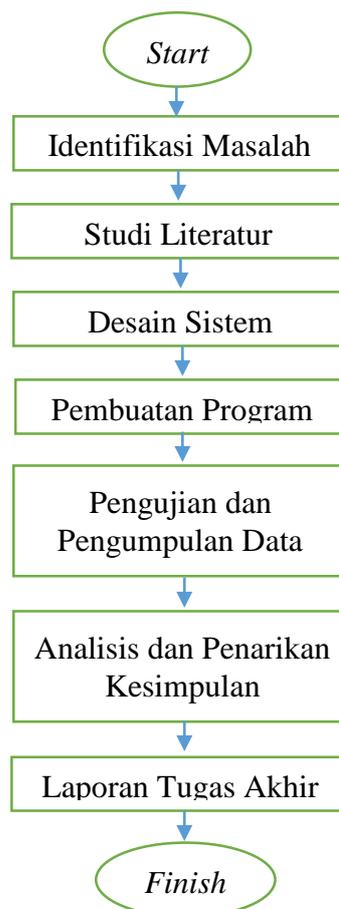


BAB III

METODOLOGI PENELITIAN

Proses penelitian Tugas Akhir ini dilaksanakan dalam beberapa tahapan kegiatan. Tahapan-tahapan tersebut saling terkait dan saling mendukung satu sama lain, serta dilaksanakan secara urut sesuai dengan yang telah direncanakan, yaitu dimulai dengan identifikasi masalah hingga diakhiri dengan penulisan laporan Tugas Akhir. Adanya metodologi penelitian ini bertujuan agar proses penelitian dapat berjalan dengan baik dan menghasilkan keluaran yang memuaskan serta untuk membantu pembaca dalam memahami jalannya alur dari penelitian ini. Adapun uraian dari tahapan-tahapan tersebut adalah sebagai berikut:



Gambar 3.1 Diagram Alir Tahapan Penelitian Tugas Akhir

3.1 Identifikasi Masalah

Mengidentifikasi masalah yang kemudian menjadi latar belakang dalam sebuah kegiatan penelitian merupakan tahap pertama yang harus dilakukan oleh penulis. Pengidentifikasian masalah sangat penting dilakukan sehingga didapatkan rumusan masalah dan selanjutnya terciptalah tujuan dari kegiatan penelitian tersebut. Dalam penelitian ini, permasalahan didapatkan penulis dari pengalaman mengikuti Kontes Robot Sepak Bola Indonesia Beroda. Permasalahan utama yang menjadi topik dalam penelitian ini adalah masih minimnya potensi dari *image processing* yang diaplikasikan pada robot sepak bola tim UMY (Mr_Dev), yaitu masih sebatas sistem penjejakan bola. Oleh karena itu, dalam penelitian ini akan dilakukan penggalian potensi dari *image processing* secara lebih mendalam untuk mendeteksi objek-objek penting lainnya dalam permainan sepak bola, yaitu bola, lapangan, robot lawan dan juga gawang lawan. Dengan demikian diharapkan mampu semakin meningkatkan performa Mr Dev dalam ajang perlombaan KRSBI-Beroda di tahun mendatang.

3.2 Studi Literatur

Studi literatur dilakukan guna memperdalam wawasan umum yang berkaitan dengan penelitian tugas akhir ini, serta digunakan sebagai dasar teori yang kemudian akan membantu dalam melakukan pembahasan dan analisis hasil pengujian. Studi Literatur diharapkan mampu menunjang tingkat keberhasilan suatu kegiatan penelitian dan membantu penulis dalam memecahkan permasalahan dalam penelitiannya. Adapun sumber literatur yang digunakan dalam penelitian ini berasal dari internet, jurnal, buku dan tugas akhir.

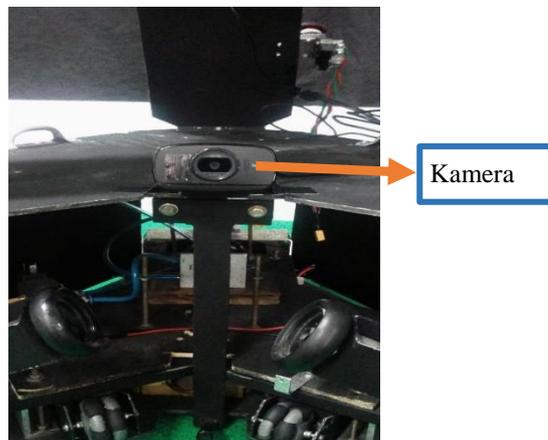
3.3 Desain Sistem

Sistem deteksi objek berbasis warna pada penelitian ini dibuat dengan menggunakan hardware berupa dua buah kamera webcam bertipe Logitech c525 sebagai pengindera, satu buah bola stainless berukuran 4 inch sebagai cermin pada kamera omnivision, satu buah laptop bertipe Lenovo Ideapad z410 sebagai pengolah atau pemroses data, dan sebuah arduino mega sebagai penerima data-data

output dari sistem. Kedua kamera webcam tersebut terhubung ke laptop melalui kabel USB. Kedua kamera tersebut memiliki peranan yang berbeda, yaitu sebagai berikut,

1. Kamera Depan

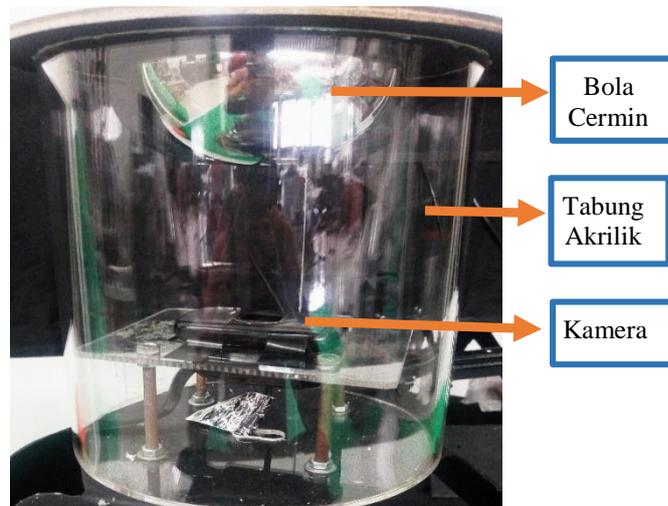
Kamera ini diletakkan tepat di atas penendang robot dengan ketinggian 30 cm dan menghadap ke arah depan. Selain itu, posisi dari kamera depan ini tidaklah tegak lurus, melainkan sedikit condong ke arah depan. Kamera depan ini akan berperan dalam mengindera objek-objek yang berada pada wilayah di depan robot, atau lebih khususnya untuk mendeteksi objek-objek yang berjarak relatif jauh dari robot.



Gambar 3.2 Posisi Kamera Depan pada Robot Sepak Bola Mr Dev

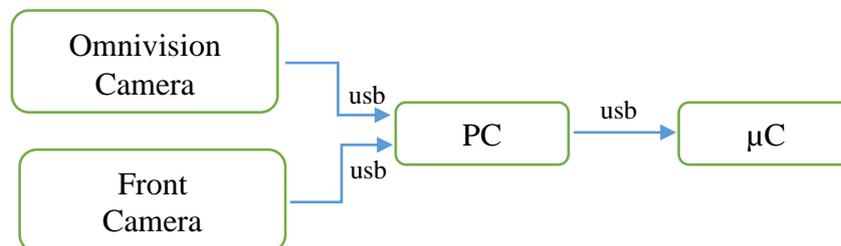
2. Kamera Omnivision

Kamera ini diletakkan tepat di titik tengah, di atas robot pada ketinggian 66 cm dan menghadap ke atas (ke arah bola cermin). Sedangkan jarak antara kamera dengan bola cermin adalah sebesar 9 cm. Sistem kamera omnivision ini memiliki peran sebagai pengindera area-area yang menjadi titik buta dari kamera depan. Titik buta dari kamera depan antara lain adalah bagian sisi kanan, kiri dan belakang robot. Hal tersebut dimungkinkan karena dengan bantuan dari bola cermin, jangkauan penginderaan kamera omnivision menjadi sebesar 360° , atau dengan kata lain dapat menjangkau keseluruhan wilayah disekitar robot.



Gambar 3.3 Posisi Kamera Omnivision pada Robot Sepak Bola Mr Dev

Kedua kamera tersebut akan saling melengkapi satu sama lain, yaitu dimana kamera omnivision memiliki jangkauan lebar penginderaan 360° tetapi jangkauan jaraknya kurang, sedangkan kamera depan sebaliknya, memiliki jangkauan jarak yang lebih jauh, tetapi lebar penginderaannya kurang. Hasil penginderaan kedua kamera tersebut kemudian akan dikirimkan ke PC (laptop) untuk kemudian dilakukan pengolahan citra. Keluaran dari pengolahan citra tersebut adalah berupa data-data koordinat dan sudut dari objek-objek yang telah terdeteksi. Selanjutnya, data-data tersebut akan dikirimkan menuju ke *microcontroller* melalui komunikasi serial. Komunikasi tersebut dilakukan dengan menggunakan media berupa kabel USB *type A to type B*, dimana tipe A akan terhubung dengan PC sedangkan tipe B akan terkoneksi dengan *microcontroller*.



Gambar 3.4 Block Diagram Sistem *Image Processing* pada Robot Sepak Bola Mr Dev

3.4 Pembuatan Program

Pada penelitian tugas akhir ini, perancangan dan pembuatan program dilakukan dengan menggunakan bahasa pemrograman python dan library OpenCV 3.4.0 dalam IDE (*Integrated Development Environment*) Python 2.7. Berikut adalah daftar *library* yang digunakan dalam program sistem deteksi objek berbasis warna yang diimplementasikan pada robot sepak bola beroda tim Mr Dev,

1. OpenCV

Library OpenCV berisikan fungsi-fungsi untuk melakukan semua pekerjaan yang berkaitan dengan pengolahan citra, seperti *thresholding*, *opening*, *closing*, dan lain sebagainya.

2. Numpy

Library Numpy bertugas untuk melakukan semua pekerjaan operasi numerik dalam sistem *image processing* ini.

3. Imutils

Library Imutils bertugas untuk melakukan pengolahan dasar citra, seperti *translation*, *rotation*, *resizing*, *skeletonization*, menyortir kontur, mendeteksi *edges* dan lain sebagainya.

4. Serial

Library Serial menyediakan fungsi-fungsi komunikasi serial, sehingga sistem *image processing* dapat mengirimkan data-data tertentu melewati komunikasi serial ke *microcontroller*.

5. Math

Library Math menyediakan fungsi – fungsi matematika yang didefinisikan dalam standar bahasa pemrograman C. Fungsi – fungsi yang disediakan antara lain seperti Trigonometri, perpangkatan, logaritma, konversi sudut, dan lain sebagainya.

3.4.1 Algoritma Pendeteksian Objek Berbasis Warna

Pada penelitian ini, semua deteksi objek dilakukan dengan berdasarkan warna, dimana semua prosesnya dimulai dari pengambilan citra RGB dan diakhiri dengan mengirim data koordinat objek-objek tersebut ke *microcontroller*. Secara

garis besar, berikut adalah beberapa proses yang akan dilalui dalam pendeteksian objek.

1. Konversi Citra RGB Menjadi HSV dan *Grayscale*

Ruang warna RGB sangat tidak direkomendasikan untuk digunakan di dalam pengolahan citra berbasis warna. Hal tersebut dikarenakan pada ruang warna RGB nilai *luma* dan *chroma* bergabung menjadi satu, sehingga hasil pengolahan citra berbasis warna menjadi tidak optimal. *Luma* itu sendiri merupakan tingkat kecerahan dari suatu warna, sedangkan *chroma* adalah tingkat intensitas atau kekuatan dari warna. Pengolahan citra berbasis warna akan lebih optimal ketika dapat mengakses informasi *luma* dan *chroma* secara terpisah pada suatu warna. Oleh karena itulah, maka perlu dilakukan proses konversi dari RGB ke ruang warna lain yang memisahkan antara nilai *luma* dan *chroma*, seperti misalnya HSV. Dalam penelitian ini, konversi RGB menjadi HSV digunakan dalam proses pendeteksian bola, lapangan, dan tanda warna lawan. Sedangkan dalam proses pendeteksian gawang, RGB dikonversikan menjadi citra *Grayscale*.

2. *Masking* atau *Thresholding* warna

Proses *Masking* warna merupakan proses yang dilakukan untuk menyeleksi warna yang ingin dideteksi. Setiap warna memiliki nilai *range* nya masing-masing, sehingga sebelum dilakukan proses *masking* warna, maka perlu melalui tahap *tunning* nilai parameter batas atas dan bawah terlebih dahulu. Dalam penelitian ini, karena digunakan ruang warna HSV, maka nilai parameter batas atas dan bawah yang harus *ditunning* terlebih dahulu adalah *Hue*, *Saturation*, dan *Value*. Sedangkan pada citra *Grayscale*, nilai parameter yang harus *ditunning* hanya satu, yaitu *range* tingkat intensitas warna yang akan dideteksi, dimana bernilai antara 0 (hitam) hingga 255 (putih).

3. *Opening* Warna

Dalam penelitian ini, tahap proses *opening* warna hanya dilakukan pada objek-objek yang berukuran besar saja, seperti misalnya lapangan, deteksi bola pada kamera depan, dan gawang. Sedangkan deteksi bola pada

kamera omnivision dan tanda warna lawan tidak melalui tahap ini. Semua proses *opening* yang dilakukan pada penelitian ini menggunakan nilai kernel yang sama, yaitu matriks berukuran 5 x 5. Ukuran kernel yang digunakan di dalam proses *opening* terbilang kecil, hal tersebut dikarenakan apabila menggunakan nilai kernel yang terlalu besar, maka akan sangat beresiko dapat ikut mengeliminasi warna objek yang ingin dideteksi. Oleh karena itulah, proses ini menggunakan kernel bernilai kecil dan hanya dilakukan pada objek-objek berukuran besar saja. Sedangkan untuk nilai iterasi erosi-dilasi berbeda-beda pada setiap deteksi objek, bergantung pada kebutuhan masing-masing deteksi objek agar hasil pendeteksian menjadi maksimal.

Tujuan dari dilaluinya proses ini adalah untuk mengurangi tingkat *noise* berupa objek-objek lain yang memiliki warna serupa dengan objek yang ingin dideteksi. Hal tersebut dapat dilakukan karena objek-objek *noise* tersebut berada di luar lapangan, sehingga jaraknya akan lebih jauh daripada objek yang akan dideteksi. Hal tersebut mengakibatkan luas wilayah warna objek-objek *noise* tersebut menjadi lebih kecil daripada objek yang akan dideteksi, sehingga dengan demikian dimungkinkan untuk dapat mengeliminasi *noise* tersebut dengan menggunakan teknik transformasi *opening*.

4. *Closing* Warna

Berbeda dari tahap sebelumnya, tahap *closing* dilakukan pada semua proses deteksi objek. Hal tersebut dikarenakan tahap ini tidak memiliki resiko seperti halnya pada proses *opening*. Tidak seperti tahap *opening* yang akan mengeliminasi warna-warna yang memiliki luas wilayah berukuran kecil, tahap *closing* memiliki tugas untuk menutupi atau menambal suatu warna yang di dalamnya terdapat warna-warna lain. Dalam penelitian ini, nilai kernel dan nilai iterasi dilasi-erosi yang digunakan berbeda-beda pada setiap deteksi objek, bergantung pada kebutuhan masing-masing deteksi objek agar hasil pendeteksian menjadi maksimal.

5. Filter

Tahap ini dilakukan pada semua objek yang akan dideteksi, dimana dalam penelitian tugas akhir ini nilai kernel yang digunakan bernilai sama, yaitu matriks berukuran 16×16 . Jenis filter yang digunakan dalam penelitian ini adalah filter 2D *Convolution*, filter ini akan membuat warna objek terdeteksi menjadi tersamarkan atau *blur*, sehingga dapat membantu mengurangi tingkat *noise* dan memperluas wilayah warna dari objek yang ingin dideteksi.

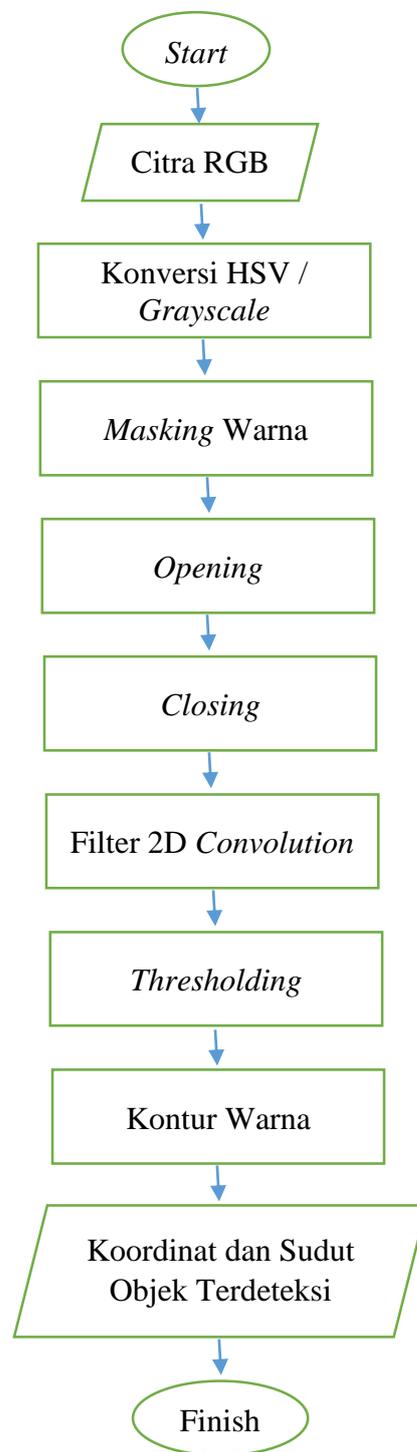
6. Thresholding

Tahap ini juga dilakukan pada semua objek yang akan dideteksi. Tahap *thresholding* akan mengembalikan citra blur menjadi jelas atau tajam kembali. Pada tahap *thresholding* ini akan terlihat efek dari tahap filter 2D *Convolution*, yaitu luas wilayah warna objek terdeteksi menjadi lebih luas, serta tingkat *noise* menjadi berkurang.

7. Kontur

Dalam Penelitian ini, tahap yang paling wajib dilalui pada semua proses deteksi objek adalah mencari kontur dari warna objek tersebut. Hal tersebut dikarenakan untuk mendapatkan data *center of gravity* atau titik pusat massa dari suatu objek, maka kontur warna dari objek tersebut harus didapatkan terlebih dahulu. *Center of gravity* dari warna objek dapat didapatkan dengan menggunakan metode analisis *moment*. Hasil dari analisis *moment* tersebutlah yang kemudian akan menjadi cikal bakal dari posisi koordinat objek terdeteksi.

Selain dimanfaatkan untuk mendapatkan data koordinat objek terdeteksi, pada penelitian ini kontur juga dimanfaatkan untuk mendapatkan titik-titik ekstrim pada bola yang kemudian dikombinasikan dengan data kontur lapangan untuk digunakan dalam algoritma anti-*noise* warna serupa bola. Selain itu, kontur juga dimanfaatkan untuk mendeteksi sisi wilayah gawang terluas (tidak terhalangi oleh robot penjaga gawang lawan) atau dengan kata lain kontur warna gawang terbesar yang mana kemudian digunakan untuk menentukan lokasi titik tembak pada gawang lawan.



Gambar 3.5 Diagram Alir Proses Deteksi Objek Berbasis Warna pada Robot Sepak Bola Mr Dev

3.4.2 Algoritma Koordinat pada Objek Terdeteksi

Pada sistem *image processing* Robot Sepak Bola Mr Dev, *output* atau keluaran dari sistem tersebut adalah berupa data-data koordinat dan sudut dari objek terhadap kamera. Data-data koordinat tersebut didapatkan dengan jalan mencari titik tengah atau *center of gravity* dari objek yang telah terdeteksi. Setelah didapatkan koordinat titik tengah objek terdeteksi, langkah selanjutnya adalah dengan membagi frame menjadi dua bagian secara horizontal dan juga secara vertikal. Jadi dengan demikian frame akan terbagi menjadi 4 bagian sama besar, yaitu kanan atas, kiri atas, kanan bawah dan kiri bawah. Secara matematis, algoritma untuk mendapatkan titik koordinat objek terdeteksi dapat dituliskan sebagai berikut,

$$\text{Koordinat Objek } x = \text{Koordinat Titik Tengah } x - \left(\frac{1}{2} \times \text{Lebar Frame}\right) \quad (3.1)$$

$$\text{Koordinat Objek } y = \left(\frac{1}{2} \times \text{Tinggi Frame}\right) - \text{Koordinat Titik Tengah } y \quad (3.2)$$

Dalam Penelitian ini, ukuran *frame* yang digunakan adalah 320 x 240p, dengan menggunakan formula di atas, maka ketika objek terdeteksi tepat berada di tengah *frame*, maka nilai koordinat *x* nya adalah 0, sedangkan apabila berada di sisi kanan *frame*, maka nilai koordinat *x* nya adalah mulai dari 1 hingga 159. Sebaliknya, jika objek terdeteksi berada pada sisi kiri *frame*, maka nilai koordinat *x* nya adalah mulai dari -1 hingga -159. Sama seperti halnya pada koordinat sumbu *x*, ketika objek terdeteksi tepat berada di tengah *frame*, maka nilai koordinat *y* nya adalah 0, sedangkan jika berada di sisi atas *frame*, maka nilai koordinat *y* objek tersebut adalah mulai dari 1 hingga 119. Sebaliknya, jika objek terdeteksi berada pada sisi bawah *frame*, maka nilai koordinat *y* objek tersebut adalah mulai dari -1 hingga -119.

3.4.3 Algoritma Sudut Objek Terdeteksi terhadap Kamera

Data-data nilai sudut objek terdeteksi terhadap kamera di dapatkan dengan memanfaatkan metode trigonometri pada segitiga siku-siku. Algoritma nilai sudut objek terdeteksi pada kamera depan dan kamera omnivision memiliki sedikit

perbedaan, hal ini mengingat jangkauan lebar kamera omnivision adalah 360° , sedangkan kamera depan hanya beberapa derajat saja.

3.4.3.1 Kamera Depan

Berikut adalah langkah-langkah atau proses yang dilakukan untuk mendapatkan nilai sudut objek terdeteksi terhadap kamera depan.

1. Membagi *frame* menjadi dua bagian secara vertikal, sehingga didapatkan *frame* terbagai menjadi dua zona atau bagian sama besar, yaitu kanan dan kiri.
2. Menggambar 3 buah garis untuk membentuk sebuah segitiga siku-siku.
3. Menghitung panjang garis yang dibutuhkan untuk menghitung sudut $\tan \theta$, yaitu panjang garis di depan dan garis di samping dari sudut yang dicari ($\tan \theta$). Secara matematis, panjang kedua garis tersebut dapat ditemukan dengan formula,

$$\text{Garis Samping} = \text{Tinggi } Frame - \text{Koordinat Objek } y \quad (3.3)$$

$$\text{Garis Depan} = \text{Koordinat Objek } x - \left(\frac{1}{2} \times \text{Lebar } Frame\right) \quad (3.4)$$

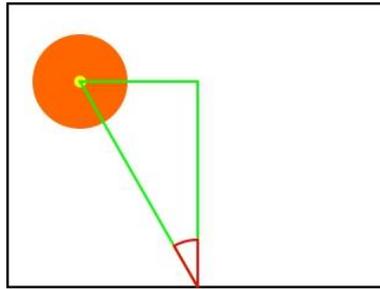
4. Menghitung radian $\tan \theta$

$$\text{rad } \tan \theta = \frac{\text{Garis Depan}}{\text{Garis Samping}} \quad (2.3)$$

5. Mengkonversi radian $\tan \theta$ ke dalam bentuk derajat

$$\text{Derajat Objek} = \text{rad } \tan \theta \times \frac{180}{\pi} \quad (2.4)$$

Algoritma ini akan menghasilkan sudut objek terdeteksi dimana ketika objek berada tepat didepan kamera (di tengah *frame*), nilai sudutnya adalah 0° , sedangkan jika berada disisi kanan kamera (sisi kanan *frame*), maka nilai sudutnya akan bernilai positif, yaitu lebih dari ($>$) 0° hingga batas jangkauan sisi kanan kamera. Sebaliknya, jika objek terdeteksi berada disisi kiri kamera (sisi kiri *frame*), maka nilai sudutnya akan bernilai negatif, yaitu kurang dari ($<$) 0° hingga batas lebar kiri kamera.



Gambar 3.6 Metode Trigonometri Segitiga Siku-Siku pada Kamera Depan

3.4.3.2 Kamera Omnivision

Berikut adalah langkah-langkah atau proses yang dilakukan untuk mendapatkan nilai sudut objek terdeteksi terhadap kamera omnivision.

1. Membagi *frame* menjadi empat bagian, yaitu secara vertikal dan juga horizontal, sehingga didapatkan *frame* terbagi menjadi empat zona atau bagian sama besar, yaitu kanan atas, kiri atas, kanan bawah dan kiri bawah.
2. Menggambar 3 buah garis untuk membentuk sebuah segitiga siku-siku.
3. Menghitung panjang garis yang dibutuhkan untuk menghitung sudut $\tan \theta$, yaitu panjang garis di depan dan garis di samping dari sudut yang dicari ($\tan \theta$).

$$\text{Garis Samping} = \text{Koordinat Objek } y - \left(\frac{1}{2} \times \text{Tinggi Frame}\right) \quad (3.5)$$

$$\text{Garis Depan} = \text{Koordinat Objek } x - \left(\frac{1}{2} \times \text{Lebar Frame}\right) \quad (3.4)$$

6. Menghitung radian $\tan \theta$

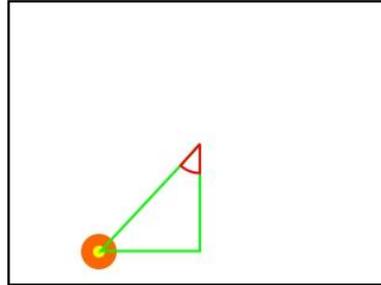
$$\text{rad } \tan \theta = \frac{\text{Garis Depan}}{\text{Garis Samping}} \quad (2.3)$$

7. Mengkonversi radian $\tan \theta$ ke dalam bentuk derajat

$$\text{Derajat Objek} = \text{rad } \tan \theta \times \frac{180}{\pi} \quad (2.4)$$

Algoritma ini akan menghasilkan sudut objek terdeteksi dimana jika objek terletak tepat di depan robot, nilai sudutnya adalah 0° , sedangkan jika objek berada di sisi kanan robot, baik itu kanan atas maupun kanan bawah, maka nilai sudutnya akan bernilai positif, yaitu mulai lebih dari ($>$) 0° hingga 180° . Sebaliknya jika

objek terdeteksi berada disisi kiri robot, baik itu kiri atas maupun kiri bawah, maka nilai sudutnya akan bernilai negatif, yaitu mulai kurang dari ($<$) 0° hingga -180° .



Gambar 3.7 Metode Trigonometri Segitiga Siku-Siku pada Kamera Omnivision

3.4.4 Algoritma Sistem Anti-Noise Warna Bola

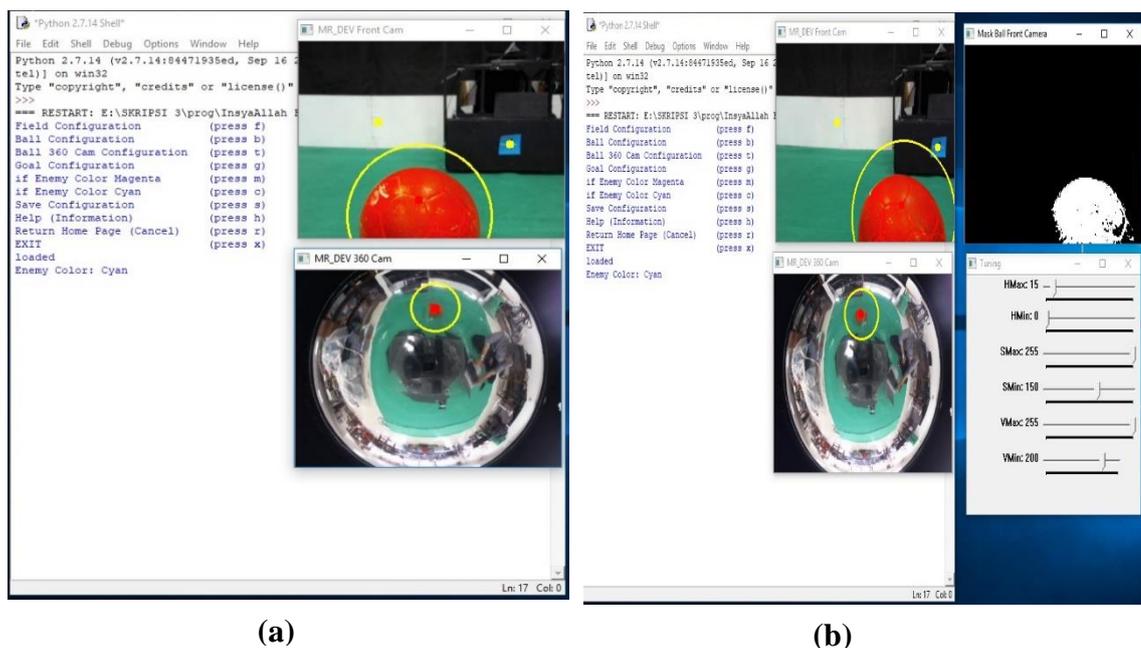
Noise yang dimaksud dalam penelitian ini adalah warna-warna serupa warna bola yang berada di luar arena sepak bola. *Noise* warna bola tersebut akan sangat mengganggu pergerakan dari robot, dimana ketika terdapat warna serupa bola yang berada di luar lapangan, maka akan ada kemungkinan robot justru akan bergerak ke arah warna *noise* tersebut.

Algoritma yang digunakan dalam sistem anti-*noise* pada warna bola adalah dengan cara memanfaatkan data kontur dari lapangan dan juga data posisi titik-titik ekstrim pada bola. Titik-titik ekstrim yang dimaksud adalah titik paling atas, paling bawah, paling kanan dan paling kiri pada bola. Setelah kedua data tersebut didapatkan, maka selanjutnya adalah mengeksekusi kedua data tersebut dengan menggunakan fitur *point polygon test* yang terdapat di dalam *library* OpenCV. Fitur tersebut memungkinkan untuk dapat mengetahui jarak terdekat dari suatu titik menuju ke kontur suatu objek terdeteksi.

Hasil dari perintah tersebut adalah ketika titik tersebut berada di dalam kontur objek terdeteksi, maka jaraknya akan bernilai positif, sedangkan jika tepat berada pada garis kontur, maka akan bernilai 0. Sebaliknya, jika titik tersebut berada di luar kontur objek terdeteksi, maka jaraknya akan bernilai negatif.

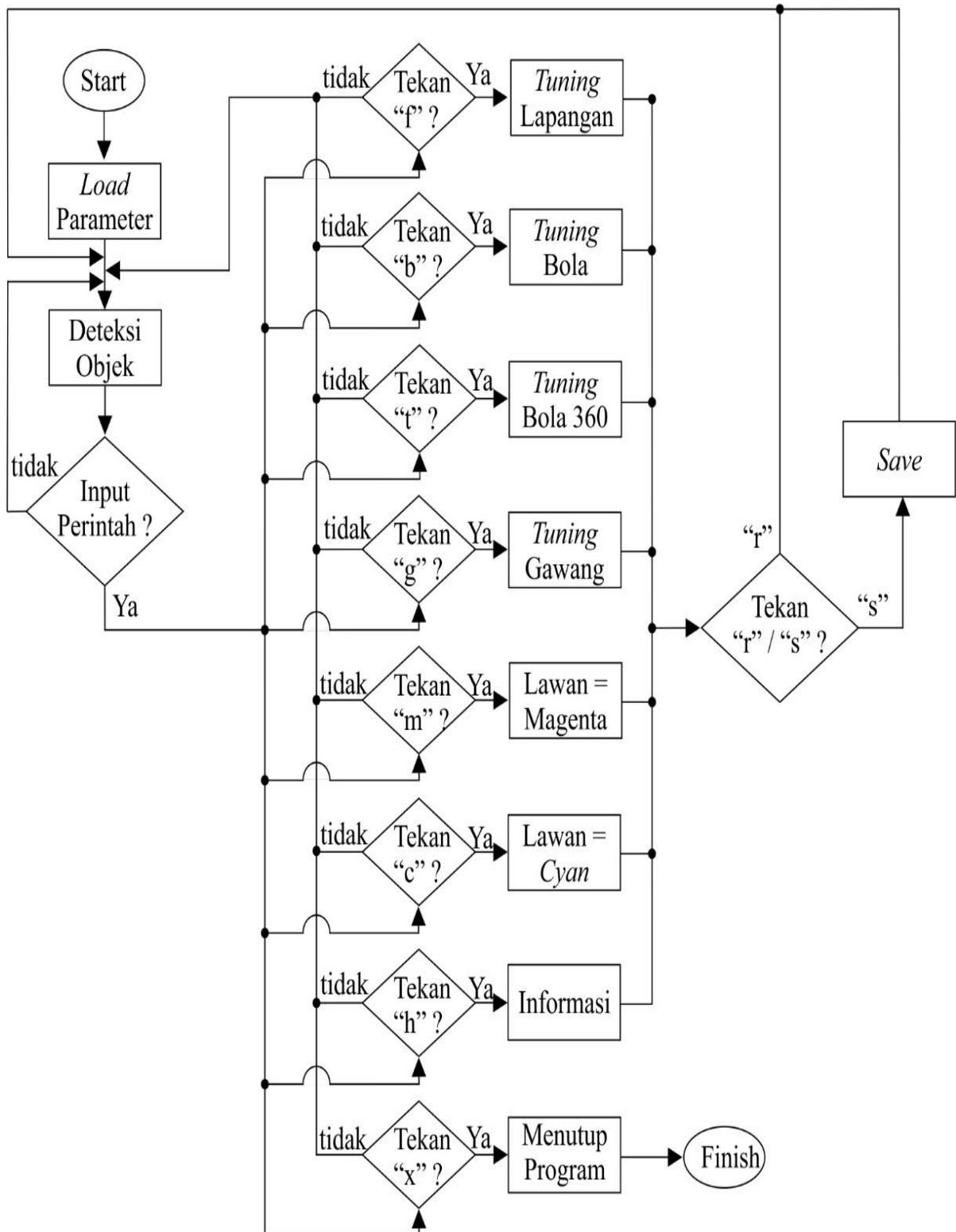
3.4.5 Tampilan Sistem Utama

Pada sektor tampilan program, dibagi menjadi tampilan utama dan tampilan sub menu. Pada menu utama program akan menampilkan daftar submenu atau perintah-perintah untuk menjalankan program dan 2 buah *frame* hasil dari pengindareraan kamera depan dan omnivision yang telah melewati proses pengolahan citra. Sedangkan pada tampilan sub menu, berisikan dengan fitur *tuning range* warna HSV yang akan digunakan dalam proses *masking* warna. Fitur *tuning range* warna HSV ini tersedia untuk bola (kamera depan), bola (kamera omnivision), lapangan, dan gawang. Sehingga dengan demikian tidak perlu lagi untuk mengubah nilai parameter batas atas dan bawah HSV melalui *source code*. Program ini juga dilengkapi dengan fitur *save* dan *load*, sehingga parameter-parameter hasil *tuning* tidak akan hilang meskipun program telah ditutup dan akan dimuat kembali saat program diaktifkan.



Gambar 3.8 (a) Tampilan Utama Program, **(b)** Contoh Tampilan *Tuning Range* Warna HSV Bola pada Kamera Depan

Pemberian perintah-perintah untuk menjalankan sistem *image processing* dilakukan dengan cara menekan huruf tertentu pada keyboard. Daftar perintah dicantumkan pada tampilan utama program, sehingga akan memudahkan dalam pengoperasiannya. Berikut adalah alur jalannya sistem *image processing*,



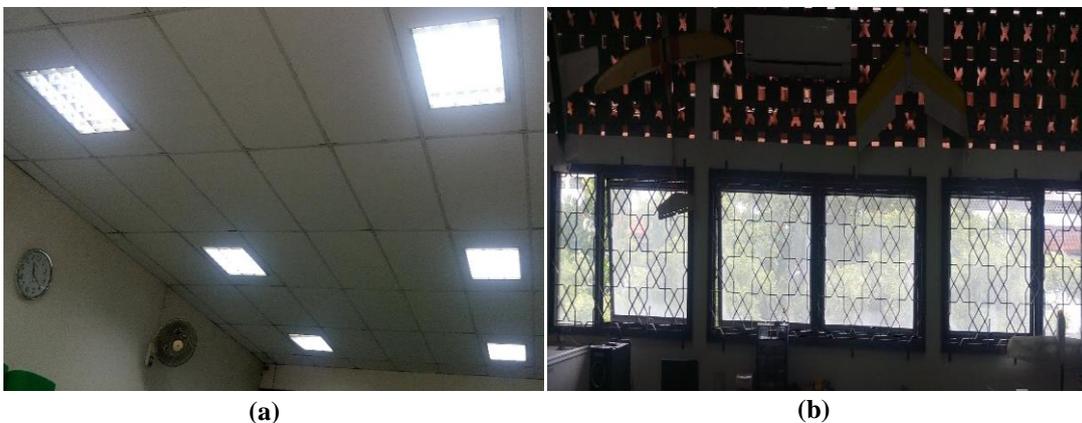
Gambar 3.9 Diagram Alir Program

3.5 Pengujian Sistem

3.5.1 Lokasi Pengujian

Pada penelitian ini, lokasi pengujian dan pengambilan data diusahakan memiliki kondisi yang semirip mungkin dengan pada saat kontes, seperti warna-warna objek di dalam arena pertandingan (bola, lapangan, gawang dan tanda warna lawan), pencahayaan, ukuran lapangan, dan lain sebagainya. Berdasarkan faktor – faktor tersebut, maka pengujian tidak dapat dilakukan pada ruangan terbuka atau *outdoor*, dimana sumber pencahayaannya lebih didominasi oleh cahaya matahari. Sumber pencahayaan dari sinar matahari memiliki beberapa kelemahan, yaitu seperti tidak konsisten atau mudah berubah-ubah intensitasnya, dan sangat cepat perubahan sudut pencahayaannya. Oleh karena itulah, maka pengujian dan pengambilan data dilakukan di ruangan tertutup, dimana sumber utama pencahayaannya adalah berasal dari cahaya lampu, sehingga intensitas pencahayaannya lebih stabil dan sudut pencahayaan yang tetap.

Lokasi yang dianggap paling ideal untuk melakukan pengujian dan pengambilan data adalah di ruangan MRC (*Microcontroller & Robotic Club*). Pada ruangan MRC, sumber pencahayaan utama yang digunakan untuk pengujian dan pengambilan data adalah lampu TL (*Tubular Lamp*) atau neon, serta dibantu penerangan dari cahaya matahari yang masuk melalui jendela. Rata-rata tingkat pencahayaan pada lokasi pengujian adalah 203,6 lux (di atas robot) dan 185,7 lux (di area sekitar robot)



Gambar 3.10 Sumber Pencahayaan (a) Primer dan (b) Sekunder pada Lokasi Pengujian

Lokasi gawang lawan ditempatkan menghadap ke arah jendela, hal tersebut dikarenakan apabila membelakangi jendela, maka akan menyebabkan kondisi silau atau *back light* pada kamera, yang mana tentu saja akan sangat mengganggu sistem *image processing*. Selain itu, dikarenakan keterbatasan tempat pada ruangan MRC, maka ukuran karpet yang digunakan sebagai lapangan adalah berukuran lebar 2 meter dan panjang 5 meter. Sedangkan ukuran gawang tetap sesuai dengan ukuran yang berlaku pada regulasi KRSBI-Beroda, yaitu dengan lebar 2 meter, tinggi 1 meter dan kedalaman 0,5 meter. Selain itu, perlindungan terhadap gangguan dari *noise* benda-benda disekitar arena pengujian sangat penting untuk dilakukan. Sebagai contoh, dikarenakan warna cat dinding menyerupai warna gawang, maka sebagai bentuk perlindungan diberikan kain berwarna hitam untuk melingkupi dinding dibelakang gawang. Selain itu, pada sisi samping lapangan dilindungi dengan papan berwarna hitam dengan tinggi 30 cm, hal tersebut diperlukan karena disekitar lokasi pengujian dan pengambilan data terdapat banyak objek yang memiliki warna hampir serupa dengan warna objek-objek di dalam arena pengujian. Jadi dengan demikian, gangguan dari *noise* benda-benda di sekitar arena dapat diminimalisir.



Gambar 3.11 Kondisi Gawang dan Lapangan untuk Pengujian dan Pengambilan Data

3.5.2 Perlakuan Pengujian Sistem

Terdapat beberapa macam pengujian sistem yang akan dilakukan, yaitu antara lain,

1. Pengujian Fitur *Tuning* Warna

Pengujian fitur *tuning* warna HSV objek (bola, lapangan, gawang) bertujuan untuk mengetahui apakah fitur tersebut telah dapat berjalan dengan baik ataukah tidak. Selain itu, pengujian ini juga bertujuan untuk mendapatkan nilai parameter batas atas dan bawah nilai warna HSV objek yang akan dideteksi.

Prosedur pengujian dilakukan dengan cara menggeser-geser posisi *scroll bar*, kemudian diamati apakah pada tampilan *masking* warna dan nilai parameternya berubah sesuai dengan perubahan posisi *scroll bar* ataukah tidak. Pada pengujian ini, penggeseran posisi *scroll bar* terus dilakukan hingga tampilan *masking* warna hanya menunjukkan warna HSV dari objek yang akan dideteksi.

2. Pengujian Perintah *Save* dan *Load*

Tujuan dari dilakukannya pengujian fitur *save* dan *load* adalah untuk mengetahui apakah perintah tersebut telah dapat berjalan dengan baik ataukah tidak. Fitur *save* bertugas untuk menyimpan data-data parameter hasil dari proses *tuning* warna dan beberapa data lainnya. Sedangkan fitur *load* bertugas untuk memanggil kembali data-data parameter yang telah disimpan sebelumnya.

Pada prosedur pengujian perintah *save*, dilakukan dengan cara menjadikan pada kondisi awal semua parameter bernilai nol (0), kemudian melakukan *tuning* atau penyetelan pada setiap warna objek yang akan dideteksi. Setelah proses *tuning* suatu warna objek selesai, lalu perintah *save* dipanggil. Terakhir adalah melihat pada file "*Storage Threshold.txt*" apakah nilai parameter yang telah di*setting* berhasil tersimpan di dalam file tersebut ataukah tidak. Sedangkan prosedur pengujian perintah *load*, dilakukan dengan cara menutup sistem *image processing* setelah menjalankan perintah *save*. Selanjutnya menjalankan kembali sistem dan mengamati apakah nilai parameter yang telah disimpan dapat terpanggil kembali ke dalam sistem ataukah tidak.

3. Pengujian Penggantian Tanda Warna Lawan

Pengujian penggantian tanda warna lawan bertujuan untuk mengetahui kinerja dari fitur tersebut, apakah telah dapat berjalan sesuai dengan fungsinya ataukah tidak. Dalam Kontes Robot Sepak Bola Indonesia Beroda, setiap tim robot akan ditandai dengan tanda warna dan dalam setiap pertandingan selalu berganti tanda warnanya yaitu antara *cyan* atau *magenta*. Oleh sebab itulah maka fitur penggantian tanda warna lawan harus dapat berjalan dengan baik.

Pengujian dilakukan dengan cara mengganti tanda warna lawan menjadi *magenta*, dan kemudian mengamati *file* “*Storage Threshold.txt*” pada urutan data ke 20 apakah berubah menjadi bernilai 2 ataukah tidak. Selanjutnya mengganti kembali tanda warna lawan menjadi *cyan* dan mengamati apakah data ke 20 pada *file* “*Storage Threshold.txt*” berubah menjadi bernilai 1 ataukah tidak.

4. Pengujian Perintah *Help*, *Return*, *Exit*

Pengujian perintah *Help*, *Return* dan *Exit* bertujuan untuk mengetahui apakah perintah-perintah tersebut dapat berjalan sebagai mana dengan fungsinya ataukah tidak. Perintah *Help* berfungsi untuk memberikan informasi mengenai semua fitur atau perintah yang terdapat di dalam sistem, perintah *Return* berfungsi untuk mengembalikan ke tampilan awal atau *homepage*. Sedangkan perintah *Exit* bertugas untuk menonaktifkan sistem secara keseluruhan.

Prosedur pengujian dilakukan dengan cara menjalankan perintah-perintah tersebut dan kemudian mengamati respon dari sistem. Hasil dari pengujian dikatakan berhasil ketika respon sistem ketika menjalankan perintah-perintah tersebut sesuai dengan fungsi atau tugasnya.

5. Pengujian Sudut pada Kamera Omnivision

Pengujian sudut pada kamera omnivision bertujuan untuk melihat tingkat akurasi dari hasil pembacaan sudut kamera omnivision pada sistem *image processing* Mr Dev.

Prosedur pengujian dilakukan dengan cara membandingkan hasil pembacaan sudut objek terdeteksi pada sistem dan pada busur derajat.

Selanjutnya menghitung selisih hasil pembacaan antara keduanya serta tingkat *error*nya. Dalam pengujian ini, kamera yang digunakan adalah kamera omnivision, sedangkan objek terdeteksi yang digunakan adalah berupa bola.

6. Pengujian Sudut pada Kamera Depan

Pengujian sudut pada kamera depan bertujuan untuk mengetahui tingkat akurasi hasil pembacaan sudut kamera depan pada sistem *image processing* Mr Dev. Selain itu, tujuan dari pengujian ini juga untuk mengetahui luas jangkauan deteksi objek pada kamera depan.

Seperti halnya pengujian sudut pada kamera omnivision, prosedur pengujian sudut pada kamera depan juga dilakukan dengan cara membandingkan hasil pembacaan sudut objek terdeteksi pada sistem dan pada busur derajat. Kemudian menghitung selisih hasil pembacaan keduanya serta tingkat *error*nya.

Pengujian selanjutnya menguji sudut terjauh objek dapat terdeteksi pada kamera depan. Pengujian sudut deteksi objek terjauh tersebut dilakukan dalam berbagai jarak untuk melihat tingkat konsistensi dari pendeteksian kamera depan. Seluruh pengujian sudut pada kamera depan dilakukan dengan menggunakan bola sebagai media objek yang akan dideteksi.

7. Pengujian Deteksi Lapangan

Tujuan dari pengujian deteksi lapangan adalah untuk mengetahui hasil dari setiap tahapan proses yang dilalui dalam pendeteksian lapangan. Sehingga dengan demikian akan terlihat pada tahapan apa sajakah proses deteksi kurang dapat berjalan dengan maksimal.

Pada penelitian ini, secara spesifik yang dimaksud dengan deteksi lapangan adalah mendeteksi tepi atau batas arena lapangan sepak bola. Prosedur pengujian dilakukan dengan cara mengamati hasil pada setiap tahapannya, yaitu mulai dari tahap konversi citra RGB hingga kontur warna lapangan sepak bola.

8. Pengujian Deteksi Bola pada Kamera Depan

Tujuan dari pengujian deteksi bola pada kamera depan adalah untuk mengetahui hasil dari setiap tahapan proses yang dilalui dalam pendeteksian bola kamera depan. Sehingga dengan demikian akan terlihat pada tahapan apa sajakah proses deteksi kurang dapat berjalan dengan maksimal. Selain itu, pengujian ini juga bertujuan untuk mengetahui jarak minimum dan maksimum objek dapat terdeteksi pada kamera depan.

Sama seperti halnya pada prosedur pengujian deteksi lapangan, pengujian deteksi bola juga dilakukan dengan cara mengamati hasil pada setiap tahapan prosesnya. Selanjutnya mengamati data hasil keluaran deteksi bola yaitu berupa koordinat dan sudut bola apakah sudah sesuai dengan algoritma yang ditetapkan atautah tidak.

Pengujian selanjutnya yang akan dilakukan adalah mengukur jarak minimum dan maksimum bola dapat terdeteksi oleh kamera depan. Prosedur pengujian dilakukan dengan cara menempatkan bola menempel di depan robot, kemudian bola mulai dijauhkan hingga sistem tidak mampu lagi untuk mendeteksinya. Prosedur tersebut diulang pada beberapa posisi sudut yang berbeda untuk melihat tingkat konsistensi deteksi bola pada kamera depan. Pada pengujian ini, alat ukur yang digunakan adalah berupa meteran dan busur derajat.

9. Pengujian Deteksi Bola pada Kamera Omnivision

Pengujian deteksi bola pada kamera omnivision bertujuan untuk mengetahui hasil dari setiap tahapan proses yang dilalui dalam pendeteksian bola kamera omnivision. Sehingga dengan demikian akan terlihat pada tahapan apa sajakah proses deteksi kurang dapat berjalan dengan maksimal. Selain itu, pengujian ini juga bertujuan untuk mengetahui jarak minimum dan maksimum objek bola dapat terdeteksi oleh kamera omnivision.

Prosedur pengujian deteksi bola pada kamera omnivision dilakukan dengan cara mengamati hasil citra pada setiap prosesnya, yaitu mulai dari *tunning* warna HSV bola pada kamera omnivision, hingga tahap kontur bola. Selanjutnya mengamati apakah data koordinat dan sudut bola pada

kamera omnivision telah sesuai dengan algoritma yang digunakan ataukah tidak.

Pengujian selanjutnya adalah menguji jarak minimum dan maksimum pendeteksian bola pada kamera omnivision. Pengujian tersebut dilakukan pada setiap 45° hingga mengitari robot, sehingga akan terdapat 8 titik lokasi pengujian. Prosedur pengujian dilakukan dengan cara menempatkan bola menempel pada *body* robot selanjutnya menjauhkan bola hingga sistem tidak mampu lagi mendeteksi bola. Perangkat pengukuran yang digunakan adalah berupa meteran.

10. Pengujian Sistem Anti *Noise* Bola

Tujuan dari pengujian ini adalah untuk mengetahui apakah sistem anti *noise* bola yang telah disematkan di dalam sistem *image processing* dapat berjalan sesuai dengan fungsinya ataukah tidak.

Pengujian ini dilakukan dengan cara menempatkan bola di dalam lapangan sepak bola dan menempel pada papan pembatas lapangan berwarna hitam, kemudian bola akan diangkat secara perlahan hingga ketinggian tertentu hingga bola tidak lagi dapat terdeteksi oleh sistem. Prosedur pengujian tersebut dipilih dikarenakan pada saat kontes KRSBI-Beroda diselenggarakan, batas lapangan atau arena pertandingan adalah berupa papan hitam yang diletakkan melingkupi arena pertandingan, sehingga pengujian anti *noise* serupa warna bola yang paling cocok adalah dengan cara menggunakan metode tersebut. *Noise* yang dimaksud dalam pengujian ini adalah warna serupa warna bola yang terletak di luar lapangan sepak bola.

11. Pengujian Deteksi Tanda Warna Lawan

Tujuan dari pengujian deteksi tanda warna lawan adalah untuk mengetahui hasil pengolahan citra pada setiap tahapan proses yang dilalui dalam pendeteksian tanda warna lawan. Jadi dengan demikian akan terlihat pada tahap manakah proses deteksi kurang berjalan dengan maksimal. Selain itu, pengujian ini juga dilakukan untuk mengetahui seberapa jauh kamera depan dapat mendeteksi tanda warna lawan.

Pengujian ini dilakukan dengan cara mengamati hasil pada setiap tahapan prosesnya, kemudian mengamati apakah data keluarannya yang berupa koordinat dan sudut tanda warna lawan telah sesuai dengan algoritma yang diterapkan ataukah tidak.

Pengujian selanjutnya adalah untuk mengetahui jarak minimum dan maksimum deteksi tanda warna lawan. Prosedur pengujian dilakukan dengan cara menempatkan tanda warna lawan sedekat mungkin dengan kamera depan dan kemudian menjauhkannya secara perlahan hingga sistem tidak mampu lagi untuk mendeteksinya. Perangkat pengukuran yang digunakan dalam pengujian ini adalah berupa meteran.

12. Pengujian Deteksi Gawang lawan

Pada penelitian ini, secara spesifik deteksi gawang lawan yang dimaksud adalah untuk mendapatkan lokasi titik tembak pada gawang lawan. Pengujian deteksi gawang lawan bertujuan untuk mengetahui hasil pengolahan citra pada setiap tahapan proses yang dilalui dalam pendeteksian tanda warna lawan. Jadi dengan demikian akan terlihat pada tahap manakah proses deteksi kurang berjalan dengan maksimal. Selain itu juga untuk mengetahui apakah sistem telah berhasil menentukan posisi titik tembak pada gawang lawan ataukah belum.

Seperti halnya pada pengujian deteksi objek lainnya, pada pengujian ini juga dilakukan dengan cara mengamati hasil citra pada setiap tahapan prosesnya. Kemudian mengamati apakah data koordinat dan sudut titik tembak pada gawang lawan telah sesuai dengan algoritma yang diterapkan ataukah tidak.

13. Pengujian Nilai FPS

Pengujian nilai FPS (*Frame Per Second*) bertujuan untuk mengetahui rata-rata nilai FPS pada saat sistem menjalankan proses pengolahan citra. Dimana semakin tinggi nilai FPS yang didapat maka akan semakin baik.

Metode pengujian yang dilakukan adalah dengan cara menjalankan sistem hingga menghasilkan 120 *frame* citra. Selanjutnya menghitung

lamanya waktu yang diperlukan oleh sistem untuk mencapai 120 *frame* tersebut. Langkah terakhir adalah menghitung nilai FPS dengan formula sebagai berikut,

$$FPS = \frac{\text{Jumlah Frame}}{\text{Waktu Operasi}} \quad (3.6)$$

14. Pengujian Pengiriman Data

Pengujian pengiriman data bertujuan untuk mengetahui apakah sistem berhasil mengirimkan data-data penting yang dibutuhkan oleh *microcontroller* ataukah tidak.

Metode yang digunakan pada pengujian pengiriman data adalah dengan cara mengirimkan data dari sistem *image processing* pada PC menuju ke *microcontroller* dengan menggunakan kabel USB *type c*. Selanjutnya untuk melakukan pengamatan, maka data tersebut dikirimkan kembali dari *microcontroller* menuju ke sistem *image processing* pada PC. Kemudian pada sistem *image processing*, data tersebut akan ditampilkan untuk mengetahui apakah terjadi *corrupt* data pada proses pengiriman ataukah tidak. Urutan data yang akan dikirimkan dalam pengujian ini adalah dimulai dari koordinat x , y dan sudut bola pada kamera depan, kemudian koordinat x , y dan sudut bola pada kamera omnivision, selanjutnya koordinat x , y dan sudut titik tembak ke gawang lawan, serta terakhir adalah koordinat x , y , dan sudut tanda warna lawan, sehingga dengan demikian total terdapat 12 data.