PEMANFAATAN SENSOR KAMERA *SMARTPHONE* ANDROID PADA *OBJECT FOLLOWER ROBOT*

Muhammad Resna Rizki Pratama[1], Galih Hermawan[2]

[1,2] Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
E-mail : resnarizki29@gmail.com[1], galih.hermawan@yahoo.com[2]

ABSTRAK

Smartphone saat ini sudah tidak lepas dengan salah satu sensor yaitu kamera. Kamera pada *smartphone* memiliki fungsi utama untuk mengambil foto atau merekam video sehingga dapat mengabadikan setiap momen kehidupan ke dalam memori penyimpanan *smartphone*. Selain dari mengambil foto atau merekam video, kamera pada *smartphone* dapat digunakan dalam hal lainnya seperti *Augmented Reality*, permainan, maupun sebagai sensor untuk pengendalian robot.

Tujuan pada penelitian ini adalah mencoba memanfaatkan kamera pada *smartphone* berbasis Android sebagai pengendalian gerak robot yang akan dibangun dengan mikrokontroler DFRobot. Dengan proses pendeteksian objek yang menggunakan metode *Canny Edge Detection* dan *Contour Extraction*. Data hasil pendeteksian kemudian akan dihitung dengan algoritma kendali PID. Kendali PID diterapkan sebagai pengendali kecepatan motor DC dan gerak setir robot. Sehingga akhirnya didapatkan nilai keluaran berupa nilai kecepatan untuk motor DC pada sisi kiri dan kanan robot serta arah putaran motor DC yang akan dikirimkan ke mikrokontroler melalui *Bluetooth*.

Berdasarkan hasil pengujian yang dilakukan, robot dapat mengikuti benda dengan persentase keberhasilan sistem sebesar 77%.

Kata kunci: Robot, Kamera, Smartphone, Android, Mikrokontroler, Citra, Kontrol, PID, *Canny*, *Contour*.

1. PENDAHULUAN

Melihat dari sejarah, perkembangan teknologi beberapa tahun terakhir melesat sangat cepat. Beberapa di antaranya ada yang sudah menjadi alat pendukung produktivitas atau sudah menjadi bagian dari kehidupan manusia. Mulai dari *smartphone* sampai robot-robot canggih yang menjadi pendukung produktivitas sehari-hari. Robot saat ini sudah bisa melakukan beberapa pekerjaan manusia maupun pekerjaan yang sulit dilakukan secara cepat oleh manusia. Beberapa contoh robot seperti *Mobile Robot*, *Industrial Robot*, *Humanoid*, *Hobby and Competition Robot*, dsb. Fungsi dari robot yang diciptakan pun semakin beragam mulai dari membantu proses produksi produk, militer, asisten pribadi, permainan, maupun untuk keperluan kompetisi.

Kecanggihan robot tak terlepas dari beberapa komponen yang saling mendukung satu sama lain. Dari beberapa komponen tersebut terdapat salah satu yang merupakan alat indra dari robot tersebut, yaitu sensor. Sensor yang bisa ditemukan di pasaran untuk membuat sangat banyak macamnya. Untuk penglihatan sebuah robot, bisa menggunakan kamera. Salah satu informasi dasar yang bisa didapatkan berupa nilai *Red – Green – Blue* (RGB) dari setiap kumpulan titik yang membangun gambar tersebut. Kemudian informasi tersebut bisa diproses lebih lanjut agar bisa diperoleh informasi lainnya. Pada tahap selanjutnya, bisa dilakukan proses *filtering* warna, pendeteksian objek, ataupun menggunakan metode pendeteksian lainnya untuk bisa mendapatkan informasi yang ingin diperoleh dari gambar tersebut.

Pada penelitian yang telah dilakukan oleh Ratnasari Nur Rohmah dan Laksono Budi Prianggodo [1], menerapkan sistem pendeteksian objek untuk menggerakkan robot *mobile*. Menggunakan modul kamera *Raspberry PI* untuk menangkap gambar yang kemudian gambar tersebut akan diproses oleh *board Raspberry PI*. Metode yang digunakan untuk mendeteksi objek berbentuk lingkaran yang terdapat dalam gambar yang tertangkap modul kamera adalah *Circle Hough Transform* dan *filtering* warna *HSV*. Kemudian, dari hasil pendeteksian tersebut *Raspberry PI* mengirimkan sinyal perintah ke motor DC untuk menggerakkan robot.

Pada penelitian tugas akhir ini akan menggunakan kamera yang terdapat pada *smartphone* berbasis Android. Proses pendeteksian objek maupun penginstruksian perintah gerak robot *mobile* juga akan dilakukan oleh aplikasi yang dijalankan pada *smartphone* Android. Selain itu, juga terdapat board *DFRobot* digunakan untuk mengontrol motor DC. Komunikasi data antara *smartphone* Android dan *board* DFRobot Romeo BLE menggunakan *Bluetooth 4.0*. Adapun metode pendeteksian objek yang akan digunakan pada penelitian ini adalah *Canny Edge Detection* dan

Contour Extraction. Maka penelitian tugas akhir ini mengambil judul "**Pemanfaatan Sensor Kamera *Smartphone* Android pada *Object Follower Robot***".

1.1 Batasan Masalah

Dalam penelitian ini dibatasi dengan beberapa batasan penelitian agar ruang lingkup permasalahan tidak melebar sehingga menjadi lebih jelas dan terarah. Batasan penelitian ini adalah sebagai berikut:

1. Robot yang digunakan merupakan robot mobil yang menggunakan mikrokontroler DFRobot Romeo BLE dan motor DC sebagai penggerak utamanya.
2. Smartphone akan ditempatkan pada bagian atas robot.
3. Smartphone yang digunakan menggunakan sistem operasi Android.
4. Smartphone memiliki kamera 13MP (mega pixel).
5. Pengiriman data antara smartphone dan mikrokontroler menggunakan media nirkabel Bluetooth 4.0.
6. Perancangan perangkat lunak pada smartphone Android menggunakan Android Studio dengan menggunakan bahasa pemrograman Java dan C++. Bahasa C++ digunakan untuk melakukan pemrosesan gambar.
7. Menggunakan Android Native Development Kit (NDK) agar Android Studio dapat mengompilasi kode C++.
8. Pemrosesan data akan dilakukan pada aplikasi yang akan dibangun untuk smartphone Android.
9. Mikrokontroler hanya akan menerima data kecepatan motor dan menggerakkan motor DC.
10. Pemrosesan gambar dilakukan dengan menggunakan library OpenCV Android.
11. Perancangan perangkat lunak pada mikrokontroler menggunakan Arduino IDE.
12. PID Controller digunakan untuk mengontrol perbedaan kecepatan motor bagian kiri dan kanan robot agar bisa berbelok kiri dan kanan dan mengatur kecepatan setir motor.
13. Pendeteksian objek akan menggunakan metode Canny Edge Detection dan Contour Extraction.
14. Objek yang akan digunakan untuk dideteksi yaitu bola pingpong berwarna oranye.
15. Robot tidak menggunakan servo motor untuk menggerakan smartphone Android.

1.2 Maksud dan Tujuan

Maksud dari penelitian yang akan dilakukan adalah membuat robot bergerak otomatis mengikuti objek yang terdeteksi oleh kamera *smartphone* Android.

Adapun tujuan yang ingin dicapai pada penelitian ini adalah menguji sejauh mana *smartphone* Android mampu mendeteksi objek dengan menggunakan metode *Canny Edge Detection dan Contour Extraction* serta menguji apakah robot dapat mengikuti objek dengan baik.

1.3 Citra

Definisi dari Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek dalam bentuk 2 dimensi (2D) yang mempunyai panjang dan lebar. Citra terdapat dua macam yaitu citra diskrit dan citra kontinu. Citra kontinu dihasilkan dari sistem optik yang menerima sinyal analog, misalnya mata manusia dan kamera analog. Citra diskrit dihasilkan melalui proses digitalisasi terhadap citra kontinu [2].

1.4 Robot

Kata robot pertama kali diperkenalkan oleh Karel Capek pada sebuah pentas sandiwara yang diadakan pada tahun 1921 dalam pertunjukkannya R.U.R. (Rossum's Universal Robots). Kata robot itu sendiri berasal dari kata robota dalam bahasa Ceko yang berarti budak, pekerja, atau kuli. Pengertian lain dari robot adalah suatu alat yang terdiri dari sistem mekanik, sistem elektronik, dan sistem kendali yang berperilaku seperti manusia yang memudahkan pekerjaan manusia. Dari berbagai definisi yang ada, definisi paling dapat diterima adalah dari "Robot Institute of America", yang mendefinisikan robot sebagai manipulator yang berfungsi jamak yang dapat diprogram ulang dan dirancang untuk memindahkan benda kerja, komponen-komponen, peralatan atau perangkat khusus dengan berbagai macam tugas [3].

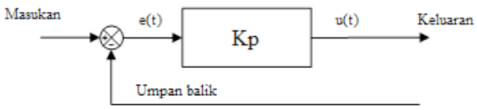
1.5 Arduino

Arduino adalah salah satu mikrokontroler single-board yang bersifat open source. Arduino dikembangkan oleh sebuah tim yang beranggotakan orang-orang dari berbagai belahan dunia. Anggota inti dari tim ini adalah Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis, dan Nicholas Zambetti [4]. Arduino dirancang agar bisa memudahkan penggunaannya pembuatan prototype maupun penggunaan elektronik dalam berbagai bidang.

Pada board Arduino terdapat komponen utama yaitu sebuah chip mikrokontroler dengan jenis AVR. Mikrokontroler itu sendiri adalah chip atau IC (integrated circuit) yang bisa diprogram langsung melalui komputer dari Arduino IDE. Tujuan menanamkan program pada mikrokontroler adalah agar rangkaian elektronik dapat membaca input, memproses input tersebut dan kemudian menghasilkan output seperti yang diinginkan.

1.6 Kendali Proporsional

Kendali proporsional berfungsi untuk memperkuat nilai kesalahan penggerak (nilai error). Kendali proporsional akan mengeluarkan nilai keluaran yang berbanding lurus dengan nilai kesalahan (selisih antara nilai aktual dengan nilai yang diinginkan) [5].



Gambar 1. Diagram blok Kendali Proporsional

Gambar 1 Diagram blok Kendali Proporsional

Hubungan antara nilai keluaran kendali proporsional u(t) dengan sinyal error e(t) dibuat oleh persamaan:

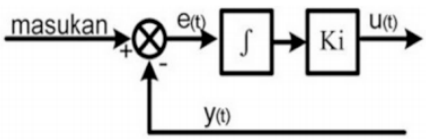
$$u(t) = K_p e(t) \qquad (1)$$

Ciri-ciri kendali proporsional dapat dijelaskan sebagai berikut.

1. Jika nilai K*P* kecil maka kendali proporsional hanya mampu melakukan koreksi kesalahan yang kecil, sehingga akan menghasilkan respons sistem yang lambat.
2. Jika nilai K*P* dinaikkan, respons sistem menunjukkan semakin cepat mencapai keadaan mantapnya.
3. Jika nilai K*P* diperbesar mencapai harga yang berlebihan, akan mengakibatkan sistem bekerja tidak stabil, atau sistem akan berosilasi.

1.7 Kendali Integral

Kendali integral akan menghilangkan kesalahan keadaan tunak (offset) yang dihasilkan oleh kendali proporsional. Kendali integral akan menghasilkan nilai yang berbanding lurus dengan besar dan lamanya error. Kendali integral mempercepat perpindahan proses menuju nilai setpoint (nilai yang diinginkan) dan menghilangkan steady-state error yang muncul pada kendali proporsional. Namun karena kendali integral merespons terhadap nilai error yang terakumulasi sebelumnya, maka dapat menyebabkan overshoot. Kendali integral memiliki karakteristik seperti halnya sebuah integral. Keluaran pengendali sangat dipengaruhi oleh perubahan yang sebanding dengan nilai sinyal kesalahan. Keluaran pengendali ini merupakan jumlahan yang terus menerus dari perubahan masukannya. Kalau sinyal kesalahan tidak mengalami perubahan, keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan [5].



Gambar 2. Diagram blok Kendali Integral

Diagram blok di atas dapat diwujudkan dalam bentuk persamaan yang menunjukkan hubungan antara nilai keluaran kendali integral u(t) dengan sinyal error e(t) sebagai berikut:

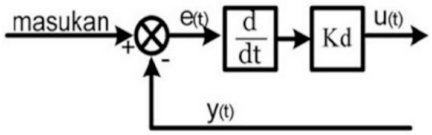
$$u(t) = K_i \int_0^t e(t)\, dt \qquad (2)$$

Ciri-ciri kendali Integral dijelaskan sebagai berikut.

1. Keluaran pengendali membutuhkan selang waktu tertentu sehingga kendali integral cenderung memperlambat respons.
2. Ketika sinyal kesalahan berharga nol, keluaran pengendali akan bertahan pada nilai sebelumnya.
3. Jika sinyal kesalahan tidak berharga nol, keluaran akan menunjukkan sinyal kesalahan dan nilai Ki.
4. Kendali integral Ki yang berharga besar akan mempercepat hilangnya offset. Tetapi semakin besar nilai konstanta Ki akan mengakibatkan peningkatan osilasi dari sinyal keluaran pengendali.

1.8 Kendali Derivatif

Keluaran kendali derivatif memiliki sifat seperti halnya suatu operasi derivatif. Perubahan yang mendadak pada masukan pengendali akan mengakibatkan perubahan yang sangat besar dan cepat juga pada keluaran [5].



Gambar 3. Diagram blok Kendali Derivatif

Diagram blok di atas diwujudkan dalam bentuk persamaan yang menunjukkan hubungan antara nilai keluaran kendali integral u(t) dengan sinyal error e(t) sebagai berikut:

$$u(t) = K_d \frac{de(t)}{dt} \qquad (3)$$

Ciri-ciri kendali derivatif sebagai berikut

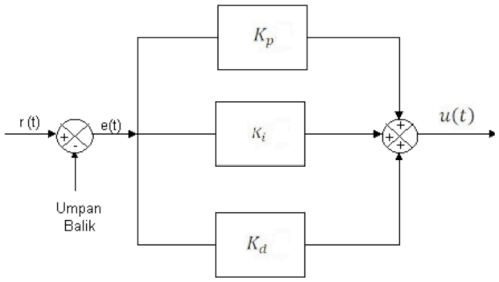
1. Pengendali ini tidak dapat menghasilkan keluaran bila tidak ada perubahan pada masukannya (berupa sinyal kesalahan).
2. Jika sinyal kesalahan berubah terhadap waktu, maka keluaran yang dihasilkan

pengendali tergantung pada nilai Kd dan laju perubahan sinyal kesalahan.

3. Kendali derivatif mempunyai karakteristik untuk mendahului, sehingga pengendali ini dapat menghasilkan koreksi yang signifikan sebelum pembangkit kesalahan menjadi sangat besar. Jadi kendali derivatif dapat mengantisipasi pembangkit kesalahan, memberikan aksi yang bersifat korektif, dan cenderung meningkatkan stabilitas sistem.

1.9 Kendali PID

Kendali PID merupakan gabungan dari kendali Proporsional, kendali Integral, dan kendali Derivatif. Elemen-elemen kendali P, I dan D masing-masing secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem, menghilangkan offset dan menghasilkan perubahan awal besar.



Gambar 4. Diagram blok Kendali PID

Dari diagram blok di atas dapat diperoleh persamaan:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)\, dt + K_d \frac{de(t)}{dt} \qquad (4)$$

2. ISI PENELITIAN

2.1 Analisis Masalah

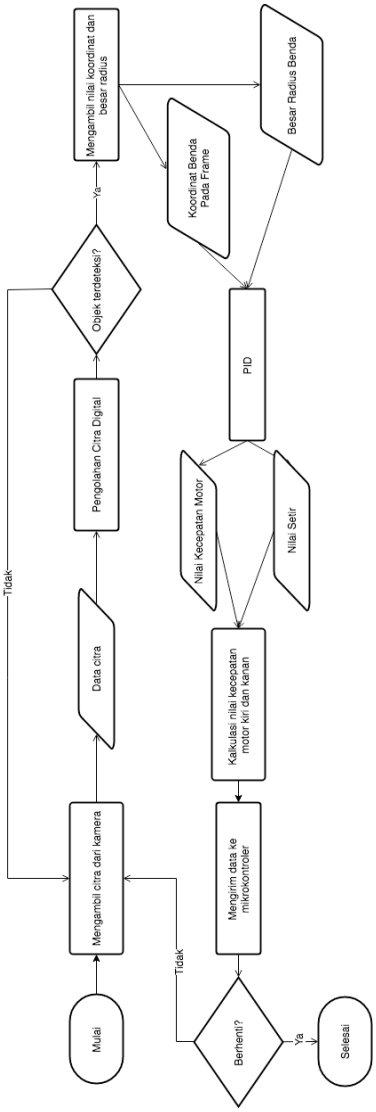
Berdasarkan hasis analisis dari penelitian sebelumnya yaitu "Rancang Bangun Robot Beroda dengan Object Tracking Sebagai Dasar Pengendalian Gerakan Robot" oleh Ratnasari Nur Rohmah dan Laksono Budi Prianggodo dari Universitas Muhammadiyah Surakarta didapatkan bahwa resolusi dari kamera yang digunakan akan mempengaruhi hasil pendeteksian objek. Semakin besar resolusi yang digunakan akan semakin jauh jarak objek bisa dideteksi oleh robot.

Berdasarkan uraian di atas, pada penelitian ini penulis akan membuat robot pengikut objek (Object Follower Robot) dengan memanfaatkan kamera yang berada pada smartphone berbasis Android. Kamera pada smartphone Android yang akan digunakan mempunya kamera yang mampu mengambil citra berkualitas 12MP (mega pixel). Sedangkan kamera yang digunakan pada penelitian sebelumnya yaitu modul kamera Raspberry Pi yang hanya bisa mengambil citra berkualitas 5MP.

2.2 Analisis Solusi

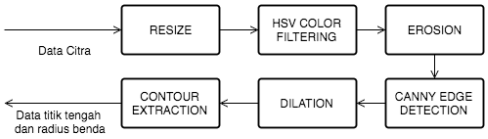
Penyelesaian masalah akan dilakukan dengan cara melakukan beberapa proses pendeteksian dan kemudian dari hasil pendeteksian tersebut akan didapatkan koordinat tengah dan nilai besar kontur benda yang telah terdeteksi. Dari koordinat tengah benda dan nilai besar kontur benda tersebut akan diproses lebih lanjut menggunakan algoritma pengendali PID untuk didapatkan nilai setir robot dan nilai kecepatan DC.

Adapun gambaran umum dari sistem robot yang akan dibangun adalah sebagai berikut:



Gambar 5. Gambaran Umum Sistem

2.3 Analisis Proses Pendeteksian



Gambar 6. Alur Proses Pendeteksian

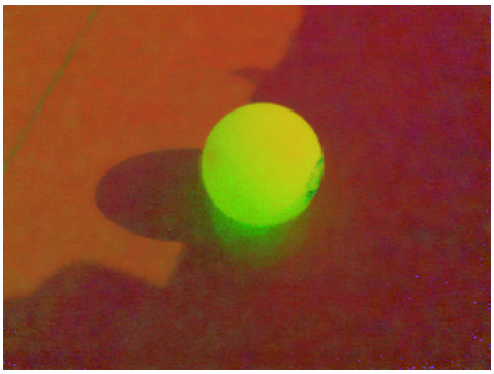
Citra mentah yang langsung diambil dari kamera memiliki ukuran resolusi yang sangat besar.

Besarnya resolusi akan berpengaruh pada penggunaan memori serta lama pemrosesan citra. Untuk itu perlu dilakukan downscaling atau megubah ukuran resolusi gambar menjadi lebih kecil. Pada tahap pertama citra yang diambil akan dikecilkan ukurannya menjadi 320x240 pixel.



Gambar 7. Citra yang Sudah Dikecilkan

Tahap selanjutnya adalah mengubah ruang warna RGB (Red-Green-Blue) ke ruang warna HSV (Hue-Saturation-Value). Proses ini dilakukan untuk menyegmentasikan warna secara mudah.



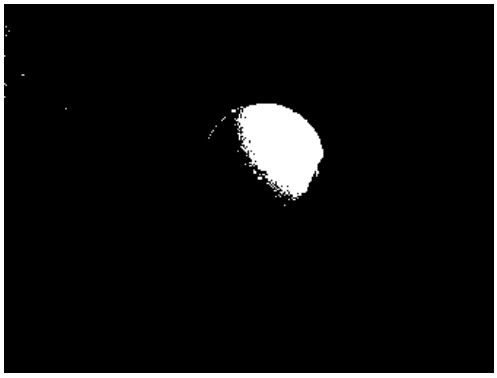
Gambar 8. Hasil Konversi ke Ruang Warna HSV

Setelah warna citra dikonversi ke ruang warna HSV, maka tahapan selanjutnya adalah melakukan segmentasi atau *color filtering*. Pada penelitian ini, benda yang akan digunakan untuk dideteksi yaitu bola pingpong berwarna oranye. Sebagai contoh untuk nilai minimum atas maksimum HSV yang akan digunakan yaitu:

$HSV_{MIN} = (0, 125, 150)$

$HSV_{MAX} = (60, 255, 255)$

Kemudian nilai HSV dari setiap *pixel* pada citra akan diseleksi berdasarkan batasan nilai HSV tersebut. Maka hasil yang akan didapatkan adalah citra biner dari setiap *pixel* yang nilainya berada di dalam batasan nilai HSV



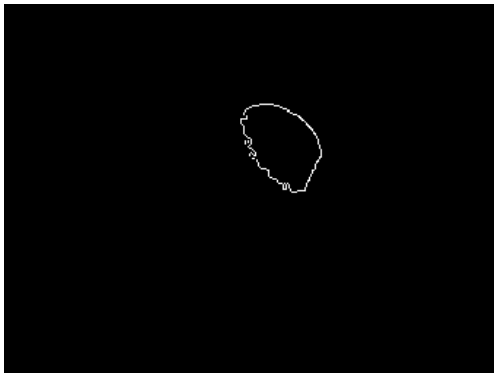
Gambar 9. Hasil Filter Warna

Citra biner dari hasil segmentasi masih didapatkan derau (*noise*) di dalamnya. Terlihat titik-titik berukuran kecil warna putih yang berada di luar area bola. Untuk menghilangkan *noise* tersebut, dilakukan proses erosi(*erosion*). Karena proses *erosion* ini mengkikis pixel, make setiap titik-titik kecil tersebut akan terkikis dan menghilang dari citra.



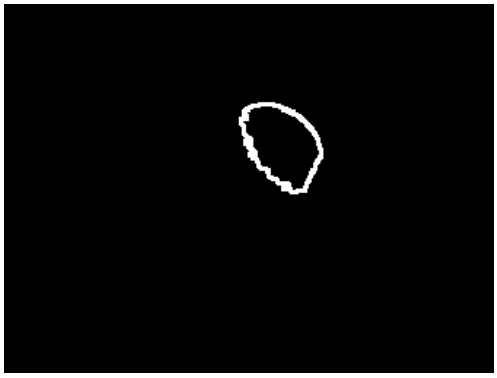
Gambar 10. Citra Setelah Proses *Erosion*

Setelah itu maka dilakukan proses pendeteksian tepian dengan metode *Canny Edge Detection*.



Gambar 11. Hasil Pendeteksian Garis Pinggir

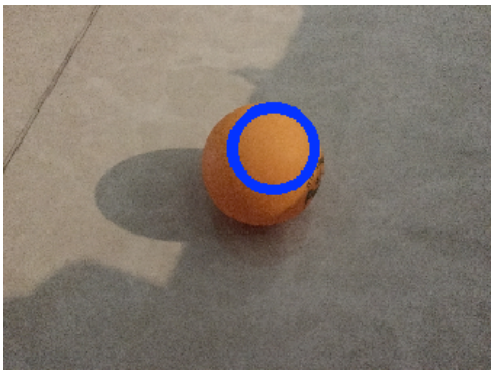
Setelah didapatkan citra dengan hasil garis tepi yang terdeteksi maka dilakaukan proses dilasi (*dilation*) untuk memperjelas garis tersebut.



Gambar 12. Citra Setelah Proses Dilation

Setelah itu, maka dilanjutkan dengan proses ekstraksi kontur (contour extraction). Proses ini akan dibagi menjadi beberapa tahap yaitu:

1. Pencarian kontur.
2. Seleksi kontur berdasarkan besar area kontur.
3. Ambil nilai koordinat tengah dan radius kontur.



Gambar 13. Hasil Akhir Pendeteksian

Setelah didapatkan nilai koordinat tengah, dan nilai radius maka proses pendeteksian telah selesai. Sebagai simulasi pendeteksian dari nilai-nilai tersebut bisa digunakan untuk menggambar lingkaran seperti yang terlihat pada Gambar 13. Kemudian nilai-nilai tersebut kemudian akan digunakan pada proses selanjutnya dengan algoritma PID Controller.

2.4 Analisis Algoritma PID Controller

PID Controller atau kendali PID pada sistem berperan penting untuk menggerakkan robot. Kendali PID akan mengatur kecepatan dari robot yang akan dibangun. Hasil keluaran dari algoritma kendali PID yaitu nilai setir dan nilai maksimum kecepatan motor DC. Kendali PID bekerja dengan diberikan nilai setpoint. Nilai setpoint ini adalah nilai acuan yang kita inginkan. Jarak posisi benda dan adalah sebuah error. Error merupakan nilai simpangan antara variabel terukur dengan nilai acuan. Kamera pada smartphone Android digunakan sebagai elemen feedback untuk mengumpanbalikan error.

Kendali PID merupakan kendali yang menggabungkan kelebihan dari kendali P, I dan D. Jika diterapkan pada sistem kendali PID akan memperbaiki respons, menghilangkan error steady state, dan memberikan efek redaman pada sistem yang berosilasi.

Persamaan kendali PID diberikan oleh persamaan (5) sebagai berikut :

$$u(t) = K_p e(t) + K_i \int_0^t e(t)\,dt + K_d \frac{de(t)}{dt} \qquad (5)$$

Jika persamaan (6) didiskretisasi maka akan menjadi :

$$u(t) = K_p e(t) + K_i T \sum_0^t e_t + \frac{1}{T} K_d (e_t - e_{t-1}) \qquad (6)$$

Apabila diterapkan ke dalam bahasa pemrograman maka persamaannya menjadi :

$$output = K_p * error + K_i (error + last_error) * Ts + \frac{K_d}{Ts} (error - last_error) \qquad (7)$$

2.5 Analisis Perhitungan Kecepatan Motor DC Kiri dan Kanan

Pada tahapan ini, nilai setir dan nilai kecepatan maksimum motor DC yang dikeluaran dari algoritma kendali PID akan dihitung kembali untuk didapatkan nilai kecepatan untuk motor DC sebelah kiri dan kanan robot. Nilai kecepatan dibedakan karena jenis robot mobile yang akan dibangun berupa robot mobil yang pergerakan kiri dan kanannya diatur dari kecepatan motor DC pada sebelah kiri dan kanan robot. Untuk dapat berbelok ke kiri, motor sebelah kiri harus bergerak lebih lambat dari pada motor sebelah kanan. Sebaliknya, untuk dapat berbelok ke kanan, motor kanan harus bergerak lebih lambat dari pada motor sebelah kiri. Namun jika ingin bergerak lurus, maka motor di kedua sisi harus bergerak sama.

Untuk mendapatkan nilai kecepatan motor kiri dapat menggunakan persamaan (8) dan nilai kecepatan motor kanan dapat menggunakan persamaan (9).

$$LS[s,v] = \begin{cases} v, & s \geq 0 \text{ and } v \geq 0 \\ |v| - \left(|v| \left(\frac{|s|}{100}\right)\right), & (s < 0 \text{ and } v \geq 0) \text{ or } (s > 0 \text{ and } v < 0) \end{cases} \qquad (8)$$

$$RS[s,v] = \begin{cases} v, & s \leq 0 \text{ and } v \geq 0 \\ |v| - \left(|v| \left(\frac{|s|}{100}\right)\right), & (s > 0 \text{ and } v \geq 0) \text{ or } (s < 0 \text{ and } v < 0) \end{cases} \qquad (9)$$

S merupakan nilai setir dan V merupakan nilai kecepatan maksimum motor DC. Jika dihitung dengan kedua rumus tersebut maka motor DC akan melambat sesuai dengan nilai setir yang diberikan. Sehingga, robot akan dapat berbelok ke kiri dan ke kanan.

Selanjutnya untuk arah perputaran motor kiri dan kanan digunakan persamaan (10).

$$D[v] = \begin{cases} Maju, & v \geq 0 \\ Mundur, & v < 0 \end{cases} \quad (10)$$

2.6 Analisis Serialisasi Data

Data kecepatan dan arah perputaran motor DC akan dikirimkan ke mikrokontroler melalui Bluetooth. Sebelum melakukan pengiriman data, data harus diserialisasikan agar dapat dikirim. Jumlah data yang akan dikirimkan terdapat 4 yaitu:

Tabel 1. Data Kecepatan dan Arah Putaran Motor DC

No	Nama Data	Tipe Data	Nilai
1	MotorKiriSpeed	Integer	0-255
2	MotorKiriDir	Boolean	1/0 (Maju/Mundur)
3	MotorKananSpeed	Integer	0-255
4	MotorKananSpeed	Boolean	1/0 (Maju/Mundur)

Keempat data tersebut akan dikirimkan ke mikrokontroler secara bersamaan dengan tipe data byte. Sebagai contoh data yang akan dikirimkan adalah:

Tabel 2. Contoh Data Kecepatan dan Arah Putaran Motor DC

No	Nama Data	Nilai
1	MotorKiriSpeed	150
2	MotorKiriDir	1
3	MotorKananSpeed	150
4	MotorKananSpeed	1

Maka keempat data tersebut akan diubah menjadi tipe data byte menjadi:

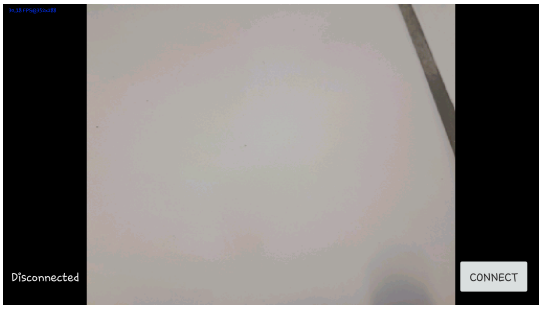
Tabel 3. Konversi ke Tipe Data Byte

No	Nama Data	Nilai
1	MotorKiriSpeed	0x00000096
2	MotorKiriDir	0xFF
3	MotorKananSpeed	0x00000096
4	MotorKananSpeed	0xFF

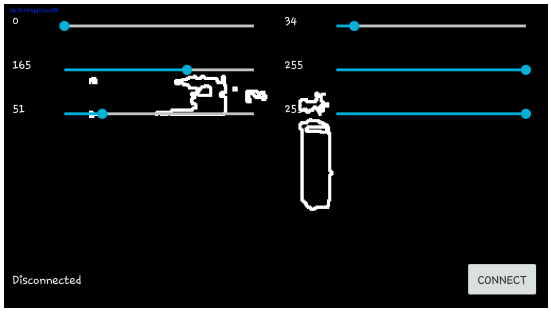
Seluruh nilai byte tersebut akan dikumpulkan menjadi sebuah larik tipe byte sehingga data yang dikirim akan berbentuk seperti [00, 00, 00, 96, FF, 00, 00, 00, 00 96, FF].

2.7 Implementasi Antarmuka

Antarmuka yang telah diimplementasikan pada sistem yaitu:



Gambar 14. Implementasi Tampilan Utama



Gambar 15. Implementasi Tampilan Pengaturan

2.8 Pengujian

Tabel 4. Pengujian Mendeteksi Benda

No	Butir Pengujian	Kesimpulan
1	Mengarahkan bola pingpong bewarna oranye	[√] Diterima [ ] Ditolak
2	Mengarahkan bola pingpong bewarna oranye dengan kondisi pencahayaan gelap.	[√] Diterima [ ] Ditolak
3	Mengarahkan bola pingpong bewarna oranye dengan jarak jauh	[√] Diterima [ ] Ditolak
4	Mengarahkan bola pingpong bewarna oranye dan di bilakang terdapat benda dengan warna yang sejenis.	[ ] Diterima [√] Ditolak

Tabel 4. Pengujian Menghitung Kecepatan Motor

No	Butir Pengujian	Kesimpulan
1	Mengarahkan bola pingpong sangat jauh	[√] Diterima [ ] Ditolak
2	Mengarahkan bola pingpong dekat	[ ] Diterima [√] Ditolak
3	Mengarahkan bola pingpong sangat dekat	[√] Diterima [ ] Ditolak

3. PENUTUP

3.1 Kesimpulan

Dari hasil penelitian, analisis, perancangan, implementasi dapat diambil kesimpulan sebagai berikut:

1. Sistem dapat mendeteksi benda yaitu bola pingpong berwarna oranye dalam kondisi normal.
2. Sistem tidak sepenuhnya dapat mendeteksi benda ketika di sekitar benda yang memiliki warna yang mirip dengan warna benda.
3. Sistem mendeteksi benda non lingkaran sebagai lingkaran jika warna benda masuk dalam batas warna HSV yang telah diatur.
4. Sistem mampu memberikan nilai kecepatan berdasarkan jarak benda.
5. Sistem masih kesulitan untuk dapat berhenti ketika benda berada di dekat robot.
6. Robot bisa mengikuti benda yang terdeteksi.
7. Bingkai citra yang digunakan masih terlalu kecil untuk dapat mendeteksi benda. Sehingga robot tidak bisa mengikuti benda ketika benda tiba-tiba berubah posisi.

3.2 Saran

Pembangunan sistem pada penelitian ini masih terdapat banyak kekurangan. Oleh karena itu perlu dilakukan pengembangan dan penyempurnaan lebih lanjut. Adapun saran yang mungkin bisa diterapkan pada pembangunan tahap selanjutnya adalah sebagai berikut:

1. Menggunakan metode pendeteksian benda yang lebih handal.
2. Menerapkan kecerdasan buatan lainnya seperti pengenalan objek agar sistem dapat leluasa mendeteksi benda yang diinginkan.
3. Menambahkan motor servo untuk menggerakkan smartphone Android agar kamera dapat mengikuti bola secara leluasa sehingga area pendeteksian benda semakin luas.
4. Menggunakan sensor tambahan seperti sensor ultrasonik untuk mendeteksi daerah sekitar robot.
5. Menggunakan media kabel usb untuk perpindahan data antara sistem dengan mikrokontroler semakin cepat dibandingkan menggunakan media Bluetooth.

DAFTAR PUSTAKA

[1] R. N. Rohmah dan L. B. Prianggodo, "Rancang Bangun Robot Beroda dengan Object Tracking Sebagai Dasar Pengendalian Gerakan Robot," *Jurnal PROtek,* vol. 03, no. 1, September 2016.

[2] R. Munir, Pengolahan Citra Digital, Informatika, 2007.

[3] "Robot," [Online]. Available: https://en.wikipedia.org/wiki/Robot. [Diakses 10 03 2017].

[4] F. Djuandi, Pengenalan Arduino, www.tokobuku.com, 2011.

[5] "Realisasi Kontrol PID Proporsional," [Online]. Available: http://elektro-kontrol.co.id/2011/06/realisasi-kontrol-pid-proporsional.html. [Diakses 03 2017].