

Tracking Bola Menggunakan Robotino

Gentang Syabba Nahla¹, Setiawardhana, S.T.², Ir. Dadet Pramadihanto, M.Eng, Ph.D²
Mahasiswa¹, Dosen²
Politeknik Elektronika Negeri Surabaya
Institut Teknologi Sepuluh Nopember
Kampus PENS-ITS Keputih Sukolilo Surabaya 60111
Telp (+62)31-5947280, 5946114, Fax. (+62)31-5946114
Email : gt.comsby@gmail.com

Abstrak

Dalam tugas akhir ini, digunakan Robotino sebagai piranti kerasnya dan sebagai piranti lunaknya dibuatlah program menggunakan bahasa pemrograman bahasa C++ dan *library open source computer vision* sehingga Robotino dapat mendeteksi bola dengan metode *image processing* menggunakan *webcam* yang terpasang pada Robotino, lalu melakukan *tracking* untuk menangkap dan menggiring bola menggunakan gripper yang terpasang pada Robotino kemudian membawa bola ke arah gawang. Proses *image processing* digunakan untuk mendeteksi obyek bola dan gawang, sehingga dari proses ini didapatkan data mengenai titik tengah obyek dan luasan obyek yang terdeteksi. Dari data yang didapatkan tersebut digunakan sebagai inputan variabel pada perhitungan Motor Controller Robotino sehingga akan didapatkan hasil keputusan nilai yang diperlukan dalam mengatur kecepatan gerak motor Robotino pada saat mendekati obyek.

Kata Kunci : *Robotino, deteksi, deteksi bola, pengolahan citra, pelacakan*

Abstract

In this final project, we used Robotino as a hardware tool and as the software is made programs using C++ programming languages and open source computer vision library, so Robotino can detect the ball with image processing method using webcam installed on Robotino, and then do tracking to catch and lead ball using a gripper attached to Robotino then carried the ball toward the goal. The process of image processing is used to detect the ball and goal, so from this process obtained data about the midpoint of the object and extent of the detected object. From the data obtained is used as input variables in the calculation of Robotino Motor Controller so we will get the decision value that is required to regulate the velocity of the Robotino motor when approaching an object.

Keyword : Robotino, tracking, ball tracking, image processing

1. Pendahuluan

1.1 Latar Belakang

Tracking objek bisa dipahami sebagai sebuah contoh kasus yang cukup istimewa dari sekian banyak permasalahan umum yang dipelajari dalam bidang computer vision. Perkembangan tingginya kemampuan dari suatu komputer saat ini dan meningkatnya kebutuhan akan analisa video dilakukan secara otomatis yang telah banyak sekali menghasilkan sesuatu yang hebat dan menarik dalam algoritma tracking objek.

Penggunaan tracking objek ini merupakan sebuah permasalahan yang penting bagi sejumlah aplikasi yang

mungkin bisa menguntungkan dalam penerapannya, seperti traffic monitoring, automated surveillance, sistem navigasi suatu kendaraan, terutama dalam bidang robotika yaitu mobile robot, robotic soccer dan masih banyak lagi yang bisa diaplikasikan.

Sudah diketahui dengan hasil yang baik bahwa vision digunakan sebagai sebuah sensor robot yang handal dikarenakan kemampuannya untuk nonkontak dan penghitungan langsung dari lingkungan dan juga ketentuan dari beberapa informasi

Pada proyek akhir ini kami akan membuat sistem yang bertujuan dalam penggunaan sebuah robot untuk melakukan tracking / pelacakan sebuah bola lalu mengikutinya pada jarak yang bisa berubah-ubah.

Motivasi pada aplikasi khusus ini dikembangkan sebagai solusi dan perilaku yang baru untuk sebuah lingkup Robot Sepakbola. Dalam sistem robot ini kami menggunakan kamera sebagai sensor mata dari robot tersebut

Setiap prosedur tracking dibagi dalam dua blok utama yang berperan dalam mencari objek dan mengikutinya. Yang pertama adalah permasalahan deteksi objek yang bisa diselesaikan melalui pemrosesan data sensor (kamera). Sedangkan untuk tugas yang kedua adalah mengenai pengendalian.

1.2 Rumusan Permasalahan

Berdasarkan uraian tersebut di atas, dalam pengerjaan proyek akhir ini timbul beberapa masalah diantaranya adalah :

1. Cara menentukan obyek mana yang harus dideteksi.
2. Cara menemukan obyek yang dimaksud sebagai target jika terdapat benda lain di sekitarnya.
3. Cara menemukan posisi target sehingga target bisa masuk dalam pandangan kamera.
4. Cara mengontrol pergerakan tracking robot sehingga target tetap terlacak.

1.4 Tujuan Proyek

Tujuan dari proyek akhir ini adalah membuat program dalam penggunaannya pada sebuah robotino untuk melakukan tracking sebuah bola dan gawang, dan bola tersebut digiring untuk dimasukkan ke gawang. Motivasi pada program khusus ini dikembangkan sebagai solusi dan perilaku yang baru untuk sebuah lingkup Robot Sepakbola. Dalam sistem robot ini menggunakan kamera sebagai sensor mata dari robot tersebut.

1.3 Penelitian Terkait

Pada proyek akhir sebelumnya mempunyai judul “Robot Sepakbola dengan Robotino”. Pada prinsipnya secara teknis hampir sama dengan proyek akhir yang sedang dikerjakan oleh penulis, namun perbedaannya terletak pada program yang dibuat. Yaitu, jika pada pembuatan proyek akhir sebelumnya si penulis menggunakan Software Robotino View yang sudah disediakan oleh Perusahaan Festo Didactic yang merupakan pembuat Robotino sekaligus pembuat software Robotino View tersebut. Sedangkan pada proyek akhir yang sedang dikerjakan penulis saat ini, yaitu tidak menggunakan software Robotino View tersebut tetapi membuat dari awal dengan bahasa pemrograman yang hasilnya nanti bisa bekerja seperti proyek akhir sebelumnya yang menggunakan software Robotino View. Untuk lebih jelasnya seperti apakah penggunaan Robotino View pada proyek akhir sebelumnya akan dijelaskan pada penjelasan sub bab berikut.

1.5 Kontribusi Proyek

Proyek Akhir ini nantinya diharapkan dapat dikembangkan untuk membantu menyelesaikan permasalahan pada robotino sehingga nantinya bisa berpartisipasi dalam kontes robot sepakbola yang menggunakan robotino.

2. Teori Penunjang

2.1 Konsep Sistem *Object Tracking*

Tugas dari suatu penangkapan / deteksi obyek didefinisikan sebagai mengenali suatu obyek tertentu, sedangkan *tracking* / pelacakan didefinisikan sebagai mendekati sebuah obyek (juga disebut target) sampai cocok lokasinya. Kedua tugas tersebut memiliki kegunaan dalam lingkup yang luas dari suatu aplikasi seperti contohnya sebagai robot sepakbola dan *automated surveillance*.

Untuk kasus *tracking*, terdapat dua kasus yang bisa timbul. Pada kasus pertama, ketika gerakan target diketahui, permasalahannya yaitu untuk merencanakan lintasan robot bergerak menuju ke titik tempat yang sesuai. Titik pertemuan *custom-defined* pada beberapa jarak kritis dari obyek. Pada kasus kedua, ketika lintasan dari target obyek tidak diketahui di depannya, maka perlu bergantung pada beberapa jenis estimasi dari posisi dan kecepatan yang datang dari data sensor.

2.2 Pengolahan Citra

Pengolahan citra adalah suatu metode yang digunakan untuk mengolah gambar sehingga menghasilkan gambar yang sesuai dengan yang diinginkan kita. Pengambilan gambar bisa dilakukan dengan menggunakan kamera atau alat lain yang bisa digunakan untuk mentransfer gambar misalnya scanner atau kamera digital.

Bahasan kali ini berfokus pada pengambilan gambar menggunakan web kamera. Sehingga citra yang dihasilkan sudah berbentuk sinyal digital dan mudah dikenali atau dibaca komputer.

Citra digital adalah citra kontinyu yang sudah didiskritkan baik koordinat spasial maupun kecerahannya. Citra digital dianggap matrik dengan ukuran $M \times N$ dimana baris dan kolom menunjukkan titik-titiknya. Citra berwarna menggunakan metode RGB. Adapun masing-masing warna dalam tabel memiliki 3 buah kombinasi angka yaitu R, G, dan B yang menentukan proporsi warna merah, warna hijau dan warna biru dari warna tersebut. RGB masing-masing memiliki range antara 0 hingga 63 sehingga jumlah warna yang dapat kita pilih untuk mengisi warna pada sebuah cell di tabel ialah $63 \times 63 \times 63 = 16$ juta warna. Tetapi seluruh tabel hanya dapat diisi dengan 256 pilihan warna. Kita dapat mengubah intensitas warna dari sebuah warna pada tabel dengan cara menggunakan *interrupt-interrupt*.

Seringkali citra yang dihasilkan atau didapatkan memiliki kualitas yang kurang bagus. Hal ini dapat dikarenakan gangguan teknis peralatan yang digunakan atau adanya *noise* sekitarnya. Dapat juga dikarenakan kurang bagusnya pencahayaan. Untuk ini dilakukan manipulasi data-data untuk memperbaiki kualitas citra.

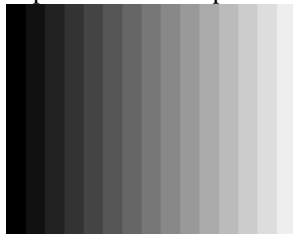
2.3 Color Filtering

Color Filtering adalah metode yang berguna untuk menemukan sebuah warna yang terdapat pada sebuah gambar. Dengan ditemukannya warna yang kita cari pada sebuah gambar, kita dapat menentukan proses apa yang selanjutnya harus dilakukan.

Pada dasarnya pencarian ini menggunakan kombinasi dari komponen *Red*, *Green*, dan *Blue* yang terdapat pada gambar. Nilai dari masing-masing komponen ini didapat dari hasil beberapa kali eksperimen. Hasil dari eksperimen ini adalah sebuah kombinasi dari komponen warna *Red*, *Green*, dan *Blue*. Kombinasi warna ini yang kemudian dijadikan *filter* yang merupakan penentu sebuah warna diloloskan atau tidak.

Keluaran dari metode ini langsung akan menghasilkan sebuah gambar biner. Gambar biner sendiri adalah sebuah gambar yang hanya memiliki 2 derajat keabuan yaitu hitam dan putih seperti pada Gambar 2.1. Dimana warna putih pada gambar adalah merupakan representasi dari sebuah pixel yang warnanya diloloskan. Sedangkan warna hitam pada gambar merupakan representasi dari pixel yang warnanya tidak diloloskan.

Sehingga dari proses diatas, kita dapat menyimpulkan bahwa warna putih pada gambar adalah objek yang sedang kita cari. Karena warna putih pada gambar merupakan warna pada pixel yang sesuai dengan filter yang kita dapat dari hasil eksperimen.



Gambar 2.1. Contoh Gambar Derajat Keabuan

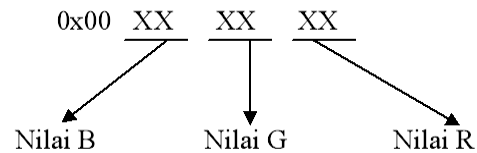
2.4 Sistem Warna RGB

Dalam pengolahan citra warna dipresentasikan dengan nilai hexadesimal dari 0x00000000 sampai 0x00ffffff. Warna hitam adalah 0x00000000 dan warna putih adalah 0x00ffffff. Definisi nilai warna diatas seperti Gambar 2.1 di bawah, variabel 0x00 menyatakan angka di belakangnya adalah hexadesimal.

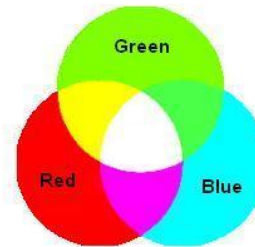
Terlihat pada Gambar 2.2 bahwa setiap warna memiliki range nilai 00 (angka desimalnya adalah 0) dan ff (angka desimalnya adalah 256), atau mempunyai nilai derajat keabuan $256 = 28$. Dengan demikian range warna

yang digunakan adalah $(28)(28)(28) = 224, 2$ (atau yang dikenal dengan istilah True Colour pada Windows). Nilai warna yang digunakan diatas merupakan gabungan warna cahaya merah, warna hijau dan warna biru seperti yang terlihat pada gambar dibawah. Sehingga untuk menentukan nilai dari suatu warna yang bukan warna dasar digunakan gabungan skala kecerahan dari setiap warnanya.

Dari definisi Gambar 2.3 di bawah untuk menyajikan warna tertentu dapat dengan mudah dilakukan, yaitu dengan mencampurkan ketiga warna dasar RGB. Berikut merupakan contoh-contoh kombinasi warna yang bisa digunakan dapat dilihat pada Tabel 2.1.



Gambar 2.2. Nilai warna RGB dalam Hexadesimal



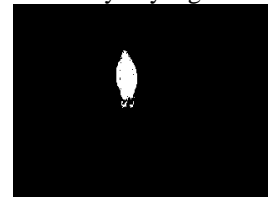
Gambar 2.3. Komposisi warna RGB

Tabel 2.1. Kombinasi warna dalam Hexsadesimal

Nilai	Warna	Nilai	Warna
0x00000000	Hitam	0x0000A000	Orange
0x000000FF	Merah	0x00888888	Abu-abu
0x0000FF00	Hijau	0x00FF00AA	Ungu
0x00FF0000	Biru	0x00AAFF00	Hijau Muda
0x0000FFFF	Kuning	0x00AA00FF	Merah Muda
0x00FF00FF	Magenta	0x00AAFFFF	Kuning Muda
0x00FFFF00	Cyan	0x000088AA	Coklat
0x00FFFFFF	Putih	0x00AA0088	Ungu

2.5 Segmentasi Warna

Segmentasi merupakan proses untuk memisahkan obyek yang kita ambil dengan latar belakang menggunakan teknik *Color Filtering* pada warna-warna yang mendekati warna obyek yang kita tentukan.



Gambar 2.4. Contoh segmentasi api pada gambar

2.6 Thresholding

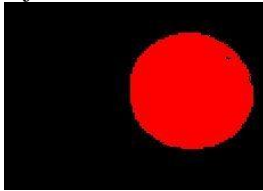
Thresholding digunakan untuk mengatur jumlah derajat keabuan yang ada pada citra. Dengan menggunakan thresholding maka derajat keabuan bisa diubah sesuai keinginan, misalkan diinginkan menggunakan derajat keabuan 16, maka tinggal membagi nilai derajat keabuan dengan 16. Proses thresholding ini pada dasarnya adalah proses perubahan kuantisasi pada citra, sehingga untuk melakukan thresholding dengan derajat keabuan dapat digunakan rumus:

$$x = w / b$$

dimana :

x adalah nilai derajat keabuan setelah thresholding
 w adalah nilai derajat keabuan sebelum thresholding
 b adalah jumlah derajat keabuan yang diinginkan

Berikut ini contoh *thresholding* untuk mengubah citra berwarna menjadi 2 warna.



Gambar 2.5. Hasil Thresholding warna merah

2.7 Erosi

Erosi merupakan proses penghapusan titik-titik batas objek menjadi bagian dari latar, berdasarkan structuring element yang digunakan. Pada operasi ini, ukuran obyek diperkecil dengan mengikis sekeliling objek.

Cara yang dapat dilakukan ada 2:

- Dengan mengubah semua titik batas menjadi titik latar
- Dengan mensek semua titik di sekeliling titik latar menjadi titik latar.

Algoritma untuk operasi Erosi:

Untuk semua titik dalam citra

Cek apakah tersebut titik latar

- Jika ya maka ubah semua tetangganya menjadi titik latar
- Jika tidak maka lanjutkan



(a) Citra asli (b) Citra erosi terhubung-4 (c) Citra erosi terhubung-8

Gambar 2.6. Proses Erosi

2.8 Dilasi

Dilasi merupakan proses penggabungan titik-titik latar menjadi bagian dari objek, berdasarkan structuring element yang digunakan. Proses ini adalah kebalikan dari

erosi, yaitu merubah latar disekeliling objek menjadi bagian dari objek tersebut.

Terdapat 2 cara untuk melakukan operasi ini, yaitu:

- Dengan cara mengubah semua titik latar yang bertetangga dengan titik batas menjadi titik obyek, atau lebih mudahnya set setiap titik yang tetangganya adalah titik obyek menjadi titik obyek.
- Dengan mengubah semua titik di sekeliling titik batas menjadi titik obyek, atau lebih mudahnya set semua titik tetangga sebuah titik obyek menjadi titik obyek.

Algoritma untuk operasi dilasi citra biner:

Untuk semua titik dalam citra

Cek apakah tersebut titik obyek

- Jika ya maka ubah semua tetangganya menjadi titik obyek
- Jika tidak maka lanjutkan



(a) Citra asli (b) Citra dilasi terhubung-4 (c) Citra dilasi terhubung-8

Gambar 2.7. Proses Dilasi

2.9 Canny Edge Detection

Metode Canny ini tidak memiliki operator khusus, namun metode ini terdiri dari beberapa langkah khusus. Metode Canny akan mendeteksi tepi dengan mencari nilai gradien maksimal lokal dari sebuah citra I . Gradien tersebut dihitung menggunakan turunan dari Gaussian filter.

Metode Canny menggunakan dua thresholds, yang berguna untuk mendeteksi tepian yang terlihat jelas, dan tepian yang kurang jelas atau lemah, termasuk juga tepian yang kurang jelas yang terlihat pada output yang terhubung dengan tepian yang jelas.

Metode ini lain dengan metode sebelumnya, yang sebelum dilakukan proses deteksi tepi harus dilakukan terlebih dahulu proses penghilangan noise. Metode Canny lebih utama akan mendeteksi tepian yang kurang jelas, yang tidak dapat diperoleh dengan menggunakan metode lain.

2.10 Computer Vision

Menurut dalam buku karya Rinaldi Munir yang berjudul *Pengolahan Citra Digital dengan Pendekatan Algoritmik*.

Terminologi lain yang berkaitan erat dengan pengolahan citra digital adalah computer vision atau machine computer. Pada hakikatnya, computer vision mencoba meniru cara kerja visual manusia (Human Vision). Human Vision sesungguhnya sangat kompleks, manusia melihat obyek dengan indera penglihatan (mata) lalu obyek citra diteruskan ke otak untuk diinterpretasi

sehingga manusia mengerti obyek apa yang tampak dalam pandangan matanya. Hasil interpretasi ini mungkin digunakan untuk mengambil suatu keputusan.

Computer vision merupakan proses otomatis yang mengintegrasikan sejumlah besar proses untuk persepsi visual, seperti akuisisi citra, pengolahan citra, klasifikasi, pengenalan (recognition), dan membuat keputusan. Computer vision terdiri dari teknik-teknik untuk mengestimasi ciri-ciri objek dalam citra, pengukuran ciri yang berkaitan dengan geometri objek, dan menginterpretasi informasi geometri tersebut. Secara ringkas bisa dilihat persamaan berikut :

$$Vision = Geometry + Measurement + Interpretation$$

Proses-proses dalam computer vision dibagi dalam 3 (tiga) aktifitas :

1. Memperoleh atau mengakuisisi citra digital.
2. Melakukan teknik komputasi untuk memproses atau memodifikasi data citra. (operasi-operasi pengolahan citra).
3. Menganalisis dan menginterpretasi citra menggunakan hasil pemrosesan untuk tujuan tertentu, misalnya memandu robot, mengontrol peralatan, memantau manufaktur dan lain-lain.

Pengolahan citra merupakan proses awal (preprocessing) pada computer vision, sedangkan pengenalan pola merupakan proses untuk menginterpretasi citra. Teknik-teknik di dalam pengenalan pola memainkan peranan penting dalam computer vision untuk mengenali objek.

2.11 Robotino

Robotino penuh dengan berbagai fungsi, sistem robot berkualitas tinggi dengan unit penggerak omnidirectional. Dengan 3 unit penggerak memberikan robotino gerakan ke segala arah – kedepan,kebelakang, dan kesamping – dan robot dapat berputar juga. Robotino juga dilengkapi dengan webcam dan beberapa tipe sensor, dari sensor analog sampai sensor jarak, sebagai contoh adalah binary untuk pelindung tabrakan dan digital untuk mengetahui kecepatan aktual. Robotino dapat digunakan langsung – tanpa PC.

Unit pengontrol pada robotino terdiri dari sebuah PC yang tertanam dengan sebuah compact flash card, dimana telah terinstall sistem operasi linux dan beberapa contoh aplikasi. Contoh aplikasi dapat langsung dijalankan dari Robotino control panel.

Robotino dapat diprogram dengan software Robotino View dari PC via wireless LAN. Robotino View bekerja mengirimkan sinyal kepada unit motor penggerak dan juga menampilkan, mengubah, dan mengevaluasi nilai sensor. Robotino bahkan dapat diprogram dengan Robotino View secara aktual/real-time.

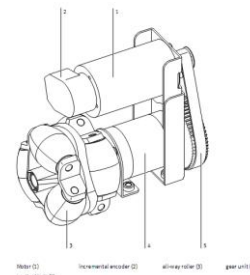
Linux dan API C++ juga dapat digunakan untuk memprogram Robotino.

Webcam juga dapat digunakan untuk menampilkan dan mengelola gambar kamera secara real-time dengan bantuan Robotino View. Dengan cara ini, aplikasi seperti jalur dan pengenalan objek dapat diimplementasikan.

Sensor tambahan dan aktuator dapat dikoneksikan dengan I/O interface.



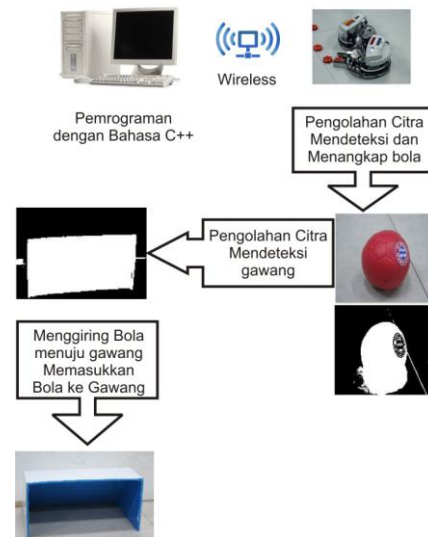
Gambar 2.8. Robotino tanpa gripper dan dengan gripper



Gambar 2.9. Roda Omnidirectional Robotino

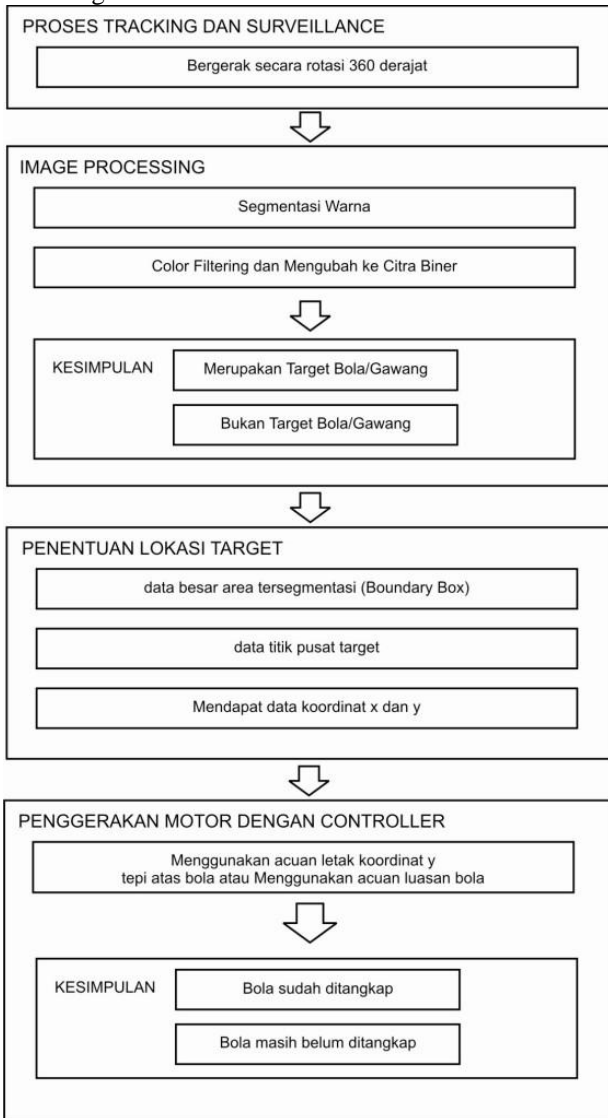
3. Perencanaan Sistem

Secara umum sistem yang dibangun pada proyek akhir ini dapat dilihat pada gambar – gambar diagram berikut :



Gambar 3.1. Blok Diagram Secara Umum

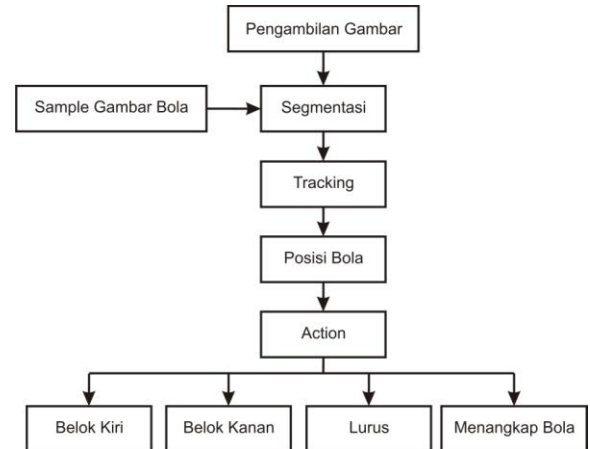
Secara garis besar sistem terbagi 2 proses, yaitu image processing dan Robot control.



Gambar 3.2. Blok Diagram sistem

3.1 Sistem Pencarian

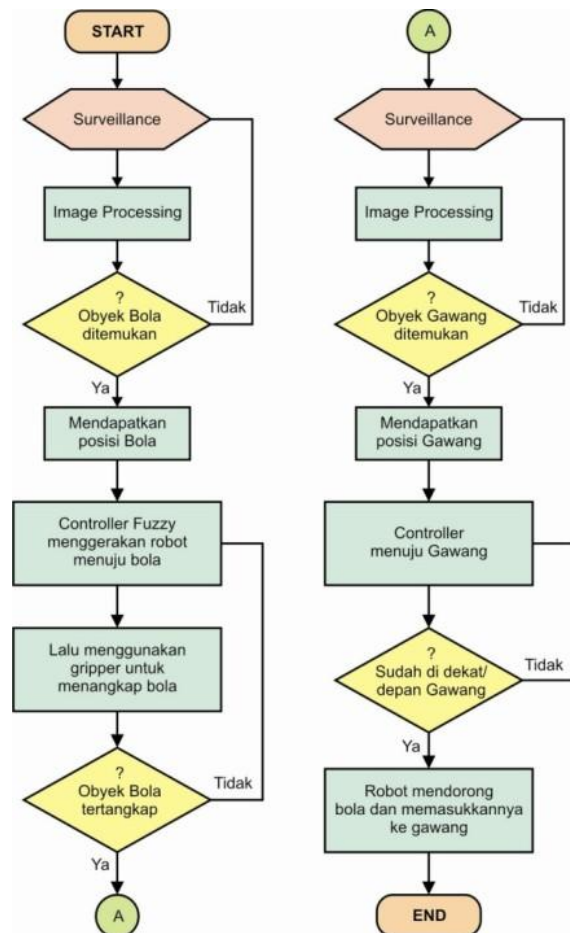
Dalam pembuatan program pencarian objek bola berwarna merah menggunakan blok diagram secara umum dapat dilihat pada gambar di bawah :



Gambar 3.3. Diagram Sistem Pencarian / Deteksi Bola

3.2 Sistem Robotino Tracking

Pada bagian ini merupakan desain flowchart dari sistem Robotino melakukan tracking dalam proyek ini yang dibuat oleh penulis. Flowchart dapat dilihat pada Gambar 3.9.



Gambar 3.4. Flowchart sistem Robotino tracking target bola dan gawang

4. Implementasi

Pengujian program dilakukan terhadap Robotino apakah Robotino dapat mendeteksi bola, menuju bola, mendeteksi gawang, menggiring bola, dan memasukkan bola ke dalam gawang. Adapun faktor yang menentukan keberhasilan Robotino mencetak gol adalah

- Faktor cahaya

Faktor cahaya berperan besar dalam hal *image processing*, tingkat keberhasilan segmentasi warna ditentukan oleh faktor ini, apabila cahaya kurang atau redup, kemungkinan Robotino tidak dapat mendeteksi bola maupun gawang sehingga proses terjadinya gol pun tidak terjadi dan dapat dinyatakan gagal.

- Faktor posisi Robotino

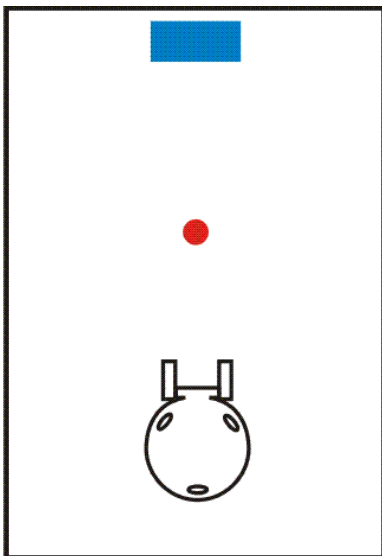
Faktor posisi turut menentukan apakah Robotino dapat menggiring bola menuju gawang dengan baik, biasanya posisi Robotino membelakangi mulut gawang lebih sulit menggiring bola daripada posisi Robotino menghadap mulut gawang.

Pengujian dalam tugas akhir ini juga meliputi pengujian fungsi segmentasi warna obyek, fungsi pelacakan (*tracking*), dan fungsi pergerakan motor Robotino untuk bergerak mendekati obyek.

Berikut merupakan Pengujian dan Analisa program terhadap faktor cahaya dan posisi Robotino. Format pengujian yang dilakukan sebagai berikut :

1. Apakah proses mendeteksi bola berhasil?
2. Apakah Robotino berhasil bergerak sesuai arah posisi bola?

Contoh pengkondisian Robotino di lapangan seperti gambar di bawah ini.



Gambar 3.5.

5. Hasil Dan Kesimpulan

Proses pengujian diatas dilakukan 3 macam pengujian :

1. 3 Pengujian I

Menghasilkan :

1. 3 Deteksi bola berwarna merah sukses
2. 2 Proses Boundary Box terbuat dengan baik, 1 kurang baik
3. 2 Dengan noise yang kecil, 1 dengan noise yang cukup banyak

2. 6 Pengujian II

Menghasilkan :

1. 6 Deteksi bola berwarna merah sukses
2. 4 Tracking bola sukses, 2 sedikit melenceng
3. 2 Pengujian III, mengenai hasil proses tracking

Menghasilkan :

1. Dari proses yang didapat, terlihat bahwa koordinat x kamera dari posisi koordinat yang positif jauh nilainya semakin kecil, artinya titik tengah kamera semakin mendekati titik tengah bola. Proses tracking berhasil.
2. Sebaliknya dari proses kedua ini didapat, terlihat bahwa koordinat x kamera dari posisi koordinat yang negatif jauh nilainya semakin besar, artinya titik tengah kamera semakin mendekati titik tengah bola. Proses tracking berhasil.

Dari 3 pengujian diatas dapat dianalisa :

- a. Apabila pendeteksian bola berwarna merah berhasil maka peluang keberhasilan Robotino melakukan tracking bola pun besar.
- b. Bentuk gripper berperan baik dalam menangkap bola.
- c. Kegagalan tracking bola berwarna merah dikarenakan kecepatan perputaran saat mencari bola terlalu cepat, tidak sesuai dengan jarak bola terhadap robot, sehingga Robotino bergerak tanpa arah. Hal ini dikarenakan pembuatan Boundary Box dan penentuan luasan bola kurang akurat.
- d. Segmentasi warna mempunyai tingkat keberhasilan yang tinggi apabila didukung dengan intensitas cahaya yang sesuai di lingkungannya
- e. Pada hasil percobaan / pengujian di atas, posisi titik tengah obyek sebenarnya dapat dikunci dengan baik, namun pada saat robot bergerak untuk mendekati bola maka intensitas cahaya dan sudut datang cahaya berubah-ubah sehingga banyak titik yang dituju hilang saat pelacakan dimulai. Hal itulah yang menyebabkan terjadinya kegagalan tersebut.
- f. Beberapa kondisi yang berbeda menyebabkan intensitas cahaya yang berbeda pula. Dan gerak robot menyebabkan berbeda pula sudut datang cahaya ke kamera pada Robotino . Dan juga jarak benda terhadap kamera sangat mempengaruhi dalam penguncian obyek.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan analisa dari beberapa pengujian yang diterangkan pada bab sebelumnya, kesimpulan yang didapatkan adalah

1. Jam-jam tertentu sangat berpengaruh terhadap keberhasilan dalam mendeteksi bola dan juga mempengaruhi keakuratan Robotino dalam melakukan *tracking* terhadap bola, karena perbedaan intensitas sangat berpengaruh dalam proses segmentasi warna bola
2. posisi awal Robotino mempunyai sedikit pengaruh dalam keberhasilan Robotino dalam mendeteksi bola dan melakukan proses *tracking*.

Dari 9 pengujian yang dilakukan, tingkat keberhasilan Robotino dalam melakukan *tracking* adalah 66,67 %.

5.2 Saran

Dalam proyek akhir ini tidak terlepas dari adanya kekurangan yang membutuhkan saran-saran untuk mendukung kesempurnaannya.

1. Fungsi program yang kurang sempurna sehingga mengakibatkan Robotino kadang-kadang error dalam menjalankan program.
2. bola dan gawang yang masih sederhana.
3. Lingkungan atau ruangan dengan intensitas cahaya yang mendukung.

Daftar Pustaka

- [1] Kaehler, Gary Bradski and Adrian. 2008. *Learning OpenCV - Computer Vision with the OpenCV Library*. O'Reilly Media, Inc. CA.
- [2] OpenRobotinoApiHowTo From : <http://doc.openrobotino.org/documentation/OpenRobotinoApiHowTo/OpenRobotinoApiHowTo.chm>
- [3] Forum Robotino, <http://forum.openrobotino.org>
- [4] Textbook TP 101. Pneumatics Basic Level. Festo Didactic From : http://www.festo-didactic.com/ov3/media/customers/1100/093131_les_eprobe_web.pdf
- [5] Robotino Manual. Festo Didactic. From : http://www.festo-didactic.com/ov3/media/customers/1100/544305_robotino_deen.pdf
- [6] [Ellis and Stroustrup, 1990] Ellis, M. and Stroustrup, B. (1990). *The Annotated C++ Reference Manual*. Addison-Wesley, Reading, MA.
- [7] Atmel Corporation. 2005. *Easy Image Processing: Camera Interfacing for Robotics*. From : http://www.atmel.com/dyn/resources/prod_documents/issue4_pg39_43_Robotics.pdf
- [8] Xailendra. 2004. *bahasa C++*. From : <http://erlangga.staff.gunadarma.ac.id/Downloads/files/9238/Teknik+Pemrograman+Terstruktur+2+%3D++C%2B%2B.pdf>

- [9] Kustiawan, Dedi. 2009. *Kendali Raket Pada Game Serangga Nakal Menggunakan Kamera*. Jurusan Teknologi Informasi Politeknik Elektronika Negeri Surabaya Institut Teknologi Sepuluh Nopember, Surabaya.

- [10] Image Segmentation. From :

<http://www.cs.toronto.edu/~jepson/csc2503/segmentation.pdf>

- [11] Nielsen, Simon Kracht and Carsten. *Robots in everyday human environments*. 2007. From :

<http://www.control.aau.dk/~tb/wiki/images/Thesis.pdf>

[CV Penulis]

Gentang Syabba Nahla menjalankan studi D4 bidang Teknik Informatika pada Politeknik Elektronika Negeri Surabaya – Institut Teknologi Sepuluh Nopember(PENS-ITS) semester 8.