

DAFTAR TABEL

| | |
|---|----|
| Tabel 4.1 Pengujian Jarak Jalur Tempuh | 26 |
| Tabel 4.2 Pengujian Jarak Sampah Terdekat | 27 |
| Tabel 4.3 Pengujian Sudut Belok/Putar Robot..... | 28 |
| Tabel 4.4 Pengujian Arah Putar Robot..... | 29 |

BAB I PENDAHULUAN

1.1. Latar Belakang Masalah

Robot adalah salah satu penerapan teknologi praktis yang umum digunakan karena dapat menggantikan peran operator manusia, menggantikan peran manusia dari pekerjaan yang sifatnya berulang, berbahaya, jangka waktu lama, dan dalam lingkungan beresiko tinggi. Saat ini penggunaan robot sudah mulai banyak digunakan dalam kehidupan industri dan sehari-hari, robot yang digunakan di sektor industri seperti aplikasi *Automated Guided Vehicle* (AGV) di pabrik, robot tersebut digunakan untuk aplikasi pemindah barang mengurangi faktor kelelahan dan pekerjaan repetitif untuk operator manusia. Contoh aplikasi robot dalam kehidupan sehari-hari seperti robot penyedot debu otomatis di rumah, robot tersebut menggunakan sensor ultrasonik untuk memberikan informasi gerak robot.

Dalam penelitian ini fokus pada aplikasi robot penyedot debu otomatis. Penggunaan robot penyedot debu umumnya melakukan pembersihan secara menyeluruh di setiap sela ruangan, tidak pada titik selektif yang dapat dipilih. Hal ini menyebabkan pemborosan energi baterai, keausan komponen robot karena robot melakukan pembersihan di area yang tidak perlu. Selain itu dengan sistem kerja seperti yang telah dipaparkan membuat aplikasinya menjadi kurang efektif apabila diterapkan di ruangan yang berskala besar seperti gedung olah raga atau aula serbaguna, karena robot melakukan pembersihan total di titik-titik yang masih bersih dan cukup luas areanya sehingga berpotensi merusak robot itu sendiri dan aktivitas pembersihan menggunakan robot di ruangan besar menjadi lebih lama.

Dari sini diusulkan sistem pembersihan titik selektif untuk robot menggunakan sistem navigasi pendeteksi objek mandiri, titik-titik yang terdapat sampah dideteksi oleh sensor citra, kemudian robot diarahkan ke titik-titik tersebut dengan optimasi jalur menggunakan algoritma bubble sort. Robot kemudian memungut hanya ke titik yang terdeteksi sebagai objek sampah oleh kamera. Dengan optimasi navigasi jalur pembersihan untuk titik-titik pembersihan tersebut kerja robot menjadi lebih efisien, jalur yang ditempuh lebih singkat, sehingga kerja robot lebih cepat dan komponen robot tidak mudah aus karena beban waktu kerja yang dipersingkat.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang sudah dijelaskan, masalah yang ingin dikaji dalam penelitian ini adalah:

1. Bagaimana pengenalan sampel sampah dan robot dengan sensor citra?
2. Bagaimana sistem mengidentifikasi lokasi sampel sampah dan robot?
3. Bagaimana sistem mengimplementasi jalur gerak robot?
4. Bagaimana sistem dapat mengetahui informasi untuk robot bergerak?

1.3. Tujuan dan Manfaat

Untuk memecahkan masalah yang telah dirumuskan penelitian ini memiliki tujuan sebagai berikut:

1. Membuat sistem identifikasi sampel sampah dan robot berbasis citra.
2. Membuat sistem deteksi lokasi sampel sampah dan robot dengan lokalisasi lapangan berukuran 115 cm × 153 cm.
3. Membuat sistem pemetaan jalur untuk robot bergerak dari titik awal, kemudian menuju ke semua sampel sampah hingga mencapai titik homebase yang telah ditentukan.
4. Membuat sistem untuk mengetahui informasi untuk robot bergerak, yaitu arah belok/putar, besar sudut belok/putar, dan jarak tempuh.

1.4. Batasan Masalah

Dalam perancangannya robot ini memiliki batasan-batasannya, yaitu:

1. Adanya sampel sampah yang terdeteksi.
2. Pengenalan citra objek hanya menggunakan deteksi warna.
3. Tidak adanya halang rintang di dalam area lapangan.
4. Robot yang digunakan adalah robot beroda dengan gerak diferensial.

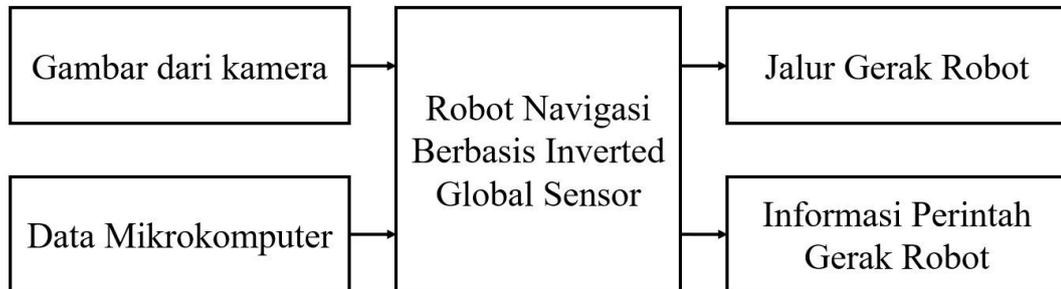
1.5. Metode Penelitian

Dalam pelaksanaan penelitian ini menggunakan langkah-langkah berikut:

1. Studi Literatur: Mengumpulkan dan mempelajari literatur-literatur yang dapat menjadi referensi tugas akhir.
2. Perancangan Sistem: Dilakukan perancangan berdasarkan pemahaman yang didapat pada studi literatur yang akan diimplementasikan pada perangkat keras.
3. Implementasi: Pada tahap ini alat yang sudah dirancang akan diimplementasikan pada lokasi yang telah ditentukan.
4. Pengujian Sistem dan Analisis Perangkat Keras: Alat yang telah diimplementasikan akan diuji berdasarkan parameter-parameter yang telah ditentukan dan kemudian hasil yang diperoleh akan dianalisis.
5. Penyusunan Buku: Penyusunan buku merupakan tahap yang dilakukan seiring dengan penerapan hasil perancangan, realisasi, pengujian, evaluasi serta analisis Tugas Akhir. Tugas Akhir.

BAB II TINJAUAN PUSTAKA

2.1. Prinsip Kerja



Gambar 2.1 Diagram fungsi

Dalam mencapai tujuan diagram fungsi di atas dari sistem robot navigasi, tahapan prinsip kerja konsep sebagai berikut:

1. Kamera akan menangkap gambar dari area yang telah ditentukan.
2. Gambar yang ditangkap oleh kamera akan diproses untuk memberikan lokalisasi objek yang terdeteksi.
3. Data yang telah diproses menghasilkan jalur gerak robot untuk memungut semua sampah yang terdeteksi hingga kembali ke homebase dan juga informasi perintah gerak robot.

2.2. Sistem Navigasi Robot dalam Ruangan

Dalam robotika, sistem navigasi ini merupakan salah satu hal yang penting. Navigasi berguna untuk memandu robot bergerak dari posisi *start* ke posisi tujuan (*goal*). Dalam pergerakannya, robot harus mampu menghindari halangan dengan waktu tempuh sesingkat mungkin [1]. Pembuatan sistem navigasi dapat dibuat dengan berbagai macam metode salah satunya yaitu metode *computer vision*.

Computer vision adalah metode untuk membuat robot dapat mengetahui lingkungan di sekitarnya menggunakan citra digital [2], metode *computer vision* dapat digunakan di luar maupun di dalam ruangan. Dalam kesempatan ini *computer vision* akan dipakai di dalam ruangan. Agar sistem dapat berjalan, maka perlu adanya *indoor localization*.

Indoor localization system adalah sebuah teknologi yang hampir sama penggunaannya dengan GPS hanya saja teknologi ini tidak menggunakan sistem komunikasi satelit dikarenakan sebuah sinyal internet atau satelit tidak bisa digunakan dengan baik di dalam gedung dan dapat menimbulkan kesalahan informasi jika tetap digunakan [3]. Teknologi ini berfungsi untuk mengetahui lokasi suatu objek di dalam ruangan. Infrastruktur nirkabel sangat dibutuhkan pada teknologi ini yang berguna untuk pengendalian robot. Karena lemahnya tingkat akurasi dari infrastruktur nirkabel maka perlu adanya algoritma untuk meningkatkan akurasi pada teknologi ini.

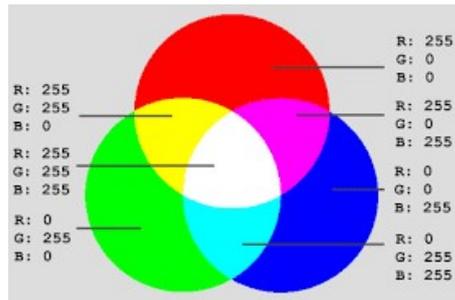
2.3. Pengolahan Citra Digital (*Image Processing*)

Citra atau *image* dapat didefinisikan sebagai sebuah fungsi dua dimensi, $f(x, y)$, di mana x dan y adalah koordinat (x, y) disebut intensitas atau level keabuan (*grey level*) dari gambar titik itu. Jika x, y dan f semuanya berhingga (*finite*), dan nilainya diskrit, maka gambarnya disebut citra digital. Sebuah citra digital terdiri dari sejumlah elemen yang berhingga, di mana masing-masing mempunyai lokasi dan nilai tertentu [4].

Citra harus diubah menjadi nilai-nilai *diskrit* agar bisa diolah di komputer dengan cara digitalisasi citra. Karena *output* dari kebanyakan sensor berupa gambar kontinu maka perlu dilakukan representasi citra menjadi fungsi *diskrit*.

2.3.1. Citra RGB

Citra RGB (*red, green, blue*) merupakan citra digital yang terdiri dari 3 warna dominan yaitu merah, hijau dan biru. Dengan 3 warna dasar tersebut dapat menghasilkan banyak warna pada komputer. Satu piksel pada citra direpresentasikan dengan 24-bit, 8-bit untuk merah, 8-bit untuk hijau, 8-bit untuk biru. Masing-masing warna memiliki intensitas berbeda dengan jumlah 8-bit (0-255), sehingga ruang warna RGB memiliki jumlah 24-bit. Jika masing-masing warna memiliki *range* 0-255, maka totalnya adalah $255^3 = 16.581.375$ variasi warna berbeda pada gambar, variasi warna ini cukup untuk berbagai gambar yang ditampilkan [5].

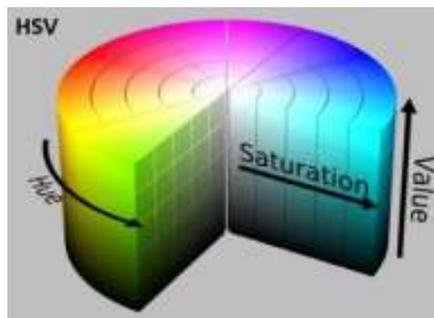


Gambar 2.2 Jumlah bit citra dalam RGB

2.3.2. Citra HSV (*Hue, Saturation, Value*)

HSV adalah sistem koordinat-silinder yang paling umum merepresentasikan poin dalam model warna RGB, yang mengatur ulang geometri RGB dalam upaya untuk perseptual yang lebih relevan daripada representasi koordinat kartesian [6].

Umumnya gambar yang ditampilkan dalam bentuk RGB akan diubah terlebih dahulu ke model HSV untuk memudahkannya dalam proses pengolahan citra. Model warna HSV ini menggambarkan warna yang sama dengan intuisi manusia untuk melihat warna, HSV juga mendeskripsikan warna dengan menggunakan perbandingan yang lebih mudah. Dalam model HSV ini terdapat 3 bagian dimana *hue* ini mewakili jenis warna, *saturation* mewakili tingkat jenuh atau pudarnya warna, dan *value* mewakili tingkat pencahayaan atau penerangan.



Gambar 2.3 Model Warna HSV

2.4. Arah Gerak Robot

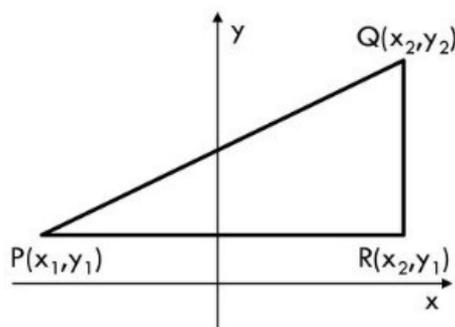
Robot adalah sebuah alat mekanik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dahulu [7]. Istilah robot berasal dari bahasa Ceko “*robota*” yang berarti pekerja atau kuli yang tidak mengenal lelah atau bosan. Robot merupakan sebuah alat mekanik yang dapat memperoleh informasi dari lingkungan (melalui sensor), dapat diprogram, dapat melaksanakan beberapa tugas yang berbeda, bekerja secara otomatis, dan cerdas (kecerdasan buatan). Robot biasa digunakan untuk tugas yang berat, berbahaya, pekerjaan berulang, pembersih limbah beracun, penjelajah bawah air dan luar angkasa, pertambangan, pekerjaan “cari dan tolong” (*search and rescue*), untuk pencarian tambang, di bidang hiburan, sampai alat bantu rumah tangga [8].

Untuk dapat robot bergerak dibutuhkan informasi berupa arah gerak yang berupa jarak tempuh, arah putar, dan sudut putar robot.

2.4.1. Teorema *Pythagoras*

Teorema *pythagoras* adalah suatu aturan matematika yang dapat digunakan untuk menentukan panjang salah satu sisi dari sebuah segitiga siku-siku. Teorema ini hanya berlaku untuk segitiga siku-siku saja, tidak bisa digunakan untuk menentukan sisi dari sebuah segitiga lain [9]. Terdapat dua syarat yang harus dipenuhi agar teorema *pythagoras* dapat berlaku yaitu hanya berlaku untuk segitiga siku-siku dan 2 sisi dalam segitiga tersebut telah diketahui panjangnya.

Teorema ini juga berfungsi untuk mencari jarak antara dua titik. Kedua titik tersebut harus diketahui lokasinya dalam bentuk titik koordinat. Dari kedua titik koordinat kita dapat mengilustrasikan gambar segitiga siku-siku yang terbentuk pada titik tersebut.



Gambar 2.4 Ilustrasi teorema *pythagoras*

Gambar 2.4 memudahkan kita untuk mengetahui cara mencari jarak antara kedua titik koordinat. Dengan menggunakan persamaan di bawah ini kita dapat mengetahui jaraknya.

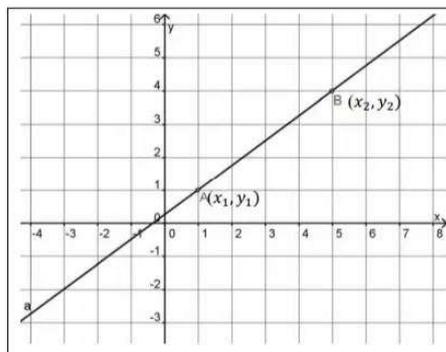
$$d(P, Q) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.1)$$

di mana:

- $d(P, Q)$ = Jarak titik P dengan titik Q
- x_1 = Koordinat sumbu x titik P
- x_2 = Koordinat sumbu x titik Q
- y_1 = Koordinat sumbu y titik P
- y_2 = Koordinat sumbu y titik Q

2.4.2. Persamaan Garis Lurus

Persamaan garis lurus adalah suatu persamaan yang jika digambarkan ke dalam bidang koordinat cartesius akan membentuk sebuah garis lurus [10]. Garis lurus yang terbentuk dapat dilalui oleh beberapa titik koordinat.



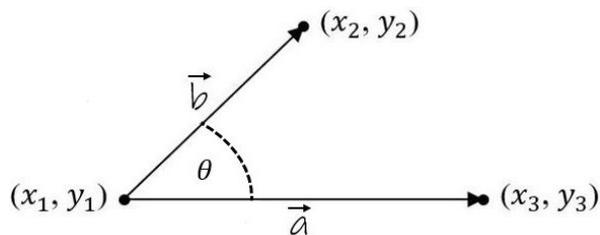
Gambar 2.5 Ilustrasi persamaan garis lurus

Pada Gambar 2.5 dapat dilihat terdapat kemiringan pada garis lurus yang tercipta. Dengan memanfaatkan persamaan garis lurus yang melewati dua titik koordinat kita dapat mencari arah belok/putar robot.

$$\frac{y-y_1}{y_2-y_1} = \frac{x-x_1}{x_2-x_1} \quad (2.2)$$

2.4.3. Sudut Vektor

Besaran-besaran yang memiliki sifat seperti pergeseran disebut vektor [11]. Jadi vektor adalah besaran yang memiliki suatu nilai dan arah. Jika diimplementasikan dalam koordinat kartesian, vektor dapat diuraikan ke arah sumbu x, sumbu y, dan sumbu z jika 3 dimensi. Dengan adanya 3 titik koordinat pada koordinat kartesian 2 dimensi kita dapat mengetahui besaran sudut yang terbentuk terhadap salah 1 titik tersebut.



Gambar 2.6 Ilustrasi sudut vektor

Dari Gambar 2.6 dapat diketahui bahwa sudut yang terbentuk dapat dicari besarnya dengan persamaan rumus sebagai berikut:

$$\cos \theta = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \cdot |\vec{b}|} \quad (2.3)$$

2.5. *Bubble Sort*

Algoritma *bubble sort* merupakan satu dari beberapa jenis algoritma untuk proses pengurutan data. Jenis algoritma ini termasuk ke dalam algoritma *comparison sort*, karena pada prosesnya dilakukan perbandingan, antara setiap elemen-elemen *array* yang disediakan [12]. Cara kerja dari algoritma ini adalah dengan menganalisa elemen pada *array* dari kiri ke kanan. Kemudian setiap elemen yang posisinya berdekatan nilainya akan dibandingkan, jika elemen di kiri lebih

besar dari kanan maka posisinya akan ditukar. Prosesnya akan terus berulang sampai setiap elemen telah sesuai urutannya [13].

Jika terdapat data *array* dengan nilai 6, 2, 4, dan 1, proses yang terjadi apabila menggunakan algoritma *bubble sort* dapat digambarkan sebagai berikut:

| | | | | | | | | | |
|-----------------------|---|---|---|---------|---------------------|---|---|---|---------|
| Proses Pertama | | | | | Proses Kedua | | | | |
| 6 | 2 | 4 | 1 | Ditukar | 2 | 4 | 1 | 6 | Tetap |
| 2 | 6 | 4 | 1 | Ditukar | 2 | 4 | 1 | 6 | Ditukar |
| 2 | 4 | 6 | 1 | Ditukar | 2 | 1 | 4 | 6 | Hasil |
| 2 | 4 | 1 | 6 | Hasil | | | | | |

| | | | | |
|----------------------|---|---|---|---------|
| Proses Ketiga | | | | |
| 2 | 1 | 4 | 6 | Ditukar |
| 1 | 2 | 4 | 6 | Hasil |

Gambar 2.7 Proses algoritma *bubble sort*

Dari gambar 2.7, dapat diketahui pada proses pertama algoritma ini mulai membandingkan elemen pertama di sebelah kiri dengan elemen di sebelah kanannya, yang diketahui elemen pertama memiliki nilai lebih besar dari elemen di sebelah kanannya, kemudian masing-masing elemen tersebut ditukar posisinya. Proses ini dilakukan hingga masing-masing elemen di sebelah kirinya memiliki nilai yang lebih kecil dari elemen di sebelah kanannya. Di tahap akhir akan mendapatkan bahwa nilai yang terkecil yang ditunjukkan pada elemen pertama telah berubah dengan nilai 1 dan nilai terbesar yang ditunjukkan pada elemen terakhir dengan nilai 6.