

## DAFTAR TABEL

Tabel III- 1. Aturan logika <i>fuzzy</i> .....	33
Tabel IV-1. Pengujian output driver motor kanan .....	47
Tabel IV-2. Pengujian output driver motor kanan .....	48
Tabel IV-3. Pengujian output PWM motor kanan .....	50
Tabel IV-4. Pengujian output PWM motor kiri .....	51

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan teknologi di bidang robotika pada saat ini mulai berkembang pesat dan menjadi ketertarikan. Hal ini karena peran robot yang sudah dapat menggantikan pekerjaan manusia. Lapangan pekerjaan manusia hampir sebagian sudah digantikan oleh peran robot. Karena yang namanya manusia pasti pernah melakukan kelalaian pada pekerjaannya dan membuat terjadinya kecelakaan kerja. Jika sebagian pekerjaan manusia bisa digantikan dengan sistem otomatisasi, maka akan mengurangi kesalahan pada pekerjaan, waktu akan lebih efisien dan dapat mengurangi terjadinya kecelakaan kerja. Salah satu teknologi di bidang robot yang sudah dibuat saat ini yaitu *Automated Guided Vehicle* (AGV).

AGV adalah sebuah robot *mobile* yang mengikuti sebuah petunjuk atau garis yang diberikan di lantai, atau menggunakan *vision* atau laser untuk bergerak ke arah tertentu [4]. AGV memerlukan sebuah sensor untuk dapat bergerak dengan otomatis tanpa operator. *Vision* adalah pilihan yang baik untuk sensor robot karena cukup fleksibel untuk mendeteksi atau mengenali fitur apa pun dengan warna dan ukuran apa pun [2]. Sistem *vision* atau kombinasi sistem penglihatan dengan sensor telah digunakan di banyak sistem lokalisasi dan navigasi [3].

Dalam penelitian Sulistyono (2009), navigasi robot menggunakan algoritma *fuzzy*. Proses *fuzzy* yang dilakukan meliputi fuzzifikasi, evaluasi *rule* dan defuzzifikasi. Dengan menggunakan algoritma ini robot dapat bergerak mengikuti jalur [1]. Logika *fuzzy* adalah peningkatan dari logika boolean yang mengenalkan konsep kebenaran sebagian. Di mana logika klasik menyatakan bahwa segala hal dapat diekspresikan dalam istilah biner (1 atau 0, hitam atau putih, ya atau tidak), logika *fuzzy* menggantikan logika boolean dengan tingkat kebenaran.

Permasalahan yang dihadapi yaitu semakin berkembangnya teknologi pada robot otomasi industri dan ingin menggantikan teknologi lama menjadi teknologi baru, yaitu jika dahulu AGV dapat mengikuti jalur warna hitam maka pada kesempatan kali ini penulis akan merancang teknologi terbaru. Penulis akan melakukan perancangan robot otomasi industri dengan menggunakan *vision*

*sensor* atau kamera untuk menangkap objek garis berwarna. Oleh sebab itu pada tugas akhir ini akan mencoba untuk mengimplementasikan *vision sensor* pada AGV.

## 1.2 Rumusan Masalah

Dari permasalahan di atas, dapat disimpulkan beberapa rumusan masalah pada penelitian ini adalah sebagai berikut :

1. Bagaimana penerapan *vision sensor* dapat menangkap objek jalur warna (merah, kuning, hijau dan hitam) di lantai yang efektif sebagai sistem navigasi untuk AGV ?
2. Bagaimana penerapan *vision sensor* pada sistem kendali AGV untuk mengatur gerak AGV ?
3. Bagaimana performansi AGV menggunakan *vision sensor* dengan kontrol logika fuzzy ?

## 1.3 Tujuan

Tujuan dari Tugas Akhir ini adalah sebagai berikut :

1. *Vision sensor* dapat menangkap objek jalur warna (merah, kuning, hijau dan hitam) dengan lebar maksimal 15cm. AGV dapat terintegrasi dengan *vision sensor* dan kontrol logika *fuzzy* sehingga dapat berjalan otomatis mengikuti jalur warna (merah, kuning, hijau dan hitam) tanpa operator dengan simpangan terbesar 2,4 cm setiap jalur warna.
2. Sinyal atau data yang dikirim oleh *vision sensor* dapat diterima oleh kendali AGV berupa nilai pergeseran jalur warna yang dideteksi. Sinyal atau data yang dapat diterima yaitu dengan kecepatan aliran data 9600 *bit-per-second* dan dengan jeda waktu inisialisasi pengiriman data 3 detik. Kendali AGV akan mengelompokkan nilai pergeseran yang diterima, jika nilai pergeseran negatif untuk belok kiri, nilai pergeseran positif untuk belok kanan dan nilai pergeseran sama dengan 0 untuk lurus.

3. Menerapkan *vision sensor* dengan kontrol logika fuzzy supaya AGV dapat mengikuti jalur warna dengan baik dan juga pergerakan lurus, belok kanan, belok kiri dapat sesuai dengan yang diinginkan.

#### 1.4 Batasan Masalah

Untuk membatasi cakupan pembahasan masalah pada Tugas Akhir ini, maka diberikan batasan-batasan sebagai berikut :

1. Tidak membahas tentang perancangan pada sensor *vision*, tetapi membahas tentang sistem navigasi dan kontrol gerak AGV menggunakan *vision sensor* dan kendali *fuzzy logic*.
2. Tidak membahas ketika AGV membawa beban.
3. Bahasa pemrograman yang digunakan untuk kendali *fuzzy logic* adalah Bahasa C.
4. *Library* pemrograman *fuzzy logic* yang digunakan adalah *ArduinoIDE*.
5. Warna garis yang di deteksi sensor *vision* adalah merah, kuning, hijau dan hitam.
6. Bahasa pemrograman yang digunakan untuk pengolahan citra adalah *Phyton* versi 3.8.1.
7. *Library* pemrograman *computer vision* yang digunakan adalah *OpenCV* versi 4.1.2.
8. Menggunakan *main board* yaitu *Personal Computer* sebagai *master* dan *Arduino Uno* sebagai *slave* untuk komunikasi antar pengolahan citra dengan kendali *fuzzy logic*.
9. Percobaan, pengujian dan pengambilan data dilakukan pada lantai ruangan Laboratorium INACOS N315 dengan luas ruangan yaitu 49,40 m<sup>2</sup>.
10. Lintasan yang dilalui harus terdeteksi dengan kamera sebagai *vision sensor*.
11. Intensitas cahaya harus cukup, tidak bisa menggunakan bantuan cahaya lampu ruangan, harus cahaya matahari.

## 1.5 Metode Penelitian

Metode penelitian yang digunakan pada Tugas Akhir ini adalah sebagai berikut :

### 1. Studi Literatur

Pada tugas akhir ini, studi literatur dilakukan dengan mempelajari teori dasar *fuzzy logic* atau logika fuzzy untuk kontrol pergerakan AGV dan mempelajari beberapa teori mengenai *sensor vision* yang dipakai untuk mendapatkan data yang dibutuhkan untuk kontrol pergerakan AGV.

### 2. Analisis Masalah

Menganalisis masalah pada sistem kendali *fuzzy logic* dan komunikasi serial yang digunakan. Mencari solusi agar pengiriman data yang dilakukan dapat akurat dan meminimalkan eror yang terjadi.

### 3. Perancangan dan Realisasi Perangkat Keras

Melakukan perancangan diagram blok sistem *fuzzy logic* dan komunikasi serial berdasarkan data yang diperoleh dengan memanfaatkan hasil studi literatur dan analisis masalah.

### 4. Pengujian Perangkat Keras

Setelah perancangan dan realisasi diselesaikan, tahap selanjutnya yaitu melakukan pengujian pada perangkat keras dan perangkat lunak untuk mendapatkan data yang diperlukan dalam Tugas Akhir yang dibuat dan mengetahui kinerja dari sistem tersebut.

### 5. Evaluasi dan Analisis Hasil

Setelah pengujian selesai, selanjutnya mengevaluasi hasil kinerja dari sistem. Apakah dari sistem tersebut perlu dilakukan perbaikan atau tidak dan menganalisis hasil dari kinerja sistem tersebut kemudian menyimpulkan penelitian yang dilakukan pada Tugas Akhir.

### 6. Penyusunan Buku

Penyusunan buku merupakan tahap yang dilakukan seiring dengan penerapan hasil perancangan, realisasi, pengujian, evaluasi serta analisis Tugas Akhir.

## 1.6 Sistematika Penulisan

Penulisan pada Tugas Akhir ini mengacu pada aturan Kamus Besar Bahasa Indonesia. Sistematika penulisan yang digunakan dalam Tugas Akhir ini adalah sebagai berikut :

### BAB I PENDAHULUAN

Pada bab ini akan membahas tentang latar belakang, rumusan masalah, tujuan, batasan masalah, metode penelitian dan sistematika penulisan Tugas Akhir.

### BAB II TINJAUAN PUSTAKA

Pada bab ini akan membahas tentang prinsip kerja konsep dari sistem yang dibuat dan dasar teori dalam merancang dan mengimplementasikan sistem dalam pembuatan Tugas Akhir.

### BAB III PERANCANGAN SISTEM

Pada bab ini akan membahas tentang sistem deteksi garis warna, perancangan sistem kontrol pergerakan, perancangan perangkat keras dan perancangan perangkat lunak.

### BAB IV PENGUJIAN DAN ANALISA DATA

Pada bab ini diuraikan mengenai pengujian sudut yang didapat oleh *vision sensor* dan pengujian kontrol pergerakan pada AGV yang telah dirancang dan analisa data yang didapat dari pengujian.

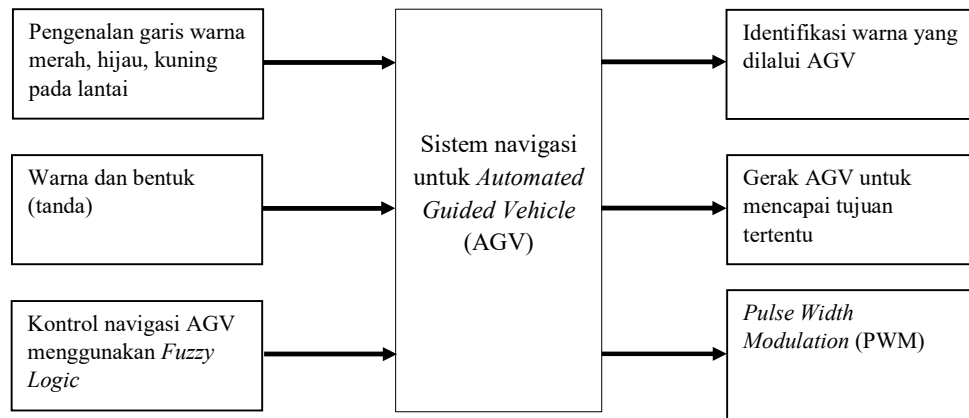
### BAB V PENUTUP

Pada bab ini berisikan kesimpulan dari keseluruhan pembahasan pada tugas akhir dan saran untuk pengembangan selanjutnya.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Prinsip Kerja Konsep



Gambar II-1. Diagram fungsi sistem

Dalam mencapai tujuan diagram fungsi di atas dari Penerapan Sensor Vision pada Sistem Navigasi Berbasis *Fuzzy Logic* untuk *Automated Guided Vehicle* (AGV), tahapan prinsip kerja konsep sebagai berikut :

1. Sensor melakukan kalibrasi untuk pengenalan garis warna (merah, kuning, hijau dan hitam) dan bentuk (tanda) pada lantai.
2. Warna di proses melalui sistem navigasi yang dibuat, ada empat kemungkinan tujuan sesuai dengan warna yang sudah ditentukan.
  - a. Jika warna merah yang diproses sistem, maka AGV menuju ruangan 1.
  - b. Jika warna kuning yang diproses sistem, maka AGV menuju ruangan 2.
  - c. Jika warna hijau yang diproses sistem, maka AGV menuju ruangan 3.
  - d. Jika warna hitam yang diproses, maka AGV akan menuju ruangan 4
3. Bentuk di proses melalui sistem navigasi yang dibuat, ada dua kemungkinan pergerakan AGV.

- a. Jika bentuk garis lurus, maka AGV berjalan lurus mengikuti bentuknya.
  - b. Jika bentuk garis melengkung, maka AGV berjalan berbelok, belok kanan ataupun belok kiri.
4. Kontrol navigasi AGV menggunakan kendali *fuzzy logic*.
- a. Jika sinyal atau data yang diterima berupa nilai pergeseran negatif, maka arah gerak AGV belok kiri.
  - b. Jika sinyal atau data yang diterima berupa nilai pergeseran positif, maka arah gerak AGV belok kanan.
  - c. Jika sinyal atau data yang diterima berupa nilai pergeseran sama dengan 0, maka arah gerak AGV lurus.
  - d. Ketiga kemungkinan di atas akan menghasilkan suatu nilai *Pulse Width Modulation* (PWM) menyesuaikan besaran sudut yang diterima oleh kendali AGV.

## **2.2 Pengertian Pengolahan Citra / *Image Processing***

Citra atau *Image* merupakan istilah lain dari gambar, yang merupakan informasi berbentuk visual. Suatu citra diperoleh dari penangkapan kekuatan sinar yang dipantulkan oleh objek. Suatu citra dapat didefinisikan sebagai fungsi intensitas cahaya dua dimensi  $f(x, y)$ , di mana  $x$  dan  $y$  adalah koordinat spasial dan  $f$  pada titik  $(x, y)$  merupakan tingkat kecerahan suatu citra pada suatu titik [6].

### **2.2.1 Pengolahan Citra Digital**

Citra digital merupakan citra  $f(x, y)$  yang dikonversi menjadi diskrit melalui proses sampling dan kuantisasi. Sampling menyatakan besar ukuran *pixel* (*picture element*) pada citra sedangkan kuantisasi menyatakan besarnya tingkat kecerahan. Pengolahan citra digital adalah sebuah ilmu yang mempelajari tentang teknik mengolah suatu citra. Citra yang dimaksud adalah citra diam (foto) maupun citra bergerak (berasal dari *webcam* yang dipakai) [3].



Citra merupakan fungsi kontinyu dengan intensitas cahaya pada bidang dua dimensi [6]. Supaya citra dapat diolah komputer, maka citra harus diubah menjadi nilai-nilai diskrit, yaitu dengan cara digitalisasi citra. Digitalisasi citra merupakan perubahan dari fungsi kontinyu menjadi nilai-nilai diskrit. Di bawah adalah sebuah matriks yang digunakan dalam merepresentasikan citra matriks.

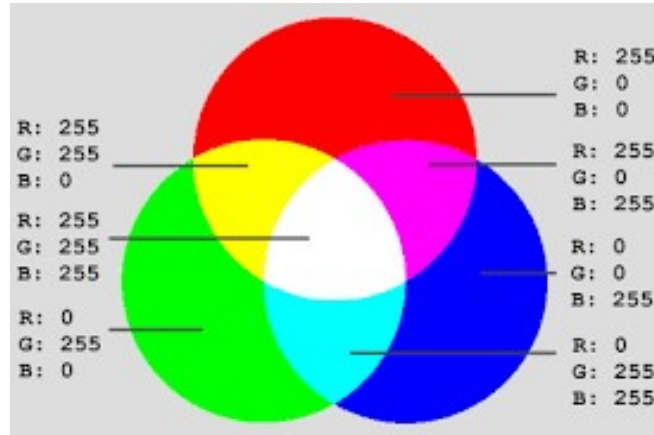
$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \dots & f(0, N-1) \\ f(1, 0) & f(1, 1) & \dots & f(1, N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1, 0) & f(M-1, 1) & \dots & f(M-1, N-1) \end{bmatrix} \quad (2.1)$$

### 2.2.2 Citra Bergerak

Citra diam adalah citra tunggal yang tidak bergerak atau biasa disebut foto. Sedangkan citra bergerak adalah rangkaian citra diam ditampilkan berurutan yang menghasilkan kesan pada mata sebagai citra diam yang bergerak. Setiap citra di dalam rangkaian itu disebut *frame*. Citra bergerak dapat disebut sebagai video yang memiliki satuan *Frame Per Second* (FPS). *Frame Per Second* adalah banyaknya citra yang bergerak setiap detiknya, semakin banyak citra/*frame* yang dihasilkan setiap detiknya maka semakin baik kualitas video yang dihasilkan dan begitu sebaliknya semakin sedikit citra/*frame* yang dihasilkan setiap detiknya maka kualitas video yang dihasilkan kurang memuaskan [2].

### 2.2.3 Citra RGB

Citra digital RGB adalah citra yang memiliki 3 warna dominan, yaitu merah (*red*), hijau (*green*), biru (*blue*). Setiap warna memiliki intensitas berbeda dengan jumlah 8-bit (0-255), sehingga ruang warna RGB memiliki jumlah 24-bit [6]. Jika setiap warna memiliki *range* 0-255, maka totalnya adalah  $255^3 = 16.581.375$  (16 K) variasi warna berbeda pada gambar, variasi warna ini cukup untuk berbagai gambar yang ditampilkan [6]. Pada gambar II-2. merupakan gambar dan jumlah bit yang ditampilkan pada citra warna dalam RGB.



Gambar II-2. Jumlah bit citra warna dalam RGB

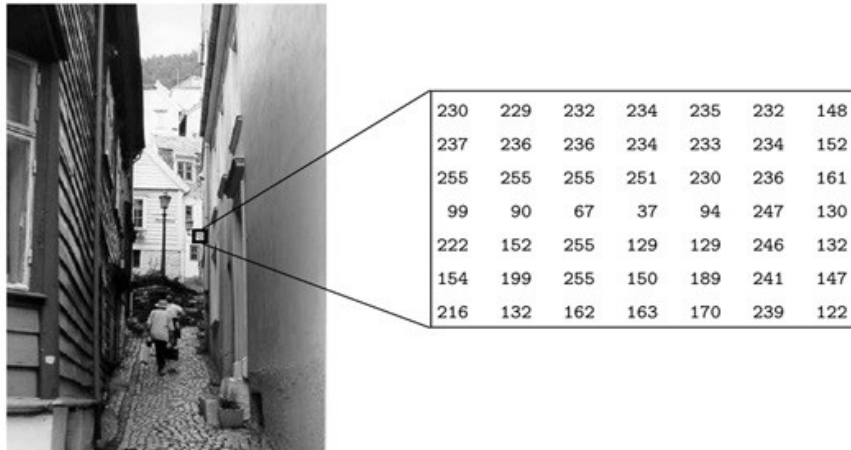
### 2.2.4 Citra *Grayscale*

Citra digital *black and white (grayscale)* setiap piksel mempunyai warna gradasi dari putih sampai hitam. Rentang warna gradasi tersebut berarti bahwa setiap piksel dapat diwakili oleh 8 bit, atau 1 *byte*. Rentang warna pada *black and white* cocok digunakan sebagai pengolahan file gambar. Salah satu bentuk fungsinya digunakan dalam bidang kedokteran (*x-ray*) [10]. *Black and white* merupakan hasil rata-rata dari *color image*, dengan demikian maka persamaannya dituliskan sebagai berikut [10].

$$I_{BW}(x, y) = \frac{I_R(x, y) + I_G(x, y) + I_B(x, y)}{3} \quad (2.2)$$

Keterangan :

- $I_R(x, y)$  = nilai piksel *Red* titik (x,y)
- $I_G(x, y)$  = nilai piksel *Green* titik (x,y)
- $I_B(x, y)$  = nilai piksel *Blue* titik (x,y)
- $I_{BW}(x, y)$  = nilai piksel *Black and White* titik (x,y)



Gambar II-3. Ilustrasi citra *grayscale*

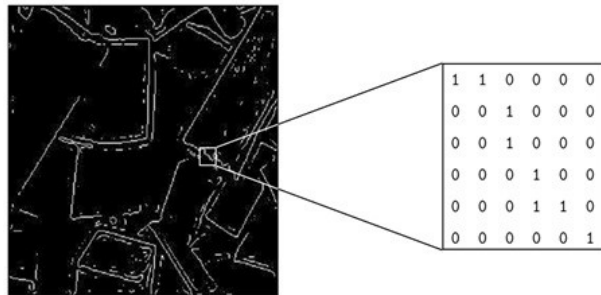
### 2.2.5 Citra *Monochrome*

Setiap piksel hanya terdiri dari warna hitam dan atau putih, karena hanya terdapat dua warna untuk setiap piksel, maka hanya perlu 1 bit per piksel (0 dan 1) atau apabila dalam 8 bit (0 dan 255), menjadikan efisien dalam hal penyimpanan. Citra *monochrome* merupakan hasil pengolahan dari black and white image, dengan menggunakan fungsi sebagai berikut [10].

$$I_{Bin}(x,y) = \begin{cases} 0 & I_{BW}(x,y) < T \\ 255 & I_{BW}(x,y) \geq T \end{cases} \quad (2.3)$$

dan dalam bentuk *floating point*

$$I_{Bin}(x,y) = \begin{cases} 0 & I_{BW}(x,y) < T \\ 1 & I_{BW}(x,y) \geq T \end{cases} \quad (2.4)$$



Gambar II-4. Ilustrasi citra *monochrome*

## 2.3 Robotika

Robotika adalah ilmu yang mempelajari tentang proses perancangan, pembuatan dan pengembangan robot serta membahas mengenai penerapan teknologi robotika pada kehidupan manusia. Dalam penerapannya, ilmu robotika erat hubungannya dengan ilmu kecerdasan buatan [11].

Robot secara garis besar disusun oleh 3 komponen utama, yaitu sensor, aktuator, dan mikrokontroler. Aktuator adalah komponen yang digunakan untuk menggerakkan robot. Pergerakan ini bersifat stasioner, seperti pergerakan sendi pada robot berbentuk tangan/lengan atau pergerakan yang bersifat *mobile*, seperti pergerakan sebuah robot beroda berpindah tempat. Sensor adalah komponen yang digunakan untuk mendapatkan masukan yang berupa nilai/variabel dari sekitar robot [5]. Analogi sensor pada makhluk hidup adalah indra yang digunakan untuk merasakan lingkungan sekitar. Tipe masukan yang bisa diterima bergantung pada kemampuan sensor tersebut untuk menerima tipe masukan yang diinginkan [5].

## 2.4 Automated Guided Vehicle (AGV)

Automated Guided Vehicle (AGV) adalah sebuah *mobile robot* yang dapat bergerak otomatis mengikuti petunjuk garis yang terdapat di lantai, atau menggunakan *vision* atau laser untuk bergerak otomatis. Untuk membantu navigasi AGV dapat digunakan beberapa jenis *steer control systems* [4]. Pada tugas akhir ini, *steer control system* yang digunakan yaitu *differential drive*. Roda bagian depan sebagai penggerak sekaligus penentu arah lurus, belok kanan ataupun belok kiri.

### 2.4.1 Navigasi Automated Guided Vehicle (AGV)

Sistem navigasi dirancang untuk diterapkan pada *autonomous mobile robot*. *Autonomous mobile robot* menggunakan sistem penggerak roda (*tracked*) dan menggunakan sistem pergerakan *differential drive* [4]. Sistem navigasi yang dirancang yaitu sistem navigasi berbasis posisi dengan metode *waypoint*. Sistem navigasi dirancang untuk pergerakan di luar ruangan, khususnya di wilayah darat dengan posisi dan rute yang telah ditentukan oleh operator (manusia). Sistem navigasi bekerja secara tunggal yang artinya tidak dilengkapi dengan sistem pendukung lainnya (contoh sistem pendukung adalah *obstacle avoidance system*). Sistem perangkat keras terutama modul-modul sensor dikondisikan

mendekati ideal sesuai anjuran *datasheet* sehingga dapat mengurangi pengaruh-pengaruh lingkungan yang mengganggu kinerja sistem [15].

#### **2.4.2 Metode Navigasi *Waypoint***

Metode navigasi *waypoint* memberikan informasi penting identifikasi titik dalam ruang fisik berupa koordinat. Sifat koordinat adalah satuan bujur dan lintang koordinat untuk sistem navigasi. Set koordinat informasi lintasan pada *autonomous mobile robot* untuk mengikuti rute yang benar untuk mencapai tujuan. Penggunaan metode ini akan lebih efisien untuk mencari koordinat pada *autonomous mobile robot* [15].

*Waypoint* adalah metode navigasi dengan titik acuan atau kumpulan koordinat yang digunakan untuk keperluan navigasi untuk mengidentifikasi sebuah titik di peta. Koordinat-koordinat itu biasanya menyertakan *longitude*, *latitude*, dan kadang *altitude* untuk keperluan navigasi di udara [15]. *Waypoint* digunakan untuk navigasi yang tidak memiliki jalur dan tidak tampak seperti navigasi di udara dan navigasi di laut, navigasi di darat kadang tidak memiliki jalur yang jelas. Khusus navigasi di darat yang tidak menggunakan manusia sebagai penentu arah melainkan robot, *waypoint* digunakan meski terdapat jalur yang jelas. Hal ini penting agar robot tetap memiliki rute.

*Waypoint* dibagi menjadi dua jenis, yaitu *waypoint fly by* dan *waypoint fly over* [15]. *Waypoint fly by* tidak melewati di atas lokasi *waypoint* namun tetap menuju ke arah tujuan, sedangkan *waypoint fly over* melewati di atas lokasi *waypoint*. Setelah satu *waypoint* terlewati, maka operator harus menetapkan *waypoint* berikutnya yang disebut dengan *waypoint* aktif.

## 2.5 Posisi Kamera

Posisi kamera pada sistem yang digunakan sangat berpengaruh pada hasil yang diproses. Terdapat dua macam peletakan kamera, yaitu *eye in hand* dan *stand alone*. Kragic dan Christensen (2011) mengemukakan bahwa peletakan unit kamera untuk sebuah sistem dapat secara *eye in hand* atau *stand alone* [17]. *Eye in hand* merupakan metode peletakan kamera pada bagian ujung dari robot atau bisa disebut terpasang pada bagian badan robot. Sedangkan *stand alone* merupakan metode peletakan kamera terpisah dari bagian badan robot atau bisa disebut tidak terpasang pada bagian badan robot, bisa diletakan pada sudut ruangan dan sebagainya. Marchand (2007) mengemukakan bahwa peletakan kamera sangat mempengaruhi gambar yang diperoleh dan pencahayaan dapat mempengaruhi gambar yang akan diproses [17]. Pada penelitian ini, sangat memperhatikan peletakan kamera dan pencahayaan, karena jika tidak sesuai maka gerak pada AGV tidak akan bagus.



Gambar II-5. Posisi kamera

## 2.6 Logika *Fuzzy*

Logika *fuzzy* adalah komponen pembentuk *soft computing*. Dasar logika *fuzzy* adalah teori himpunan *fuzzy*. Himpunan *fuzzy* adalah kelas objek dengan rangkaian nilai keanggotaan (Zadeh. 1965). Himpunan tersebut ditandai dengan fungsi keanggotaan yang diberikan kepada setiap objek dengan nilai berkisar antara nol dan satu. Notasi yang digunakan antara lain *inclusion*, *union*, *intersection*, *komplemen*, relasi, berbagai sifat dari notasi dalam konteks himpunan *fuzzy* juga diterapkan. Secara khusus, teori pemisah untuk himpunan *fuzzy* adalah memberikan pemisah tanpa harus benar-benar memisahkan himpunan *fuzzy* tersebut (Zadeh. 1965).

Menurut (Kosko, B. University of Sothern California) logika *fuzzy* dapat menghasilkan sesuatu yang mengagumkan, seperti :

1. Pengambilan keputusan, secara teori dapat membuat pertimbangan berdasarkan seluruh dokumen yang pernah di tulis.
2. Kendaraan cerdas dengan perangkat sonar dapat mengatur pengereman mendadak. Dengan *fuzzy navigator*, peta komputerisasi, serta perangkat *transmitter* dan *receiver* pada aspal, suatu kendaraan dapat mengendalikan dirinya sendiri.
3. Robot seperti manusia, yang dapat menirukan perilaku manusia.

Seperti halnya himpunan konvensional, ada beberapa operasi yang didefinisikan secara khusus untuk kombinasi dan memodifikasi himpunan *fuzzy*. Nilai keanggotaan sebagai hasil dari operasi 2 himpunan sering dikenal dengan nama *file strength* atau  $\alpha$ -predikat. Ada 3 operator dasar yang diciptakan oleh Dr. Lotfi Zadeh (1965).

A	B	$\min(A,B)$	A	B	$\max(A,B)$	A	$1 - A$
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

**AND**                      **OR**                      **NOT**

Gambar II-6. Operator Dasar

Langkah pertama untuk mendesain sistem *fuzzy logic* adalah mengambil nilai *crisp input*, dan menentukan fungsi derajat keanggotaan (*degree of membership*