat Prestasi Nasional Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi Republik Indonesia.
- Dzil, F., Khumaidi, A., Rahmat, M. B., Endrasmono, J., Syai'in, M., & Riananda, D. P. (2024). Deteksi Objek di Lapangan pada Robot Sepakbola Beroda Menggunakan Metode YOLOV5. *Jurnal Elektronika dan Otomasi Industri*, 11(2), 604-611.
- Ekadipa, F., Ikawanty, B. A., & Luqman, M. (2024). SISTEM KENDALI KECEPATAN MOTOR DC UNTUK ALAT PEMERAS MADU MENGGUNAKAN METODE PID. *Scientica: Jurnal Ilmiah Sains dan Teknologi*, 2(7), 317-329.
- Fahmi Erza, H. F., Eko Setiawan (2022). Sistem Object Tracking pada Quadcopter Menggunakan Segmentasi Citra dengan Deteksi Warna HSV dan Metode Regresi Linier Berbasis Raspberry Pi. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, 9(7).
- Florianus Lamablawa, S. A. (2022). KARAKTERISTIK LITHIUM-POLYMER BATTERY UNTUK APLIKASI RADIO YANG DI GUNAKAN.
- Hartono, A., Fitriyah, H., & Widasari, E. R. (2023). Sistem Deteksi Tomat Matang Hidroponik berdasarkan Warna Hue, Saturation dan Value menggunakan Metode Threshold berbasis Graphical User Interface. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 7(5), 2313-2319.

- Jayadi, A., & Kurniawan, J. P. S. R. (2022). Rancang Bangun Pengendalian Antena Tracker Pada Horizontal Axis Menggunakan Kendali PID.
- Maulana, M. B. R., Dewatama, D., & Fauziyah, M. (2024). KONTROL KECEPATAN MOTOR INDUKSI 3 FASA BERBASIS ARDUINO MEGA 2560 PADA LIFT 4 LANTAI. *Kohesi: Jurnal Sains dan Teknologi*, 2(6), 81-90.
- Mawarni, D. I., Indarto, I., Deendarlianto, D., & Yuana, K. A. (2023). Metode Digital Image Processing Untuk Menentukan Distribusi Ukuran Diameter Gelembung Udara Pada Microgelembung Generator. *Journal of Information System Management (JOISM)*, 4(2), 132-136.
- Mustar, M. Y., Hartanto, R., & Santosa, P. I. (2024). Exploring Attentive User Interface Input via Raspberry Pi, based on Face Landmark Detection, Eye Open-Closed Detection and Head Movements Detection. *Ingenierie des Systemes d'Information*, 29(4), 1343.
- Nugroho, H., Kurniawan, M., & Saidatin, N. (2019). Deteksi Wajah dan Mata dengan Menggunakan Metode Fitur Haar-Like pada Kamera WebCam. Prosiding Seminar Nasional Sains dan Teknologi Terapan,
- Prasetyo, P., Maulana, R., & Setiawan, E. (2022). Sistem Kontrol PID pada Purwarupa Robot Pembawa Barang berbasis Line Follower menggunakan Algoritma Path Planning. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 6(8), 3741-3750.
- Rahman, S. N., Putra, O. E., & Nugraha, A. (2020). Sistem Pengontrolan Alat Pemanggang Makanan Jarak Jauh Berbasis Arduino Mega 2560. *Majalah Ilmiah UPI YPTK*, 59-64.
- Ramadhan, A. E. N., Setiawan, W., & Khrisne, D. C. (2023). Rancang Bangun Deteksi Objek dengan Metode Filter Warna HSV pada Sistem Klasifikasi Kualitas Biji Kopi Berbasis NVIDIA Jetson Nano. *Jurnal Teknik Industri Terintegrasi (JUTIN)*, 6(4), 1500-1509.
- Sujono, Y. R., Budi, E. S., & Nugrahanto, I. (2023). Modul Pengaturan Motor Pompa DC Metode PID pada Sistem Kontrol Ketinggian Air berbasis Arduino. *Jurnal Elektronika dan Otomasi Industri*, 10(2), 128-136.

- Sulistiyowati, I., Ichsan, H. M., & Anshory, I. (2024). KONVEYOR PENYORTIR OBJEK DENGAN DETEKSI WARNA MENGGUNAKAN KAMERA ESP-32 BERBASIS OPEN-CV PYTHON. Prosiding TAU SNARS-TEK Seminar Nasional Rekayasa dan Teknologi,
- Tyas, U. M., & Buckhari, A. A. (2023). IMPLEMENTASI APLIKASI ARDUINO IDE PADA MATA KULIAH SISTEM DIGITAL. *TEKNOS: Jurnal Pendidikan dan Teknologi*, 1(1), 1-9.
- Ua, A. M. S., Marpaung, E. S. K., Ong, J., Savinka, M., Nurhaliza, P., & Ningsih, R. Y. (2023). Penggunaan Bahasa Pemrograman Python Dalam Analisis Faktor Penyebab Kanker Paru-Paru. *Jurnal Publikasi Teknik Informatika*, 2(2), 88-99.
- Wahab, F., Zaskie, C., & Arthaya, B. M. (2021). Kendali Kecepatan Robot Beroda Omni dengan Kemampuan Menuju Posisi dan Orientasi yang Diinginkan Berbasis Pengendali PID.
- Wardana, A. K. S. (2024). *PERANCANGAN PWM VOLTAGE CONTROLLER SEBAGAI PENGATUR KECEPATAN MOTOR DC SHUNT MOBIL LISTRIK* Institut Teknologi Nasional Malang].
- Wibowo, C. F., Haq, F. I., Ansaris, F. M., & Agustin, S. (2024). ANALISIS PERSENTASE WARNA BLUE GEM PADA SKIN CASE HARDENED DI COUNTER STRIKE 2 MENGGUNAKAN OPENCV DAN PYTHON. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(4), 7241-7247.
- Zakariya, M. A., Astiningrum, M., & Syulistyo, A. R. (2020). Identifikasi Kualitas Biji Jagung Manis Layak Jual dari Warna dan Tekstur Menggunakan HSV dan Gray Level Run Length Matrix (GLRLM). *Jurnal Informatika Polinema*, 7(1), 37-44.

LAMPIRAN

```

1 import PySimpleGUI as sg
2 import cv2 as cv
3 import serial
4 from time import time
5 import FungsioDeteksiObjek
6 import FungsioLahData
7 import serial
8 import matplotlib.pyplot as plt
9
10 arduino = serial.Serial(port="COM4", baudrate=9600, timeout=0.01)
11
12 FrameKamera = [
13     [sg.Text("Deteksi Bola", size=(30, 1))],
14     [sg.Image(filename="", key="Kamera"), sg.Image(filename="", key="Kamera2")]
15 ]
16
17 Trackbar = [
18     [sg.Text("hue", size=(10, 1))],
19     sg.Slider(range=(0,179), orientation="horizontal", size=(30,17), enable_events=True, key="TrackbarLowerhue", default_value=95),
20     sg.Slider(range=(0,179), orientation="horizontal", size=(30,17), enable_events=True, key="TrackbarUpperhue", default_value=125)],
21     [sg.Text("Value", size=(10, 1))],
22     sg.Slider(range=(0,255), orientation="horizontal", size=(30,17), enable_events=True, key="TrackbarLowerSaturation", default_value=90),
23     sg.Slider(range=(0,255), orientation="horizontal", size=(30,17), enable_events=True, key="TrackbarUpperSaturation", default_value=255)],
24     [sg.Text("Saturation", size=(10, 1))],
25     sg.Slider(range=(0,255), orientation="horizontal", size=(30,17), enable_events=True, key="TrackbarLowerValue", default_value=0),
26     sg.Slider(range=(0,255), orientation="horizontal", size=(30,17), enable_events=True, key="TrackbarUpperValue", default_value=255),]
27
28 Tombol = [
29     [sg.Button(button_text="START", size=(20,10),enable_events=True, key="START")],
30     [sg.Button(button_text="STOP", size=(20,10),enable_events=True, key="STOP")]
31 ]
32
33 LayoutFull = [
34     [sg.column(FrameKamera)],
35     [sg.column(Trackbar), sg.column(Tombol)],
36 ]
37
38 window = sg.Window(title="KRAI Robot Auto", layout=LayoutFull, finalize=True)
39 kamera = cv.VideoCapture(1, cv.CAP_DSHOW)
40 kondisiTombol = False
41 simpanData = False
42 PenyimpananAktu = []
43 PenyimpananNilai = []
44 waktuStart = time()
45
46 while True:
47     event, value = window.read(timeout=0.0001)
48     if event == sg.WIN_CLOSED:
49         break
50     ret, frame = kamera.read()
51     seleksi_warna, xError2, KeberadaanBola = FungsioDeteksiObjek.DeteksiObjek(
52         frame=frame,
53         L_h= value["TrackbarLowerhue"], U_h= value["TrackbarUpperhue"],
54         L_s= value["TrackbarLowerSaturation"], U_s= value["TrackbarUpperSaturation"],
55         L_v= value["TrackbarLowerValue"], U_v= value["TrackbarUpperValue"],
56         ukuranTol=100
57     )
58     seleksi_warna = cv.resize(seleksi_warna, (int(frame.shape[1]),int(frame.shape[0])))
59     imgKamera = cv.imencode('.png', seleksi_warna)[1].tobytes()
60     window["Kamera"].update(data=imgKamera)
61
62     frame = cv.resize(frame, (int(frame.shape[1]),int(frame.shape[0])))
63     imgKamera = cv.imencode('.png', frame)[1].tobytes()
64     window["Kamera2"].update(data=imgKamera)
65
66     if event == "START":
67         waktuStart = time()
68         kondisiTombol = True
69         simpanData = False
70     if event == "STOP":
71         simpanData = True
72         kondisiTombol = False
73
74     if kondisiTombol:
75         waktu = round((time() - waktuStart),2)
76         if waktu < 10:
77             FungsioLahData.TampungData(PenyimpananAktu, waktu, PenyimpananNilai, xError2)
78         else:
79             simpanData = True
80             kondisiTombol = False
81             PaketData = str(xError2) + "," + str(0) + ","
82             PaketData = str(xError2) + "," + str(0) + ","
83             data = PaketData + "\n"
84             arduino.write(data.encode("UTF-8"))
85             print(waktu)
86         else:
87             PaketData = str(xError2) + "," + str(0) + ","
88             data = PaketData + "\n"
89             arduino.write(data.encode("UTF-8"))
90
91     if simpanData:
92         FungsioLahData.SimpanData(PenyimpananAktu, PenyimpananNilai)
93         plt.plot(PenyimpananAktu, PenyimpananNilai)
94         plt.title("Grafik")
95         plt.xlabel("Waktu")
96         plt.ylabel("Error")
97         plt.grid(True)
98         plt.axhline(0, color='black',linewidth=3)
99         plt.axvline(0, color='black',linewidth=3)
100         plt.show()
101         PenyimpananAktu = []
102         PenyimpananNilai = []
103         simpanData = False

```

```

1  import pysimplegui as sg
2  import cv2 as cv
3  import serial
4  from time import time
5  import FungsDeteksiObjek
6  import FungsIOlahData
7  import serial
8  import matplotlib.pyplot as plt
9
10 #arduino = serial.Serial(port="COM4", baudrate=9600, timeout=0.01)
11 frameCamera = [
12     [sg.Text('Deteksi Bola', size=(30, 1))],
13     [sg.Image(filename='', key='Kamera'), sg.Image(filename='', key='Kamera2')]
14 ]
15
16 Trackbar = [
17     [sg.Text('hue', size=(10, 1))],
18     sg.Slider(range=(0,170), orientation='horizontal', size=(30,12), enable_events=True, key='TrackbarLowerHue', default_value=100),
19     sg.Slider(range=(0,170), orientation='horizontal', size=(30,12), enable_events=True, key='TrackbarUpperHue', default_value=120)],
20     [sg.Text('value', size=(10, 1))],
21     sg.Slider(range=(0,255), orientation='horizontal', size=(30,12), enable_events=True, key='TrackbarLowerSaturation', default_value=150),
22     sg.Slider(range=(0,255), orientation='horizontal', size=(30,12), enable_events=True, key='TrackbarUpperSaturation', default_value=255)],
23     [sg.Text('saturation', size=(10, 1))],
24     sg.Slider(range=(0,255), orientation='horizontal', size=(30,12), enable_events=True, key='TrackbarLowerValue', default_value=100),
25     sg.Slider(range=(0,255), orientation='horizontal', size=(30,12), enable_events=True, key='TrackbarUpperValue', default_value=255)]
26 ]
27
28 Tombol = [
29     [sg.Button(button_text='START', size=(20,5),enable_events=True, key='START')],
30     [sg.Button(button_text='RESET', size=(20,5),enable_events=True, key='RESET')]
31 ]
32
33 RumusAkurasi = [
34     [sg.Text('JumlahFrameDeteksi', size=(12, 1), justification='center')],
35     [sg.HSeparator()],
36     [sg.Text('JumlahFrame', size=(12, 1), justification='center')]
37 ]
38
39 PerhitunganAkurasi = [
40     [sg.Text('0', size=(5, 1), justification='center', enable_events=True, key='JUMLAHFRAMEDETEKSI')],
41     [sg.HSeparator()],
42     [sg.Text('0', size=(5, 1), justification='center', enable_events=True, key='JUMLAHFRAME')]
43 ]
44
45 DataPresentase = [
46     [sg.Text('Akurasi (%) =', size=(10, 1))],
47     sg.Column(RumusAkurasi),
48     [sg.Text('x 100%', size=(7, 1))],
49     [sg.Text('Akurasi (%) =', size=(10, 1))],
50     sg.Column(PerhitunganAkurasi),
51     [sg.Text('x 100%', size=(7, 1))],
52     [sg.Text('0.00', size=(5, 1), background_color='blue', justification='center', enable_events=True, key='PRESENTASE')],
53     [sg.Text('%', size=(3, 1))]
54 ]
55
56 LayoutFull = [
57     [sg.Column(frameCamera)],
58     [sg.Column(Trackbar), sg.Column(Tombol), sg.Column(DataPresentase)],
59 ]
60
61 window = sg.Window(title="URAI Robot Auto", layout=LayoutFull, finalize=True)
62 kamera = cv.VideoCapture(1, cv.CAP_DSHOW)
63 kondisiTombol = False
64 simpanData = False
65 PenyimpananMaktu = []
66 PenyimpananNilai = []
67 waktuStart = time()
68
69 JumlahFrame = 0
70 JumlahFrameDeteksi = 0
71
72 while True:
73     event, value = window.read(timeout=0.0001)
74     if event == sg.WIN_CLOSED:
75         break
76     ret, frame = kamera.read()
77     seleksi_warna, error2, KeberadaBola = FungsDeteksiObjek.DeteksiObjek(
78         frame=frame,
79         l_h= value['TrackbarLowerHue'], u_h= value['TrackbarUpperHue'],
80         l_s= value['TrackbarLowerSaturation'], u_s= value['TrackbarUpperSaturation'],
81         l_v= value['TrackbarLowerValue'], u_v= value['TrackbarUpperValue'],
82         ukuranPin=50
83     )
84     seleksi_warna = cv.resize(seleksi_warna, (int(frame.shape[1]),int(frame.shape[0])))
85     imgKamera = cv.imencode('.png', seleksi_warna)[1].tobytes()
86     window['Kamera'].update(data=imgKamera)
87     frame = cv.resize(frame, (int(frame.shape[1]),int(frame.shape[0])))
88     imgKamera = cv.imencode('.png', frame)[1].tobytes()
89     window['Kamera2'].update(data=imgKamera)
90
91     if event == "START":
92         waktuStart = time()
93         kondisiTombol = True
94     if event == "RESET":
95         kondisiTombol = False
96         JumlahFrame = 0
97
98     JumlahFrameDeteksi = 0
99     window['JUMLAHFRAMEDETEKSI'].update('0')
100    window['JUMLAHFRAME'].update('0')
101    window['PRESENTASE'].update('0.00')
102
103    if kondisiTombol and JumlahFrame < 1000:
104        waktu = round((time() - waktuStart),2)
105        JumlahFrame += 1
106        # data = str(i) + '\n'
107        # data = str(i).encode('UTF-8')
108        if KeberadaBola:
109            JumlahFrameDeteksi += 1
110
111    if JumlahFrame == 1000:
112        simpanData = True
113        kondisiTombol = False
114        # data = str(0) + '\n'
115        # data = str(0).encode('UTF-8')
116        # arduino.write(data.encode('UTF-8'))
117        presentase = JumlahFrameDeteksi/JumlahFrame*100
118        presentase = "{:.2f}".format(presentase)
119        window['JUMLAHFRAMEDETEKSI'].update(str(JumlahFrameDeteksi))
120        window['JUMLAHFRAME'].update(str(JumlahFrame))
121        window['PRESENTASE'].update(str(presentase))
122    else:
123        print(JumlahFrame, " ", JumlahFrameDeteksi)
124        # data = str(0) + '\n'
125        # arduino.write(data.encode('UTF-8'))

```

```

1 import numpy as np
2 import cv2
3 import math
4
5 def DeteksiObjek(frame, L_H, L_S, L_V, U_H, U_S, U_V, ukuranMin):
6     kernel = np.ones((5,5),np.uint8)
7     hsvImage = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
8     frameDetectCopy = frame.copy()
9     mask = cv2.inRange(hsvImage, (L_H, L_S, L_V), (U_H, U_S, U_V))
10    mask = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel)
11    mask = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kernel)
12    mask_contours_HS, hierarchy = cv2.findContours(mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
13    seleksi_warna = cv2.bitwise_and(frameDetectCopy, frameDetectCopy, mask=mask)
14    Keberadaan = False
15    errorX = 0
16    errorY = 0
17    luas = 0
18    xtarget = 2
19    ytarget = 2
20    wtarget = 2
21    htarget = 2
22    titikTengahx = int(frame.shape[1]/2)
23    titikTengahy = int(frame.shape[0]/2)
24    error = 0
25    cv2.line(frame, (titikTengahx-100, titikTengahy), (titikTengahx+100, titikTengahy), (255, 0, 0), 1)
26    cv2.line(frame, (titikTengahx, titikTengahy-100), (titikTengahx, titikTengahy+100), (255, 0, 0), 1)
27    if len(mask_contours_HS) != 0:
28        for mask_contour_HS in mask_contours_HS:
29            if cv2.contourArea(mask_contour_HS) > ukuranMin:
30                Keberadaan = True
31                x, y, w, h = cv2.boundingRect(mask_contour_HS)
32
33                if luas < abs(w*h):
34                    luas = abs(w*h)
35                    errorX = int((x+w/2))
36                    errorY = int((y+h/2))
37                    xtarget = x
38                    ytarget = y
39                    wtarget = w
40                    htarget = h
41                error = errorX - titikTengahx
42                cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255), 1)
43                cv2.rectangle(frame, (xtarget, ytarget), (xtarget + wtarget, ytarget + htarget), (255, 0, 0), 1)
44                cv2.circle(frame, (errorX, errorY), 3, (255, 0, 0), 1)
45    return seleksi_warna, error, Keberadaan

```

```

1 import csv
2 import matplotlib.pyplot as plt
3 from matplotlib.animation import FuncAnimation
4
5 Simpanwaktu = []
6 Simpannilai = []
7
8 def TampungData(Penyimpananwaktu, waktu, PenyimpananNilai, nilai):
9     Penyimpananwaktu.append(waktu)
10    PenyimpananNilai.append(nilai)
11
12 def SimpanData(Penyimpananwaktu, PenyimpananNilai):
13     with open("PenyimpananData(1).csv", "w", newline="") as csvfile:
14         simpan = csv.writer(csvfile)
15         simpan.writerow(["waktu", "Error"])
16         for waktu, nilai in zip(Penyimpananwaktu, PenyimpananNilai):
17             simpan.writerow([waktu, nilai])
18
19 def PerbaruiPlot(time, valueError):
20     plt.cla()
21     plt.plot(time, valueError)
22     plt.xlabel('time')
23     plt.ylabel('Nilai Error')
24     plt.legend()
25
26 def PlotData(time, valueError):
27     gambar, ax = plt.subplots()
28     funcAnimation(gambar, PerbaruiPlot(time, valueError))
29     plt.show()
30
31

```