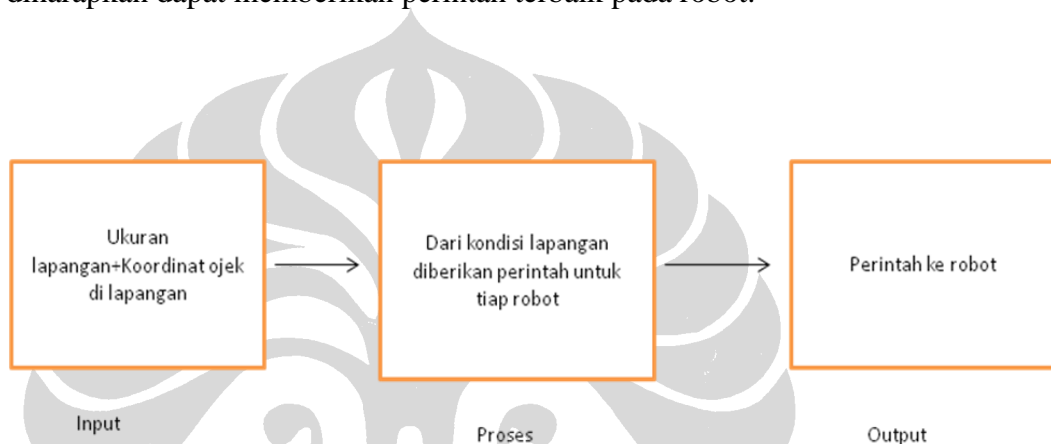


BAB 3

PERANCANGAN

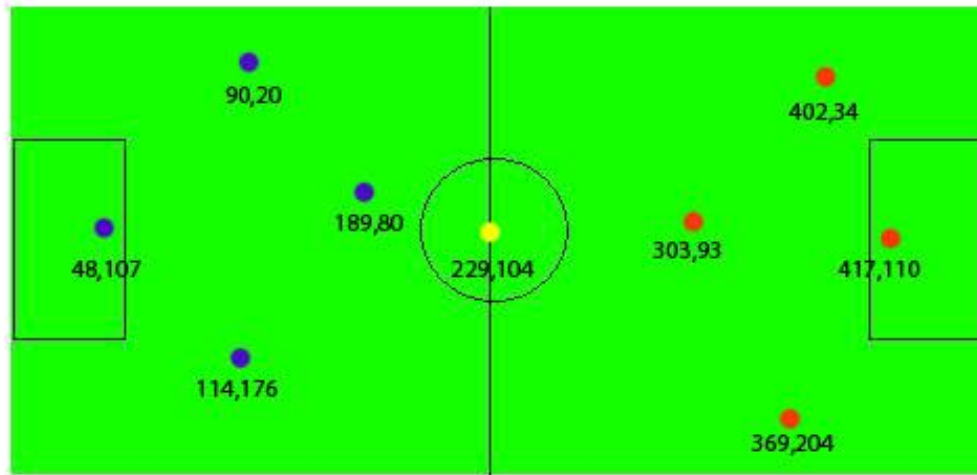
3.1 Desain Alur Penentuan Keputusan Robot

Aplikasi ini bertujuan untuk menentukan perilaku robot yang diinginkan dalam pertandingan sepak bola antar robot. Dari berbagai kondisi lapangan, aplikasi ini diharapkan dapat memberikan perintah terbaik pada robot.



Gambar 3.1 Diagram Alur Utama

Input yang dibutuhkan adalah sebuah array yang berisi koordinat robot sendiri, koordinat robot musuh, koordinat bola, dan arah menghadap robot sendiri. Berdasarkan input yang masuk, dihitung kemungkinan gol dan kemungkinan operan(*passing*), serta gerakan motor yang diperlukan untuk melakukan kedua hal tersebut. Selanjutnya, hasil perhitungan diteruskan sebagai perintah gerakan untuk robot yang dikirimkan melalui Bluetooth.



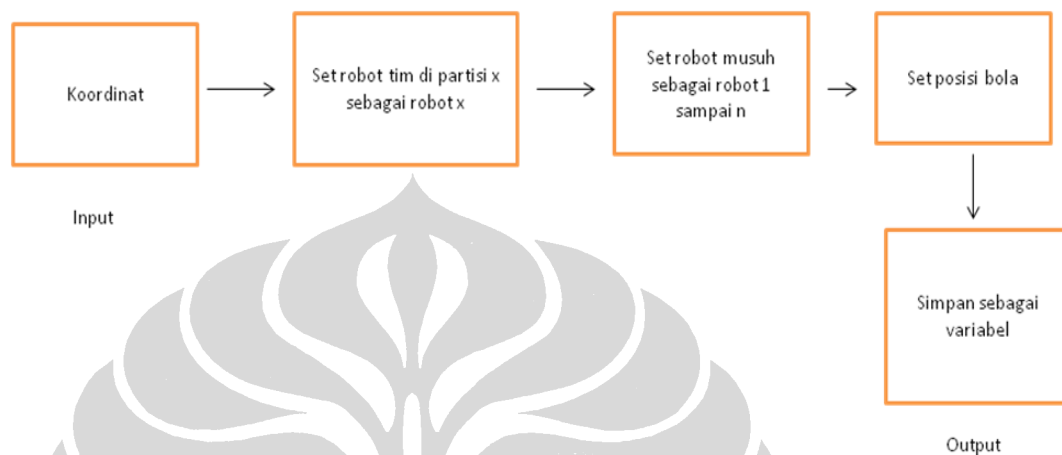
Gambar 3.2 Contoh Koordinat di Lapangan

3.2 Penentuan Input

Input yang dibutuhkan adalah hasil dari pemrosesan gambar yang ditangkap oleh kamera yang terletak diatas lapangan. Gambar tersebut harus dianalisa untuk mendapatkan posisi robot musuh, robot sendiri, dan bola dalam bentuk koordinat. Selain posisi, arah menghadap robot juga perlu dicari dari input yang diterima. Selanjutnya, data-data tersebut disimpan dan dikirimkan ke aplikasi ini dalam bentuk sebuah array.

Input array yang masuk dianggap sebagai daftar posisi robot dan bola dalam koordinat (x,y) dengan asumsi lapangan tim berada di sebelah kanan dan titik (0,0) berada di pojok kiri bawah. Pemodelan seperti ini dipilih karena memudahkan untuk pembagian partisi lapangan selanjutnya. Selain itu, pemodelan seperti ini juga akan memudahkan penghitungan dalam algoritma yang akan dipakai.

Dengan menggunakan pemodelan lapangan dalam bentuk koordinat, penghitungan jarak antar robot, sudut tembakan, dan analisa posisi musuh dapat dihitung dengan menggunakan persamaan garis dalam bidang kartesius.



Gambar 3.3 Diagram Alur Input

Setiap robot memiliki nomor urut untuk membedakan robot satu dengan robot lain saat melakukan sambungan antara komputer dengan robot. Saat melakukan sambungan, robot akan membutuhkan sebuah *port* pada komputer yang digunakan. . Setiap robot memiliki nomor *port* yang berbeda satu sama lain. Nomor urut robot merepresentasikan nomor *port* yang dipakainya. Namun nomor urut tidak harus sama dengan nomor robot.

Tapi input yang diterima belum tentu sudah terurut. Karena itu, urutannya perlu diperbaiki. Walau begitu, ada beberapa hal yang sudah diketahui dalam pemasukan input. Diantaranya, 8 variabel pertama merupakan posisi robot sendiri. 8 variabel berikutnya merupakan posisi robot musuh. 2 posisi selanjutnya merupakan posisi bola. Dan 4 variabel terakhir merupakan arah menghadap robot.

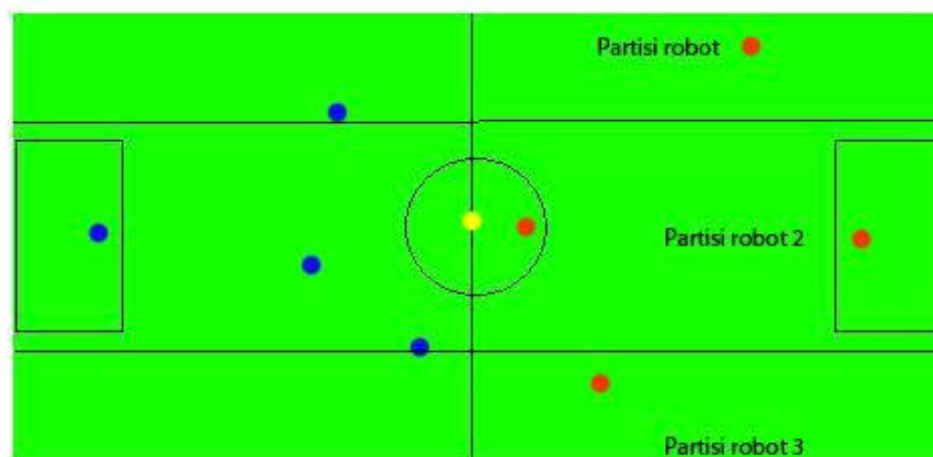
Mula-mula, input koordinat robot dan arah menghadapnya harus diurutkan sesuai posisinya untuk memudahkan dalam mengenali robot mana yang harus dijalankan. Jika urutannya ada yang tertukar, maka isi variabelnya akan di tukar dengan isi variabel lain yang belum diketahui sudah terurut atau tidak. Jika masih

belum terurut, proses akan diulang sampai variabel ini benar. Proses ini dilakukan untuk semua variabel posisi robot sampai semuanya terurut

Namun karena komunikasi tidak diperlukan untuk robot musuh, maka hanya robot sendiri yang harus diurutkan. Posisi robot musuh dianggap sudah sesuai dengan nomor urutnya. Lagipula tidak menjadi masalah bila sampai tertukar urutannya karena sistem ini tidak berkomunikasi dengan robot musuh. Setelah setiap koordinat dipasangkan dengan partisinya, barulah koordinat dapat dianalisa.

3.3 Penentuan Partisi

Untuk memudahkan penugasan robot, digunakan algoritma pembagian lapangan untuk setiap robot pemain. Lapangan yang ditugaskan kepada tiap robot dinamakan partisi. Setiap robot memiliki partisi lapangannya sendiri. Tiap partisi lapangan juga memiliki nomor partisi yang sama dengan robot yang menjaganya. Partisi ini tidak hanya digunakan untuk menentukan ruang gerak robot, tetapi juga untuk mengetahui nomor robot yang berada di lapangan.

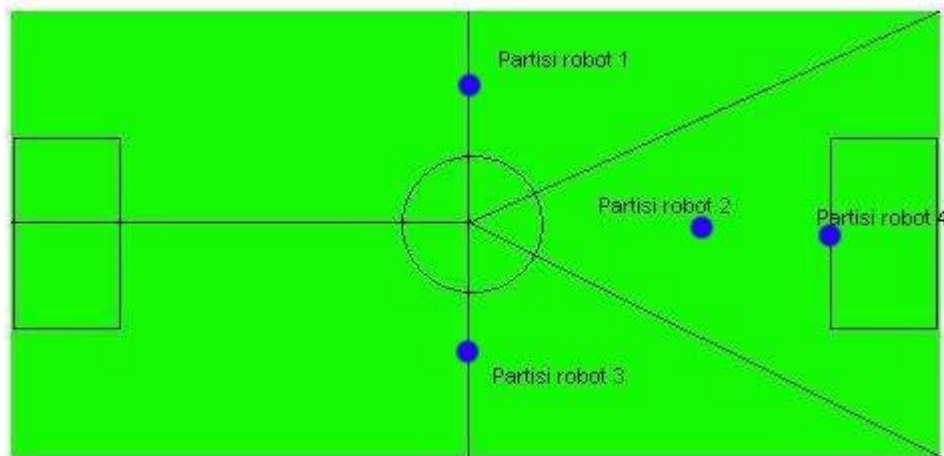


Gambar 3.4 Contoh Partisi Robot

Partisi juga digunakan untuk menentukan robot mana yang harus mengejar bola. Untuk itu, koordinat bola harus dicek dahulu. Karena pemodelan lapangannya menggunakan koordinat, maka pembagian partisinya dapat direpresentasikan sebagai persamaan garis lurus yang membagi lapangan. Posisi robot dianggap sebagai sebuah titik di bidang kartesius dengan lapangan sebagai bidang kartesiusnya. Daerah partisi robot adalah daerah di lapangan yang dibatasi oleh beberapa garis lurus. Dan untuk mengecek nomor partisi dimana suatu koordinat berada, digunakan persamaan linear sederhana.

Daerah partisi tiap robot dapat saling lepas atau saling tindih. Jika daerahnya saling lepas, maka tiap robot dapat lebih fokus ke daerahnya masing-masing dan hal ini akan mencegah saling rebut bola antara pemain sendiri. Selain itu, jika hanya satu robot yang mengejar bola maka robot lain dapat bersiaga untuk menjaga daerah belakang lapangan atau bersiap untuk menerima operan bola.

Pembagian ini juga dapat mencegah robot berkumpul di satu bagian lapangan dan membiarkan bagian lain kosong. Dengan pembagian ini, setiap bagian lapangan dijaga oleh 1 robot. Karena itu, jarak tempuh rata-rata sebuah robot untuk menuju suatu titik akan berkurang. Dengan pembagian yang tepat, jarak terjauh yang harus ditempuh robot juga akan berkurang jauh. Robot tidak harus berjalan dari ujung lapangan satu ke ujung lapangan lain.



Gambar 3.5 Contoh Posisi Standar Robot

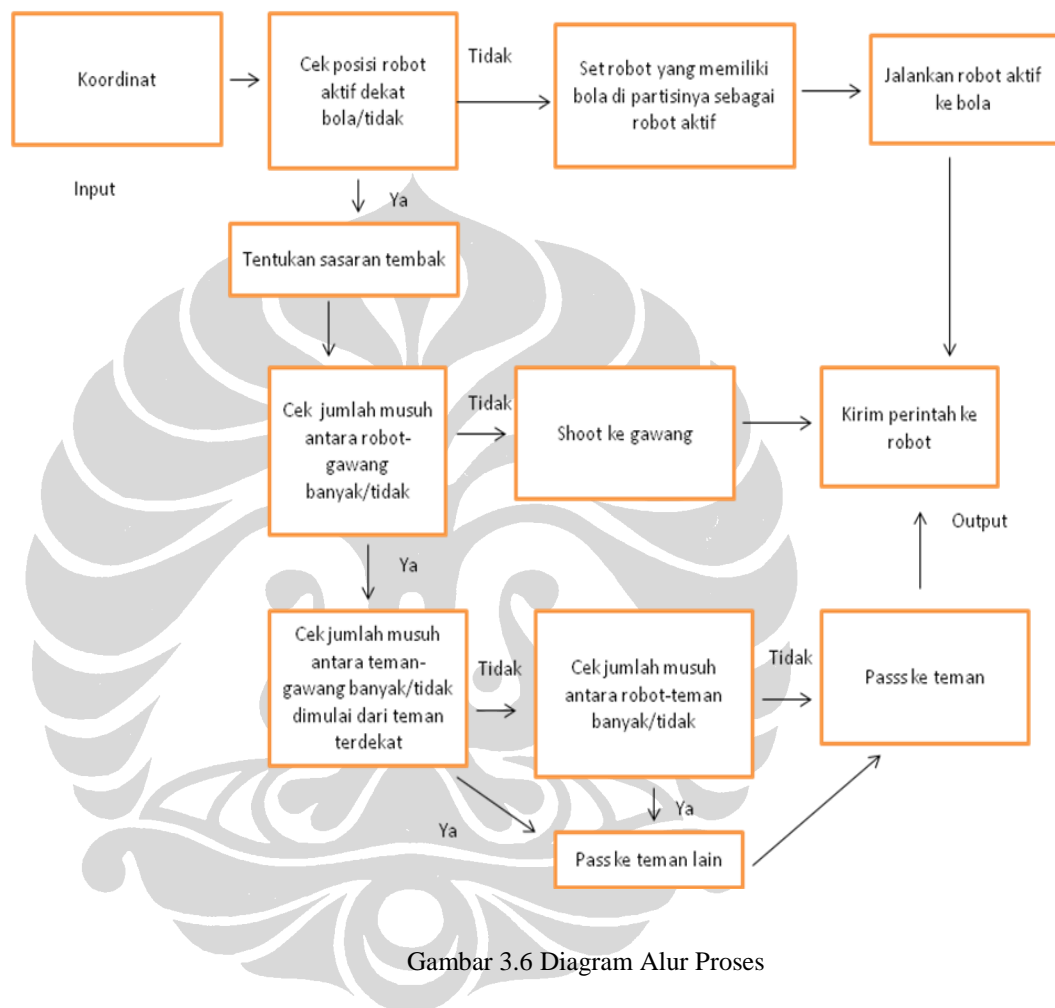
Untuk lebih mengefisienkan gerakan, tiap robot memiliki posisi standar. Robot akan menuju ke posisi ini jika tidak mengejar bola. Kecuali penjaga gawang, karena penjaga gawang harus selalu bersiaga terhadap bola yang datang. Posisi standar ini sebisa mungkin berada di tengah lapangan, karena posisi ditengah lapangan akan meminimalkan jarak terjauh yang harus ditempuh robot.

Namun dalam algoritma ini partisi yang digunakan oleh robot sedikit mengalami saling tindih. Hal ini diperlukan untuk mengantisipasi apabila robot secara tidak sengaja melenceng sedikit dari daerahnya. Apabila lapangannya saling lepas, kemungkinan robot tersebut akan bertukar dengan robot lain saat kedua robot Nerada berdekatan di garis batas. Lagipula, ini juga membuat robot lebih cepat bereaksi saat bola bergulir ke lapangannya.

3.4 Proses Pengambilan Keputusan

Setelah posisi tiap robot diisi dengan koordinat yang sesuai, ditentukan sebuah robot aktif yang bertugas mengejar bola. Untuk memudahkan pemilihan, robot selain penjaga gawang yang memiliki bola di partisinya adalah robot aktif. Selanjutnya ditentukan sasaran tembak bola. Sasaran tembak adalah posisi gawang yang menjadi tujuan tembak. Sasaran tembak ditentukan terhadap posisi

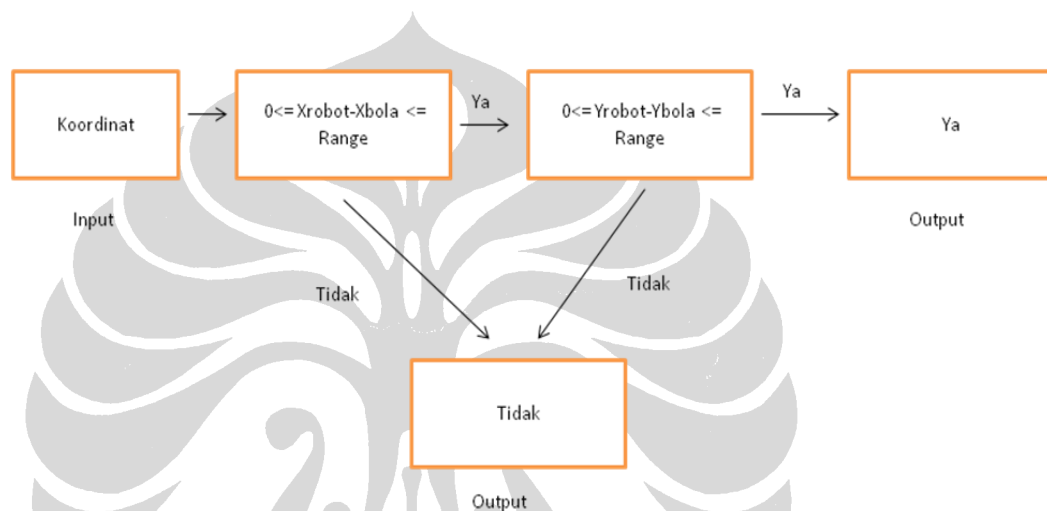
bola. Jika bola berada di bagian atas lapangan, maka sasaran tembak adalah bagian atas gawang. Dan sebaliknya bila bola berada di bagian bawah lapangan maka sasaran tembak adalah bagian bawah gawang.



Gambar 3.6 Diagram Alur Proses

Setelah itu di cek apakah posisi robot aktif berada di daerah tembak atau tidak. Daerah tembak adalah daerah di sekeliling bola yang dianggap cukup dekat dengan bola untuk bersiap melakukan tembakan ke arah gawang. Jika robot masih berada jauh dari bola, maka robot akan berjalan ke arah bola, namun jika tidak, berarti robot siap untuk menembak. Selanjutnya, jika robot sudah berada di daerah tembak, jumlah musuh yang berada diantara bola dan gawang dihitung. Kalau jumlah musuh sedikit, perintah menembak diberikan ke robot yang aktif, namun jika banyak, bola akan dioper ke robot lain.

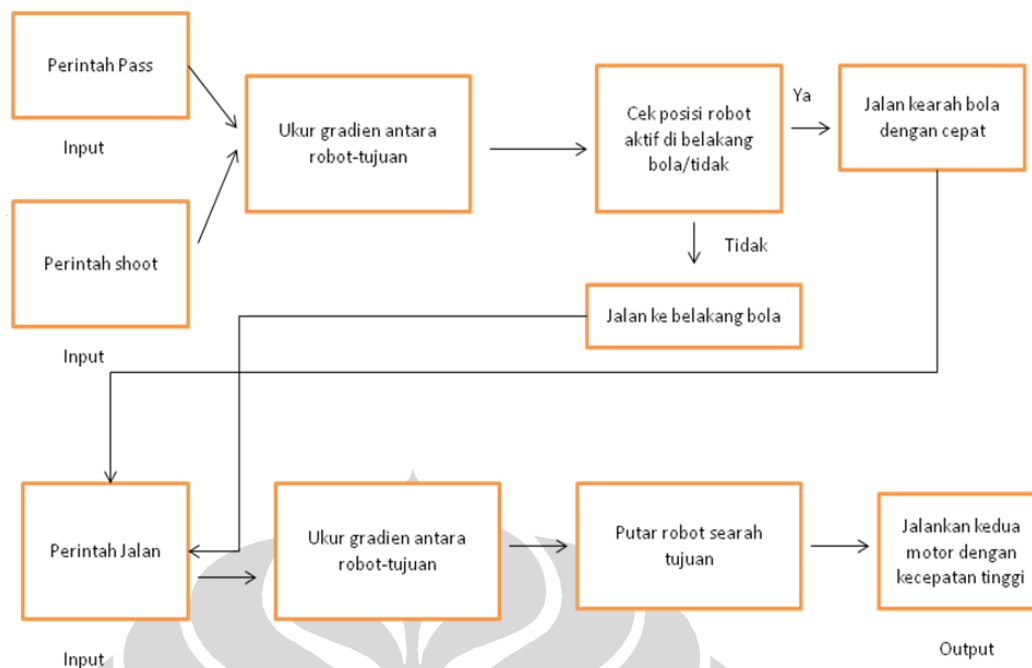
Setelah perintah untuk robot aktif selesai diberikan, robot lain dalam tim diberikan perintah untuk kembali ke posisi standar. Khusus untuk robot penjaga gawang, posisinya tergantung letak bola. Jika bola berada di bagian atas lapangan, maka penjaga gawang juga harus berada di bagian atas gawang dan begitu juga sebaliknya. Hal ini dilakukan agar posisi robot sebisa mungkin berada di tengah partisi lapangannya untuk meminimalisir jarak tempuh robot saat bola berpindah ke partisinya.



Gambar 3.7 Diagram Alur Pengecekan Posisi

Untuk mengecek apakah Robot berada di daerah tembaknya atau tidak, koordinat robot dikurangi dengan koordinat bola. Jika selisih nilai koordinat x dan y masuk dalam jarak yang diinginkan, maka robot tersebut dianggap berada di dalam daerah tembak. Namun bila tidak, berarti robot tersebut masih cukup jauh dari bola. Untuk menentukan luas daerah tembak, digunakan nilai range.

3.5 Proses Penentuan Output



Gambar 3.8 Diagram Alur Penentuan Output

Untuk menentukan output apa yang harus diberikan, mula-mula ditentukan dulu aksi apa yang harus dilakukan robot. Ada 3 jenis aksi yang dapat dilakukan robot tergantung dari proses pengambilan keputusan yang dilakukan sebelumnya. Aksi itu antara lain perintah oper(*passing*), perintah tembak(*shooting*), serta perintah jalan.

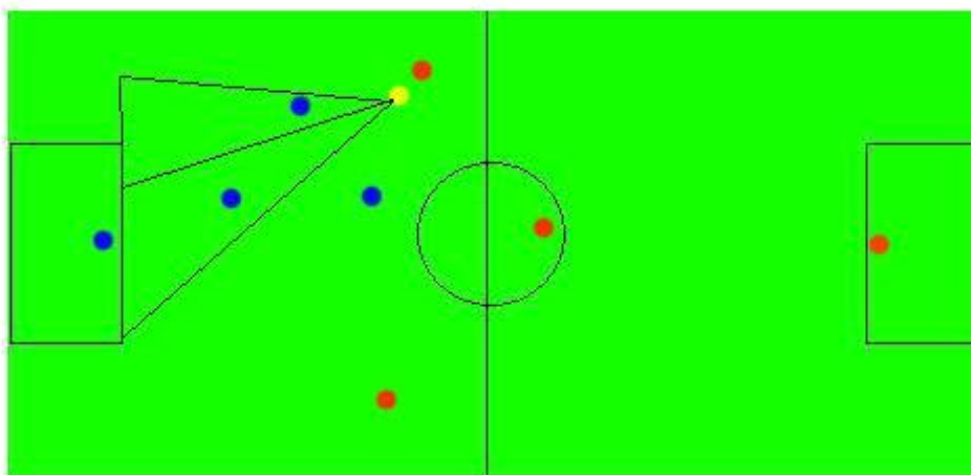
Jika robot akan melakukan perintah jalan, mula-mula diukur gradien antara robot dengan tujuannya, Selanjutnya robot akan diputar searah dengan tujuannya. Caranya dengan memberi perintah pada robot untuk memutar salah satu motornya sampai menghadap arah yang tepat. Jika robot sudah menghadap ke arah tujuan, maka robot akan diberi perintah untuk menjalankan kedua motornya sampai tiba di tujuan.

Jika robot akan melakukan perintah oper, maka gradien garis antara robot aktif dengan robot yang menjadi tujuan oper akan dihitung. Selanjutnya posisi robot aktif akan dicek apakah berada di belakang bola atau tidak. Arah belakang disini adalah arah yang berlawanan dengan arah tembak bola. Bila belum, maka robot

akan diberikan perintah jalan ke belakang bola. Namun jika sudah maka robot diberikan perintah jalan kearah bola untuk menembak bola.

3.6 Pengecekan Jumlah Musuh

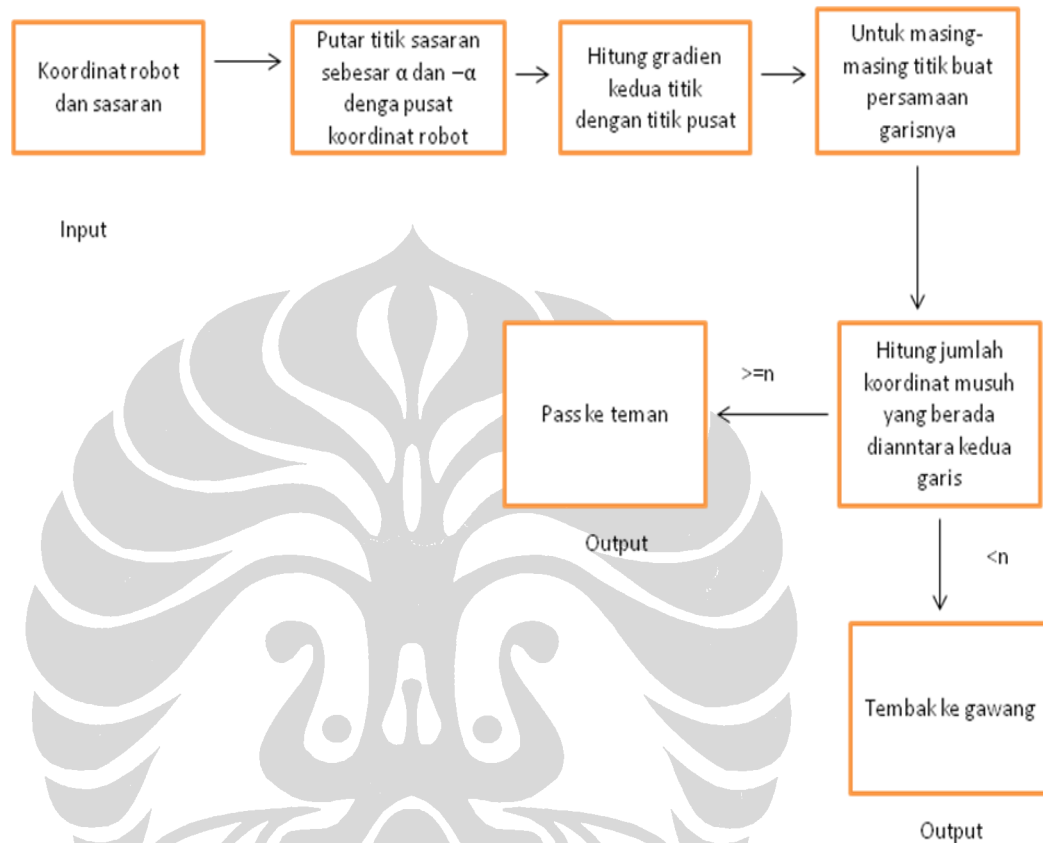
Untuk menghitung jumlah musuh yang berada antara robot dengan gawang, dibutuhkan input berupa koordinat robot dan koordinat sasaran, serta koordinat tiap musuh. Penghitungan ini berguna untuk mengetahui berapa banyak robot yang mengganggu tembakan robot. Namun tidak semua musuh yang ada di depan robot mempengaruhi tembakan robot dan harus dihitung. Hanya musuh yang mungkin dapat menahan laju bola saja yang harus dihitung.



Gambar 3.9 Contoh Pengecekan Jumlah Musuh

Di lapangan, setelah kita menembak bola, maka ada kemungkinan robot musuh akan berjalan menghampiri bola dan mengganggu tembakan bola. Daerah yang berisi musuh yang mengganggu tembakan bola dianggap berbentuk segitiga. Hal ini disebabkan saat bola mencapai titik tertentu, dia membutuhkan waktu n detik. Dalam n detik itu, robot lain dapat berjalan ke titik tersebut sejauh s . Karena nilai

n akan makin besar semakin jauh titik dari asal tembakan, maka nilai s juga akan semakin besar. Karena itu, daerah yang dibentuknya akan menyerupai segitiga.



Gambar 3.10 Diagram Alur Pengecekan Jumlah Musuh

Untuk menghitungnya, mula mula titik sasaran kita putar sejauh α dan $-\alpha$ dengan titik pusat posisi bola. Implementasinya dengan menggunakan rumus matriks.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

atau dalam persamaan linear

$$x' = x \cos \theta - y \sin \theta \quad (3.1)$$

$$y' = -x \sin \theta + y \cos \theta \quad (3.2)$$

namun karena titik pusatnya bukan (0,0), maka rumus matriksnya menjadi

$$\begin{bmatrix} x' - x_1 \\ y' - y_1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x - x_1 \\ y - y_1 \end{bmatrix}$$

atau dalam persamaan linear

$$(x' - x_1) = (x - x_1) \cos \theta - (y - y_1) \sin \theta \quad (3.3)$$

$$(y' - y_1) = -(x - x_1) \sin \theta + (y - y_1) \cos \theta \quad (3.4)$$

Selanjutnya hitung gradien garis antara masing-masing titik dengan koordinat bola dengan menggunakan rumus

$$m = \frac{y - y_1}{x - x_1} \quad (3.5)$$

Setelah gradien garis diketahui, kita dapat membuat persamaan garisnya dengan pola seperti berikut

$$y = mx + c \quad (3.6)$$

Dari persamaan itu, variabel c untuk kedua garis dapat diketahui dengan memasukkan kedua titik ke persamaan masing-masing.

Setelah itu, setiap koordinat robot musuh dicek satu persatu. Jika koordinatnya berada diantara kedua garis, maka robot tersebut berpotensi untuk menghalangi bola. Setelah semua robot musuh selesai diperiksa, jumlah robot musuh yang berpotensi menghalangi jalannya bola akan dikembalikan sebagai hasil fungsi.