

## Perancangan Robot Dengan Kemampuan Mencari, Mendekati, Dan Menggiring Bola Ke Gawang

Budi Santosa Haryanto<sup>1</sup>, Hadian Satria Utama<sup>1</sup> dan Dali S. Naga<sup>2</sup>

**ABSTRACT:** This design aims to create a robot that can search the ball, approach the ball, and keep the ball from the field/own goalpost towards the pitch area/the opponent's goalpost also pay attention to the position of the obstacles and the robot's position in the field. There are various movements such as looking for the ball, dribbling, and avoid the opponent. To be able to perform that movements the robot must be able to know what is the definition of the ball, the opponent, and goalpost. To know these objects then there must be a sensor that can read and that sensor is camera. The robot can know the object based on the different colors of each object caught by the camera. The capture of the camera then processed into a processor and the processor will determine the movement on the robot by sending current to the circuit driving the robot. Processor filters out the colors in turn at each color used. Processor then looks for the coordinates of the midpoint, and the area of the object detected. All values of the coordinates of the midpoint and the area of each color are used to determine the movement of the robot. Robot driving uses 3 driver motors DC, motors DC, and Omni wheels. The movements of the robot are going forward, rotating left/right, and standing still. The test results of the designed the robot successfully searched for the ball, approached the ball, kept the ball from the field/own goalpost towards the pitch area/the opponent's goalpost, and was prevented from going off the field.

**Keywords:** robot, wheels, football, motion, camera, processor.

**ABSTRAK:** Perancangan ini bertujuan untuk membuat robot yang dapat mencari bola, mendekati bola, dan menjauhkan bola dari daerah lapangan/gawang sendiri ke arah daerah lapangan/gawang lawan dengan memperhatikan juga posisi halangan dan posisi robot di lapangan. Terdapat berbagai macam gerakan seperti mencari bola, menggiring bola, dan menghindari lawan. Untuk dapat melakukan gerakan-gerakan tersebut, robot harus dapat mengetahui apa yang dimaksud dengan bola, lawan, dan gawang. Untuk mengetahui benda-benda tersebut maka harus terdapat sensor yang dapat membacanya yaitu kamera. Robot dapat mengetahui benda yang dimaksud berdasarkan warna-warna yang berbeda dari setiap benda hasil tangkapan kamera. Hasil tangkapan kamera kemudian diproses ke sebuah pemroses lalu pemroses menentukan penggerak pada robot dengan mengirim arus pada rangkaian penggerak robot. Pemroses memfilter warna yang ingin dideteksi saja secara bergilir pada setiap warna yang digunakan. Pemroses kemudian mencari koordinat titik tengah, dan luas dari benda yang dideteksi. Nilai-nilai koordinat titik tengah dan luas dari tiap warna yang digunakan ini lah yang menentukan pergerakan robot. Penggerak robot menggunakan 3 buah driver motor DC, motor DC, dan roda omni. Pergerakan robot yang dapat dilakukan adalah maju, berputar tempat kanan/kiri, berputar mengelilingi bola kanan/kiri, dan diam. Hasil pengujian robot yang dirancang berhasil mencari bola, mendekati bola, menjauhkan bola dari daerah lapangan/gawang sendiri ke arah daerah lapangan/gawang lawan, dan menghindari luar lapangan.

**Kata kunci:** robot, roda, sepak bola, gerakan, kamera, pemroses.

### PENDAHULUAN

Perkembangan robotika sangat pesat di negara-negara maju. Di Indonesia sendiri robotika cukup berkembang. Salah satu cara Indonesia dalam mengembangkan robotika adalah dengan mengadakan berbagai macam kontes robot. Salah satunya kontes yang di adakan di Indonesia adalah kontes robot sepak bola Indonesia. Kontes robot sepak bola di Indonesia sudah ada sejak tahun 2009 namun belum menggunakan pengolahan citra. Pada tahun 2012 barulah robot dipertandingkan seperti pertandingan sepak bola sesungguhnya dengan menggunakan aturan-aturan bermain seperti orang bermain sepak bola pada umumnya dengan menggunakan kamera untuk penglihatannya [1].

Dalam mendesain dan memrogram robot sepak bola tidaklah mudah, banyak yang harus dipelajari seperti kemampuan mendeteksi dan pergerakan robot. Berdasarkan masalah penentuan pergerakan robot yang harus dilakukan pada kondisi-kondisi tertentu, penulis merancang robot beroda yang menggunakan kamera untuk menentukan pergerakan robot dalam mencari, menggiring, menghindari suatu benda.

### DESKRIPSI KONSEP

Robot yang dirancang berfungsi untuk mencari, menggiring bola, melewati halangan, dan mengarahkan bola ke gawang lawan. Terlebih dahulu harus dilakukan kalibrasi warna bola, robot, penghalang, gawang lawan dan gawang sendiri, dan lapangan. Setelah melakukan kalibrasi warna robot, bola, gawang dan penghalang harus diletakkan di dalam lapangan setelah itu robot baru diaktifkan. Pengaktifan robot dilakukan dengan cara menekan *switch* pada robot. Setelah aktif maka pemroses pada robot *booting*. Setelah *booting* selesai pemroses langsung menjalankan program secara otomatis sehingga robot memulai gerakannya sesuai rancangan.

Pergerakan robot ditentukan oleh hasil pemrosesan dari pemroses. Saat pemroses sudah aktif robot menangkap gambar dari kamera. Hasil tangkapan dari kamera diterima oleh pemroses berupa sinyal. Pemroses kemudian mengubah sinyal-sinyal dari kamera menjadi kumpulan nilai-nilai. Nilai-nilai ini menyatakan warna yang didapat oleh kamera setiap *pixel*. Kumpulan nilai-nilai *pixel* ini kemudian digabungkan membentuk suatu *frame* matriks yang berisi semua nilai-nilai dalam *pixel*. Pemroses lalu mengambil suatu rentang nilai tertentu saja sesuai warna yang ingin ditangkap pada *frame* tersebut.

<sup>1</sup> Program Studi Teknik Elektro, Fakultas Teknik, Universitas Tarumanagara

<sup>2</sup> Program Studi Teknik Informastika, Fakultas Sistem Informasi, Universitas Tarumanagara

Pemroses lalu mendapat informasi-informasi berdasarkan warna yang ditangkap seperti posisi koordinat benda yang ingin dideteksi dan luas warna yang ditangkap terhadap suatu *frame*. Informasi-informasi inilah yang menentukan sinyal yang dikirim ke motor servo dan *driver* motor untuk menggerakkannya. Konstruksi keseluruhan robot dapat dilihat pada Gambar 1.



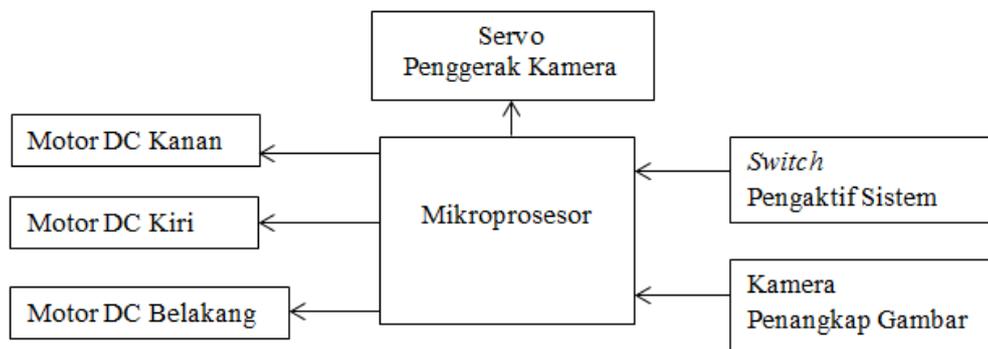
■ Gambar 1. Konstruksi Keseluruhan Robot

Robot pertama diaktifkan dengan *switch*. Setelah robot aktif, robot memulai pencarian bola secara otomatis. Jika robot tidak menemukan bola walaupun robot sudah mencari bola dengan berputar 360° robot mengidentifikasi tempat yang sedang robot tempati dan berpindah tempat dengan jarak tertentu. Ketika bola ditemukan maka robot mendeteksi bola sampai bola tepat di depan robot. Selanjutnya robot mengidentifikasi arah yang sedang dituju terhadap bola, apakah ke daerah sendiri atau daerah lawan dengan berputar di tempat secara perlahan dengan mencari warna gawang lawan. Jika robot menghadap ke arah daerah sendiri maka robot memutar bola hingga menuju ke arah lawan, jika sudah menuju arah lawan maka robot mendorong bola menuju gawang lawan.

Jika di depan robot terdapat halangan maka robot menggiring bola ke samping halangan yang lebih terbuka atau tidak terhalang. Selain itu robot juga dapat mengidentifikasi daerah luar lapangan. Robot berputar kanan hingga robot tidak menghadap ke daerah luar lapangan. Robot terus bergerak hingga bola berada di dekat gawang lawan.

#### DIAGRAM BLOK RANCANGAN

Berdasarkan deskripsi konsep robot yang dirancang memiliki dua macam *input*. *Input* pertama adalah pengaktif untuk mengaktifkan atau mematikan sistem. *Input* kedua adalah kamera untuk menangkap gambar. Gambar yang diterima dari kamera diolah oleh pemroses untuk menentukan pergerakan motor DC kanan, motor DC kiri, dan motor DC belakang. Diagram blok robot dapat dilihat pada Gambar 2.



■ Gambar 2. Diagram Blok Robot

#### Catu Daya

Catu daya adalah suatu perangkat elektronik yang berfungsi untuk mensuplai energi listrik ke suatu rangkaian elektronik lain. Salah satu catu daya yang sering ditemui adalah baterai. Berikut beberapa jenis-jenis baterai: *Zinc-carbon*, *Li-ion* (lithium-ion), dan *Li-Po*(Lithium Polymer).

#### Kamera

Kamera adalah suatu alat yang berfungsi untuk menangkap suatu gambar yang kemudian dapat menghasilkan data dari gambar tersebut sehingga dapat disimpan ke dalam memori. Sebuah kamera memiliki komponen utama lensa dan *image* sensor. *Image* sensor ini berfungsi untuk membaca cahaya yang dipantulkan benda dengan sangat cepat yang kemudian dapat menghasilkan suatu data yang dapat dibaca oleh perangkat lain seperti komputer untuk diproses.

### **Driver motor DC**

*Driver* motor DC adalah suatu alat yang digunakan untuk mengontrol menyalakan, mematikan, mengontrol arah putaran (searah jarum jam atau berlawanan arah jarum jam) dan kecepatan dari suatu motor DC. *Driver* motor DC biasa digunakan untuk pemrograman yang menggunakan pemroses yang hanya dapat menghasilkan *output* dengan arus yang kecil. *Driver* motor DC dapat berguna untuk mengubah arus yang kecil tersebut dan mengubahnya menjadi arus yang lebih besar sehingga dapat memutar motor.

### **Motor Servo**

Motor Servo adalah suatu motor yang dapat bergerak searah jarum jam atau berlawanan arah jarum jam yang memungkinkan servo dapat berhenti pada suatu sudut tertentu. Servo ada dua macam yaitu servo *continuous* yang dapat terus berputar 360° seperti motor dan servo *discontinuous* yang tidak dapat berputar 360°/memiliki batasan putar tertentu. Motor servo dikendalikan dengan memberi sinyal *pulse wide modulation* (PWM) yang sesuai dengan lebar pulsa dan *pulse cycle* servo.

### **Motor DC**

Motor DC adalah suatu motor yang menggunakan tegangan DC sebagai sumber penggerakannya. Motor DC terdiri dari dua bagian utama yaitu stator dan rotor. Stator adalah bagian yang tidak berputar yang terdapat magnet di sisinya. Rotor atau kumparan jangkar adalah bagian yang berputar. Ketika rotor terdapat tegangan maka rotor berputar karena timbul gaya gerak listrik

### **Roda Omni**

Roda omni adalah sebuah roda yang di sekelilingnya terdapat beberapa roda lagi berbentuk tabung oval yang dipasang sedemikian rupa sehingga memungkinkan suatu alat bergerak ke segala arah. Roda pada bagian pinggir roda ini terbuat dari bahan - bahan yang berbeda. Bahan yang sering digunakan untuk roda bagian pinggir roda ini adalah plastik dan karet. Gambar contoh roda omni dapat dilihat pada gambar 3.



■ **Gambar 3.** Roda Omni

### **Penurun Tegangan**

Penurun tegangan digunakan untuk mengubah tegangan yang lebih tinggi ketegangan yang lebih rendah. Penurunan tegangan ini biasanya digunakan agar satu sumber catu daya dapat digunakan untuk berbagai rangkaian elektronik yang membutuhkan tegangan atau arus yang berbeda-beda.

### **Mikroprosesor**

Mikroprosesor adalah prosesor komputer yang menggabungkan fungsi pusat prosesor unit komputer pada sebuah sirkuit terpadu atau beberapa sirkuit terpadu. Mikroprosesor adalah komponen serba guna, perangkat yang dapat diprogram yang menerima data digital sebagai *input*, memproses data digital sesuai instruksi pada memori, dan memberikan hasil sebagai *output*.

### **Open Source Computer Vision (OpenCV)**

OpenCV adalah sebuah *library* bebas yang dapat digunakan pada bahasa pemrograman C++, C, Python, dan Java yang mensupport sistem operasi windows, linux, mac OS, iOS, dan Android yang difokuskan untuk pengolahan citra. OpenCV berisi algoritma-algoritma yang memudahkan untuk mengenali isi suatu gambar [2, 3].

### **Raspberry Pi Tipe B**

Raspberry Pi Tipe B adalah sebuah *single board computer* [4] yang pada perancangan ini digunakan untuk mengontrol sistem secara keseluruhan mulai dari mengontrol kamera, servo, dan motor.

### **Kamera Webcam Logitech C170**

Kamera ini digunakan untuk menangkap gambar yang kemudian diproses oleh pemroses. Kamera ini memiliki ketajaman Gambar 1,3 megapixel dan memiliki kabel USB untuk transfer data

### Servo FUTABA S3003

Servo Futaba S3003 pada rancangan digunakan untuk menggerakkan kamera pada robot. Servo ini bekerja pada tegangan 4,8-6V, torsi pada tegangan 4,8V adalah 3,2 kg.cm dan 4,1 kg.cm pada tegangan 6V, mengkonsumsi arus sebesar 7,2 mA-8 mA, memiliki lebar pulsa 1000-2000us, menerima sinyal setiap 20ms.

### Baterai LiPo 5600 mAH 2 Cell 50-100C

Baterai LiPo 5600 mAH 2 cell digunakan untuk memberi daya ke robot agar dapat bekerja. Spesifikasi baterai Lipo yang digunakan adalah dua cell yang menyatakan baterai dapat mensuplai tegangan sebesar  $2 \times 3,7V$  ( $1 \text{ cell} = 3,7V$ ) = 7,4V spesifikasi 5600mAH menyatakan kapasitas baterai per jam, sedangkan 50C adalah faktor pengali kapasitas baterai per jam untuk mendapatkan total kapasitas baterai dapat digunakan.

### Motor DC 6V 75:1 133 rpm 8,8kg/cm

Motor DC 6V 75:1 133 rpm digunakan untuk menggerakkan robot ke suatu arah tertentu. Jumlah Motor DC yang digunakan adalah tiga buah. Motor DC yang digunakan pada robot yang dirancang adalah motor DC 6V 75:1 133 rpm. Torsi maksimal motor DC ini adalah 8,8 kg-cm. Motor DC ini dipilih karena robot yang dirancang tidak terlalu membutuhkan kecepatan tetapi lebih membutuhkan torsi untuk mengangkat beban robot yang cukup berat.

### Regulator LM7905

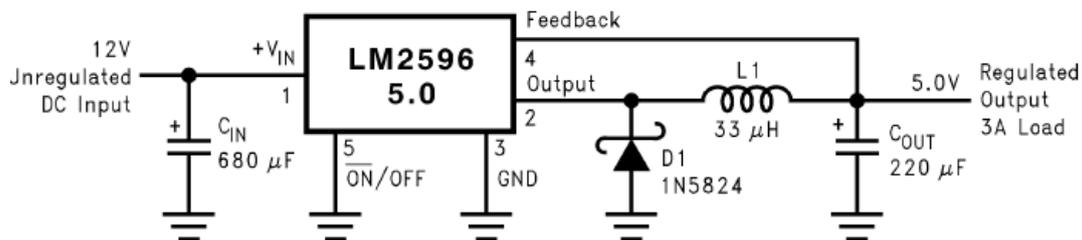
Regulator LM7905 ini digunakan untuk memberi tegangan ke motor *driver* L298. Regulator ini dapat mengubah tegangan hingga 25V dan menghasilkan arus hingga 1,5 ampere. Arus dan tegangan ini cukup untuk memberi daya pada IC motor *driver* DC L298 yang hanya membutuhkan tegangan *input* 4,5 volt- 7volt dengan arus maksimal 70mA.

### Ultimate Battery Eliminator Circuit Turnigy 15A 5-6V (UBEC)

UBEC berguna untuk menurunkan tegangan DC yang besar ke tegangan DC yang lebih kecil. UBEC biasanya terbatas untuk menurunkan tegangan pada tegangan 4 cell dan hanya mengeluarkan tegangan antara 5-6V tetapi UBEC memiliki keluaran arus yang besar. UBEC yang dipilih adalah UBEC Turnigy 15A 5-6V.

### Modul Regulator LM2596

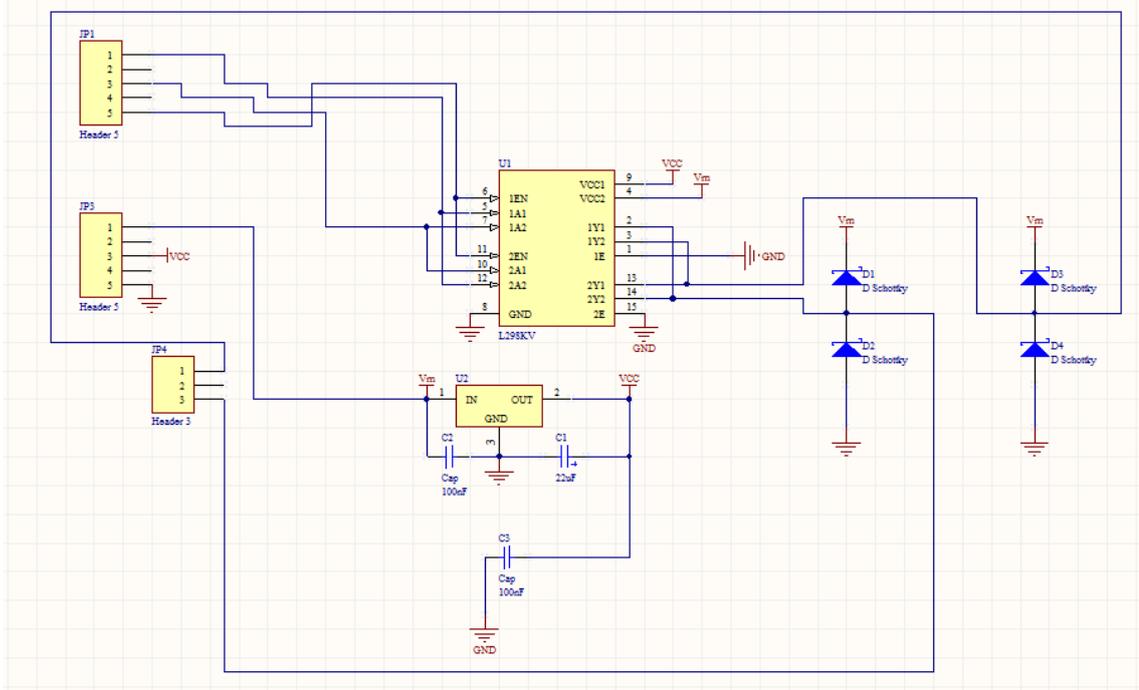
Regulator ini digunakan untuk menurunkan tegangan dari baterai ke Raspberry Pi. LM2596 digunakan karena dapat menghasilkan arus maksimal 3A, tegangan DC *input* 1,5-37 volt dan tegangan DC *output* 5 volt yang cukup untuk menjalankan Raspberry Pi yang membutuhkan tegangan 5 volt dan arus maksimal 1,2A. Gambar skematik rangkaian dapat dilihat pada gambar 4.



■ Gambar 4. Skematik LM2596 ke 5V

### Modul Motor Driver L298

Modul motor *driver* ini digunakan untuk mengendalikan gerakan motor. L298 membutuhkan tegangan *logic supply* 4,8V-7V dengan arus maksimal 70mA untuk IC L298, *input voltage* dan *enable voltage* 0,3V-7V dengan arus maksimal 100uA, *supply voltage* hingga 50V dengan arus keluaran maksimal hingga 3A ketika kedua *enable* digabungkan. Tegangan pada *logic supply voltage* berguna untuk mengaktifkan IC L298 dan *supply voltage* digunakan untuk memberi tegangan pada motor DC. Gambar skematik rancangan motor *driver* dapat dilihat pada gambar 5.



■ Gambar 5. Skematik Rancangan Motor Driver L298

### Realisasi Modul Badan Robot

Badan mekanik robot terbuat dari bahan *acrylic* dengan tebal 5 mm. Desain bentuk robot dirancang dengan menggunakan aplikasi Autodesk Inventor yang kemudian per bagian robot dicetak dengan suatu mesin laser khusus. Bagian utama robot terdiri dari tiga bagian yaitu bagian bawah, bagian tengah, dan bagian atas.

Bagian bawah robot berbentuk segi enam sama sisi dengan diameter 25,4 cm. Bagian bawah ini untuk meletakkan ketiga motor *driver*, ketiga motor DC dan caprit. Bagian tengah robot berbentuk segi enam sama sisi dengan diameter 25,4 cm dan terdapat sebuah lobang berbentuk lingkaran dengan diameter 2 cm untuk tempat kabel yang menghubungkan modul-modul dari bagian bawah robot ke modul-modul bagian atas dan tengah robot. Bagian tengah ini untuk meletakkan Raspberry Pi, *UBEC*, dan modul regulator LM2596.

Bagian atas robot berbentuk setengah segi enam sama sisi dengan diameter 25,4 cm. Bagian atas ini dipasang pada setengah bagian depan tengah robot. Bagian atas ini digunakan untuk tempat meletakkan baterai dan penopang servo yang terbuat dari *acrylic*. Pada penopang servo terdapat servo yang dipasang seperti Gambar 3.14. Pada bagian servo yang berputar terdapat penopang antara servo dan kamera yang terbuat dari *acrylic*. Gambar keseluruhan dapat dilihat pada gambar 1.

### Realisasi Program Pengendali Motor Driver DC

Realisasi program pengendali motor *driver* DC menggunakan sebuah *library* yang disebut WiringPi dan bahasa pemrograman C++ yang dapat langsung dibuat di LXterminal pada Raspberry Pi. WiringPi adalah sebuah *library* yang memudahkan pengguna untuk mengakses pin-pin GPIO pada Raspberry Pi. Informasi-informasi WiringPi dapat dilihat pada alamat *website* <http://WiringPi.com/>. Untuk mengontrol motor, Raspberry Pi harus dapat mengirim tiga sinyal dari tiga *pin input/output* (GPIO) ke satu motor *driver*. Dua pin dihubungkan ke kedua bagian *input voltage* motor *driver* dan satu pin ke bagian *enable voltage* dari motor *driver*. Untuk mengontrol keluaran *input/output* pin-pin pada Raspberry Pi digunakan suatu program dengan bahasa pemrograman C++ dan *library* khusus yang memudahkan mengakses alamat dan mengatur keluaran pin-pin GPIO pada Raspberry Pi.

### Realisasi Program Pengendali Motor Servo

Realisasi program pengendali servo didapat dengan menggunakan suatu *software* yaitu *software* servoblaster. Dengan adanya servoblaster dimungkinkan untuk mengatur keluaran pin dengan arus tertentu untuk membentuk suatu pulsa. Untuk penggunaannya biasanya harus menuliskan perintah langsung pada LXterminal oleh karena itu agar dapat mengendalikan motor servo melalui bahasa pemrograman C++ digunakan perintah *system*. Cara kerja servo adalah menerima pulsa dengan lebar pulsa tertentu setiap waktu tertentu. Sebagian besar servo bekerja pada lebar pulsa 1000us-2000us setiap 20ms.

### Realisasi Program Pengolah Gambar

Realisasi pengolahan gambar menggunakan suatu *library* yang disebut OpenCV. Untuk mempelajari fungsi-fungsi pada OpenCV dapat dilihat pada <http://opencv.org/documentation.html> atau

<http://docs.opencv.org/2.4/doc/tutorials/tutorials.html>. OpenCV memiliki sangat banyak perintah fungsi yang memudahkan untuk mengolah gambar oleh karena itu untuk mempercepat pembuatan sistem perancang menggunakan suatu program yang didapat dari internet buatan Kyle Hounslow yang perancang modifikasi agar dapat digunakan pada Raspberry Pi yang digunakan pada rancangan. Fungsi-fungsi yang terdapat pada program pengolahan gambar adalah fungsi `drawobject`, `morphos`, `trackfilteredobject`, dan utama. Program Kyle Hounslow dapat dilihat pada <https://github.com/kylehounslow/opencv-tuts/blob/master/object-tracking-tutorial/objectTrackingTut.cpp>.

### Penentuan Rentang Warna pada Model Warna HSV

Jenis model warna yang digunakan pada program adalah *Hue Saturation Value* (HSV) karena model warna ini lebih mudah diganti-ganti dan dimengerti dalam pencarian nilai warna karena tinggal memasukan nilai *hue* sesuai warna yang dilihat oleh mata kemudian tinggal mengatur *saturation* dan *value* sesuai dengan cerah-gelap gambar. *Hue* menyatakan warna asli/dasar dari gambar yang dilihat. *Saturation* menyatakan kecerahan gambar dari putih ke warna dasar. *Value* menyatakan seberapa gelap gambar dari hitam ke warna dasar. Pada program terdapat fungsi untuk kalibrasi warna. Pada fungsi seluruh warna *hue* dibagi menjadi 8 bit (0-255), begitu juga dengan *saturation* dan *value*. Selain itu untuk mendapat rentang dari suatu warna setiap *hue* dibagi menjadi *hue* untuk nilai paling rendah dan *hue* untuk warna paling tinggi yang ingin didapat begitu juga untuk *saturation* dan *value*.

Hal paling penting dalam kalibrasi warna kita harus mengetahui terlebih dahulu rentang dari warna dasar. Model warna HSV digambarkan dalam bentuk tabung dengan keliling lingkaran yang menyatakan warna-warna dasar yang disebut *Hue*, tinggi tabung yang menyatakan semakin hitam(gelap) atau semakin menuju warna dasar yang disebut *Saturation*, dan dari titik tengah lingkaran yang terang(putih) atau semakin menuju warna dasar. Rentang warna dasar ini dapat dilihat pada [https://en.wikipedia.org/wiki/HSL\\_and\\_HSV#From\\_HSV](https://en.wikipedia.org/wiki/HSL_and_HSV#From_HSV) [27]. Dari hasil perhitungan dari tabung warna ke 8 bit data yaitu (batas rentang warna yang dicari)/360\*256 diperoleh nilai dasar *hue* seperti pada Gambar 6.

Warna	Batas warna <i>hue</i>
Merah	233-255 dan 0-6
Jingga	6-35
Kuning	35-63
Hijau	63-105
Cyan	105-148
Biru	148-191
Magenta	191-233

■ Gambar 6. Tabel Nilai Rentang Warna Dasar *Hue* Dalam 8 bit

### Fungsi `Drawobject`

Potongan fungsi ini digunakan untuk membuat tanda/gambar ke tampilan window hasil tangkapan kamera terhadap warna benda yang ingin didapat. Pada rancangan fungsi yang digunakan adalah fungsi `line` untuk menggambar garis, fungsi `circle` untuk menggambar lingkaran, dan fungsi `putText` untuk menggambar tulisan. Format syntax untuk membuat lingkaran pada OpenCV adalah `circle(Mat& img, Point center, int radius, const Scalar& color, int thickness=1, int lineType=8, int shift=0)` Contoh penggunaan fungsi `circle` pada rancangan adalah `circle(frame, Point(x,y), 20, Scalar(0,255,0), 2);` `Circle` merupakan fungsi untuk menggambar lingkaran, `frame` adalah tempat yang akan digambar lingkaran, `Point(x,y)` adalah titik pusat lingkaran. 20 menyatakan jari-jari lingkaran, `Scalar(0,255,0)` menyatakan warna dari (r,g,b) jadi nilai yang tercantum digunakan untuk memberi warna hijau pada lingkaran, 2 terakhir menyatakan ketebalan garis lingkaran.

### Fungsi `Morphos`

Fungsi `morphos` yang terdiri dari fungsi `erode` dan `dilate`. Fungsi ini digunakan untuk menghilangkan *noise-noise* pada hasil tangkapan gambar.

Fungsi `erode` adalah fungsi yang membuat suatu ukuran *pixel* berwarna gelap melebar dengan cara membuat dahulu sebuah kernel atau `frame` `erode` yang kemudian dibagi menjadi beberapa kotak kecil atau lingkaran kecil dengan ukuran tertentu lalu `frame` `erode` tersebut ditumpuk ke `frame` utama, jika di suatu kotak `frame` `erode` terdapat warna hitam hampir sebesar kotak dan dalam kotak tersebut terdapat warna putih maka warna putih diubah menjadi warna hitam untuk memenuhi kotak `frame` `erode`.

Fungsi `dilate` adalah fungsi yang membuat suatu ukuran *pixel* dari sebuah `frame` yang berwarna terang melebar dengan cara membuat dahulu sebuah kernel atau `frame` `dilate` yang kemudian dibagi menjadi beberapa kotak kecil atau lingkaran kecil dengan ukuran tertentu lalu `frame` `dilate` tersebut ditumpuk ke `frame` utama, jika

di suatu kotak *frame* dilate terdapat warna putih hampir sebesar kotak dan dalam kotak tersebut terdapat warna hitam maka warna hitam diubah menjadi warna putih untuk memenuhi kotak *frame* erode.

### Fungsi *Trackfilteredobject*

Fungsi ini adalah fungsi utama untuk mendeteksi titik tengah/koordinat dan luas suatu objek dari *frame* yang didapat. Nilai titik tengah dan luas objek ini lah yang nantinya akan digunakan untuk mengetahui letak *object* yang dicari. Isis utama fungsi ini adalah perintah *findcontour*. Perintah *findcontour* memiliki format penulisan : *findContours(InputOutputArray image, OutputArray OfArrays contours, OutputArray hierarchy, int mode, int method);* ); *findContours* menyatakan fungsi mencari *contours*, *InputOutputArray image* menyatakan *output* dan *input frame* yang ingin dicari *contournya*. *OutputArrayOfArrays contours* menyatakan *contour-contour* yang didapat dari *InputOutputArray*. *OutputArray hierarchy* menyatakan *hierarchy-hierarchy* pada *InputOutputArray image*, *int mode* adalah *mode hierarrchy* yang digunakan. *Int method* adalah *metode contour* yang digunakan.

### Fungsi utama

Fungsi utama pada rancangan digunakan untuk memanggil dan menggunakan fungsi-fungsi yang telah dibuat dan berisi perintah-perintah untuk menyiapkan, mengatur, dan menangkap gambar dari kamera.

## REALISASI KESELURUHAN SISTEM

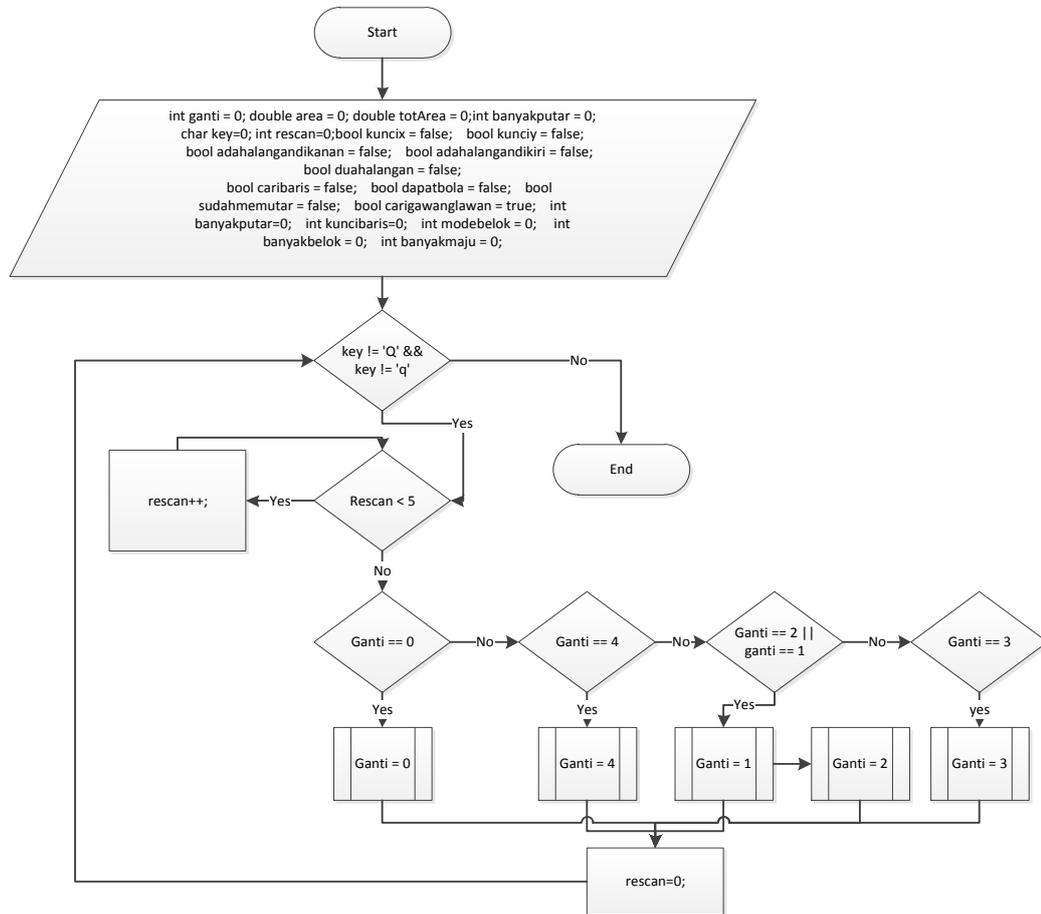
Realisasi keseluruhan sistem dilakukan dengan memasang seluruh modul. *Ground output* baterai dihubungkan ke *ground input UBEC* dan *VCC output* baterai dihubungkan ke *VCC input UBEC*. Keluaran *VCC output UBEC* dihubungkan ke pin VS atau *Vm input* pada motor *driver* belakang, kanan, dan kiri serta ke pin *VCC input Servo Futaba S3003* dan ke *VCC input LM2596*. *Ground output UBEC* dihubungkan ke *ground input Motor driver* belakang, kanan, dan kiri serta *ground* pada *input LM2596*. Keluaran dari *VCC output LM2596* dihubungkan ke *VCC input Raspberry Pi*. *Ground output LM2596* dihubungkan ke *ground input Raspberry Pi*. Setelah semua modul terpasang *switch* pada *UBEC* dipindah ke *on* sehingga arus dan tegangan dapat mengalir ke *output UBEC* dan dapat memberi daya seluruh modul yang lain. Setelah *Raspberry Pi* selesai *booting* maka program yang dirancang bekerja.

Diagram alir program untuk mengatur gerakan robot yang dirancang dapat dibagi menjadi beberapa *flowchart*. *Flowchart* yang menghubungkan antar *flowchart* disebut *flowchart* utama yang dapat dilihat pada Gambar 3.37. *Flowchart* lainnya adalah *flowchart* ganti 0, ganti 4, ganti 1/2, dan ganti 3. *Flowchart* utama program keseluruhan robot dapat dilihat pada gambar 7.

*Flowchart* ganti=0 adalah *flowchart* algoritma untuk mengolah data berdasarkan warna bola dan menentukan arah ke *flowchart* lain atau menggerakkan robot. Inti dari program algoritma ganti = 0 adalah untuk mencari bola, mendekati bola, menuju tepat depan bola, dan membawa bola menuju gawang lawan jika bola sudah tepat depan robot. *Flowchart* ganti = 0 dapat dilihat pada gambar 8.

*Flowchart* ganti=4 adalah *flowchart* algoritma untuk mengolah data berdasarkan warna halangan dan menentukan arah ke *flowchart* lain atau menggerakkan robot. Inti dari program algoritma ganti = 4 adalah untuk menghindari satu atau dua halangan ke daerah yang tidak terhalang. *Flowchart* ganti = 4 dapat dilihat pada gambar 9.

*Flowchart* ganti=1/2 adalah *flowchart* algoritma untuk mengolah data berdasarkan warna gawang lawan/sendiri dan menentukan arah ke *flowchart* lain atau menggerakkan robot. Inti dari program algoritma ganti = 1 atau ganti = 2 adalah untuk menuju gawang lawan jika sudah mendapatkan bola dan ganti = 1 atau 2 untuk melakukan perpindahan posisi di lapangan jika tidak menemukan bola di daerah lainnya. *Flowchart* ganti = 1 atau ganti 2 dapat dilihat pada gambar 10.



■ Gambar 7. Flowchart Utama

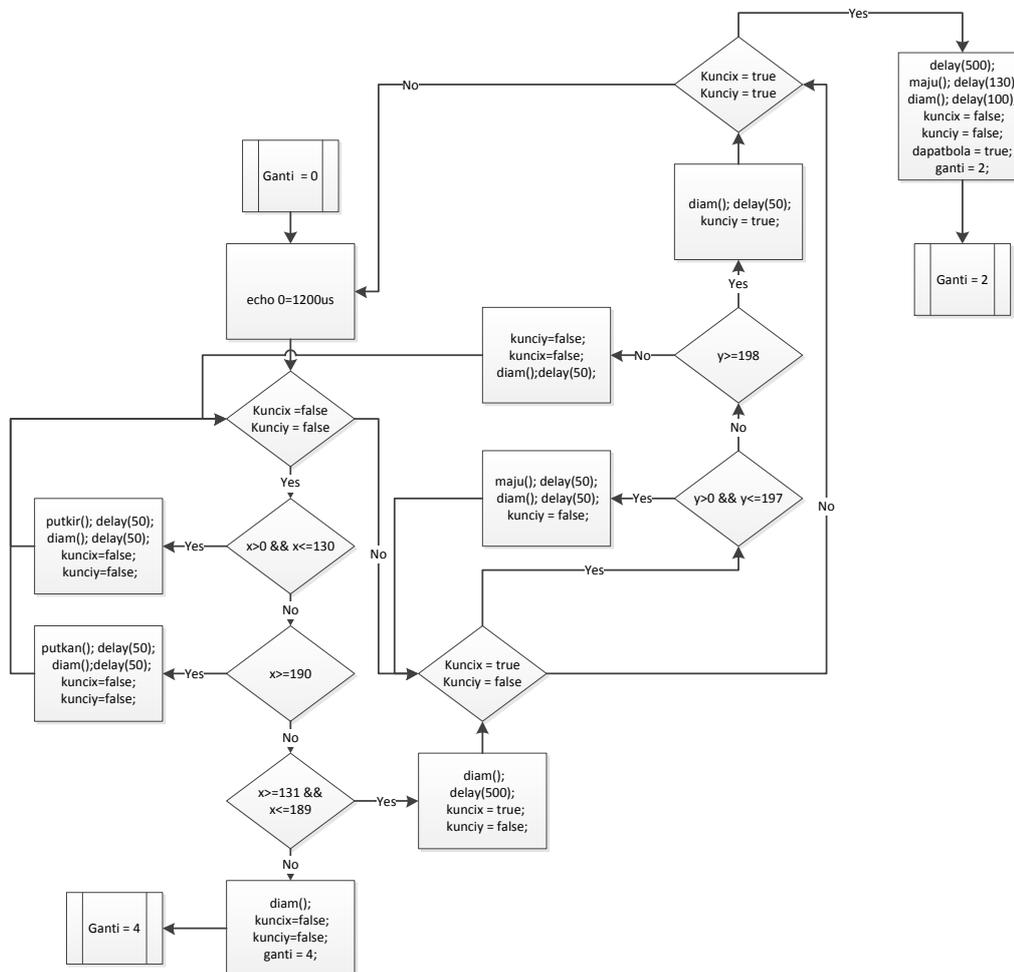
Flowchart ganti=3 adalah flowchart algoritma untuk mengolah data berdasarkan warna lapangan dan menentukan arah ke flowchart lain atau menggerakkan robot. Inti dari program algoritma ganti = 3 adalah untuk menjaga robot tetap di dalam lapangan dan mencegah robot berputar ditempat terus menerus karena tidak menemukan bola. Flowchart ganti = 3 dapat dilihat pada gambar 11.

### Pengujian Gerak Robot

Pengujian dilakukan untuk mengetahui apakah dapat menggerakkan robot maju, diam, dan berputar di tempat kanan/kiri. Pengujian dilakukan dengan seluruh modul telah terpasang dengan bantuan program pengujian gerak. Tabel hasil pengujian dapat dilihat pada Tabel 1. Dari hasil pengujian dapat dinyatakan seluruh gerakan pada rancangan dapat dilakukan yaitu maju, berputar tempat kanan/kiri, dan diam.

■ Tabel 1 Hasil Pengujian Gerak Robot

Motor Belakang		Motor Kanan		Motor Kiri		Pergerakan yang dihasilkan	Digunakan/ tidak
Input pin 2	Input pin 3	Input pin 13	Input pin 14	Input pin 10	Input pin 11		
0	0	0	0	0	0	Diam	Ya
0	0	1	0	0	1	Maju lurus	Ya
0	1	0	1	0	1	Putar ditempat ke kanan/searah jarum jam	Ya
1	0	1	0	1	0	Putar ditempat ke kiri/berlawanan searah jarum jam	Ya



■ Gambar 8. Flowchart Ganti = 0

**Pengujian dan Analisis Seluruh Sistem**

Pengujian seluruh sistem dilakukan untuk mengetahui apakah hasil rancangan dapat sesuai dengan tujuan rancangan atau tidak. Pengujian seluruh sistem dilakukan dengan beberapa tahap. Tahap berpindah tempat adalah Pengujian berpindah tempat tanpa bola dan halangan. Tabel Hasil pengujian dapat dilihat pada tabel 3.

**Pengujian dan Analisis Kalibrasi Warna**

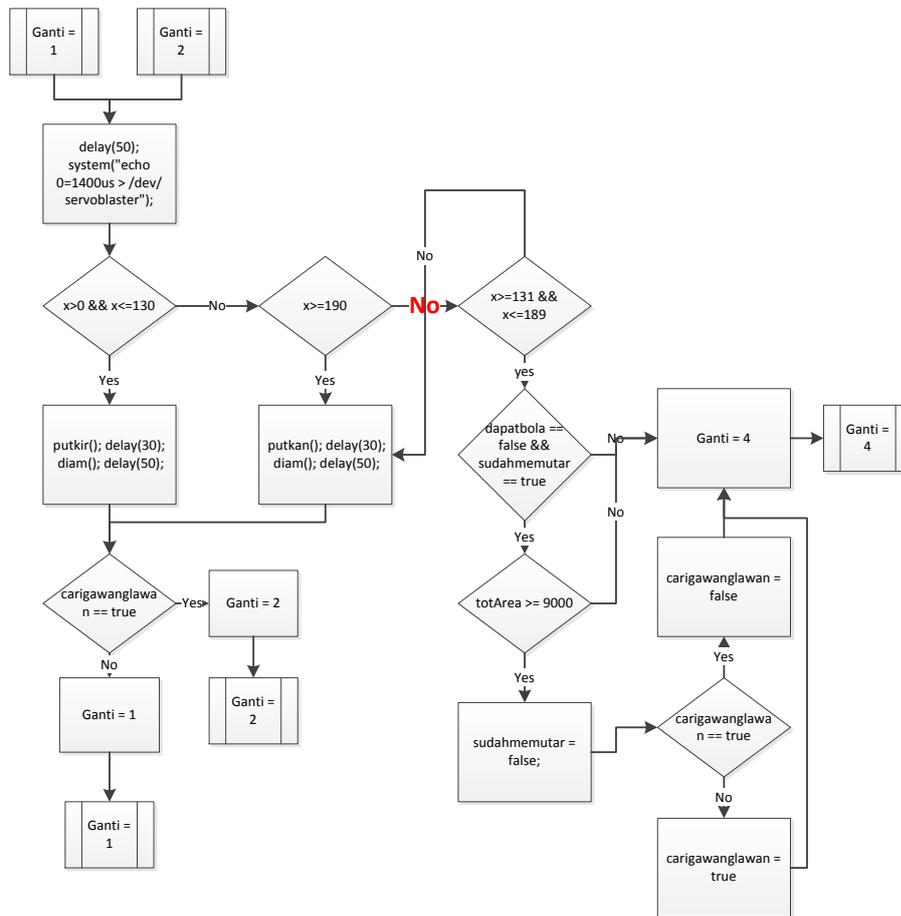
Pengujian bertujuan untuk menentukan apakah semua warna yang digunakan dapat dibedakan dan mendapat nilai HSV yang harus di-input ke program untuk dapat membaca warna yang dicari. Pengujian dilakukan dengan meletakkan seluruh warna berdempetan hingga dapat dilihat semua oleh kamera dan menggunakan sebuah software yang dapat diubah nilai-nilai HSV nya. Tabel hasil pengujian dapat dilihat pada Tabel 2. Dari hasil pengujian dapat dinyatakan sistem dapat membaca dan membedakan warna-warna yang ditentukan dengan baik.

■ Tabel 3 Hasil Pengujian Serluruh Sistem Berpindah Daerah

No	Posisi	Berhasil ke gawang lawan	Berhasil ke gawang Sendiri	Berhasil ke gawang lawan lagi
1	Dekat gawang lawan	ya	ya	Ya
2	Dekat Gawang Sendiri	ya	ya	Ya

Dari hasil pengujian pada Tabel 3 dapat dinyatakan robot dapat berpindah dari daerah dekat gawang lawan ke daerah dekat gawang sendiri untuk mencari bola pada daerah dekat gawang lawan dan dekat daerah gawang sendiri dengan baik tanpa halangan.





■ Gambar 10. Flowchart Ganti = 1 atau Ganti 2

Pengujian selanjutnya adalah pengujian untuk menghindari halangan. Dari hasil pengujian pada Tabel 4 dapat dinyatakan robot berhasil menghindari halangan ke daerah yang lebih tidak terhalang oleh halangan.

■ Tabel 4 Hasil Pengujian Menghindari Halangan

Uji ke	Halangan satu	Halangan dua	Gerak robot setelah halangan 1 tertangkap	Gerak robot setelah halangan 2 tertangkap
1	-	-	Putar kanan	Tidak diuji
2	kiri	-	Putar kiri	Tidak diuji
3	tengah	-	Putar kanan	Tidak diuji
4	kiri	kanan	Putar kiri	Putar kiri
5	tengah	kanan	Putar kanan	Putar kanan
6	tengah	tengah	Putar kanan	Tidak diuji

Pengujian selanjutnya adalah pengujian untuk mendekati bola, menghindari halangan, dan menggiring bola ke gawang lawan. Dari hasil pengujian tabel 5 dapat dinyatakan rancangan dapat memenuhi tujuan rancangan yaitu membuat robot yang dapat mencari bola, mendekati bola, dan menjauhkan bola dari daerah lapangan/gawang sendiri ke arah daerah lapangan/gawang lawan dengan memperhatikan juga posisi halangan yaitu dengan cara dapat menghindari halangan ke ruang yang lebih luas dan posisi robot di lapangan dengan cara dapat berpindah-pindah tempat dari daerah dekat gawang sendiri ke daerah dekat gawang lawan dan sebaliknya.

■ Tabel 5 Hasil Pengujian Mendekati, Menghindari, dan Menggiring Bola ke Gawang Lawan

Uji ke	Memasukan ke Capit	Menghindari Halangan	Menggiring Bola ke Arah Lawan	Menggiring Bola Menempel ke Tiang Gawang
1	Berhasil	Berhasil	Berhasil	Berhasil
2	Berhasil	Berhasil	Berhasil	Tidak
3	Berhasil	Berhasil	Berhasil	Tidak

