PERANCANGAN DAN REALISASI SISTEM AUTOMATIC GUIDED VEHICLE (AGV) MENGGUNAKAN ALGORITMA DIJKSTRA DAN FUZZY LOGIC

DESIGN AND IMPLEMENTATION OF AUTOMATIC GUIDED VEHICLE (AGV) SYSTEM USING DIJKSTRA ALGORITHM FOR POSITION INFORMATION AND NAVIGATION

[1]Samuel Febrikab Dwiprasetiabudhi, [2]Angga Rusdinar, [3]Ramdhan Nugraha
[1,2,3]Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom
Jalan Telekomunikasi, Dayeuh Kolot Bandung 40257 Indonesia
[1]febrikab@students.telkomuniversity.ac.id, [2]angga.rusdinar@telkomuniversity.ac.id,
[3]ramdhan@telkomuniversity.ac.id

ABSTRAK

Pada sebuah industri, terutama industri manufaktur, pengiriman barang baik itu barang hasil produksi, barang hampir jadi, atau bahan baku merupakan bagian penting dari proses produksi. Untuk meningkatkan hasil produksi dibutuhkan sistem pengiriman yang lebih efisien dan efektif. Salah satu perangkat yang digunakan untuk pengiriman barang adalah *Automatic Guided Vehicle* (AGV), untuk meningkatkan hasil produksi yang lebih optimal maka dibutuhkan AGV yang dapat memilih jalan terpendek dari posisi awal AGV berada hingga tempat yang dituju. Pada Tugas Akhir ini akan dibahas mengenai perancangan dan realisasi AGV beserta algoritma Dijkstra untuk pemilihan jalur terpendek. Untuk navigasi AGV dalam mengikuti jalur yang berupa garis digunakan logika *fuzzy* sedangkan untuk mengetahui titik persimpangan dengan terminal yang dituju atau terminal posisi sekarang menggunakan *Radio Frequency Identification* (RFID). Berdasarkan hasil dari pengujian sistem, tingkat keberhasilan pergerakan AGV dari terminal satu ke terminal empat sebesar 100% sedangkan dari terminal adalah 60%
Kata Kunci : automatic guided vehicle, dijkstra, RFID, *fuzzy*

ABSTRAC

At industry, especially the manufacturing industry, both freight the goods produced, nearly finished goods or raw materials is an important part of the production process. To increase production takes delivery system more efficient and effective. One device used for the delivery of goods is Automatic Guided Vehicle (AGV), to increase the production yields more optimal AGV is required to choose the shortest path from the starting position until the AGV is the destination. In this final project will discuss the design and realization of Dijkstra's algorithm AGV along the shortest path selection. For AGV navigation in the form of a line following the path used while the fuzzy logic to determine the point of intersection with the destination terminal or terminal position now using Radio Frequency Identification (RFID). Based on the results of the testing system, the success rate of AGV movement from one terminal to terminal four at 100% while the terminal is 60%

Key Word : automatic guided vehicle, dijkstra, RFID, fuzzy

1. Pendahuluan

Perkembangan industri merupakan salah satu faktor yang diperhitungkan dalam pembangunan sebuah negara. Semakin modern teknologi yang digunakan oleh industri dapat menghasilkan produksi yang lebih efisien dan efektif. Hasil produksi yang optimal dapat ditingkatkan dengan beberapa cara, salah satunya adalah kemampuan kendaraan mengetahui jalur terpendek yang harus dilalui.

Automatic Guided Vehicle merupakan kendaraan terpadu yang digunakan dalam industri manufaktur dan berfungsi untuk distribusi bahan baku dan hasil produksi dari satu tempat ke tempat yang lain. Selain dalam industri manufaktur *Automatic Guided Vehicle* juga digunakan

pada rumah sakit, kantor pos, penjara, dan beberapa institusi yang membutuhkan sistem pengantaran otomatis. Sistem operasi *Automatic Guided Vehicle* tidak memerlukan operator untuk mengemudikan gerak *Automatic Guided Vehicle* secara langsung, untuk beberapa kegunaan *Automatic Guided Vehicle* sudah diprogram untuk bergerak menuju ke suatu tujuan dengan navigasi secara otomatis sehingga operator hanya bertugas untuk mengawasi dan mengendalikan *Automatic Guided Vehicle* dari jarak jauh.

Algoritma Dijkstra adalah algoritma yang digunakan untuk mencari jalur terpendek dalam sebuah perjalanan dengan mengadopsi sistem pencarian *greedy*, yaitu pencarian melalui hasil dari jumlah bobot terkecil dari satu titik ke titik lainnya. Pada *Automatic Guided Vehicle* algoritma tersebut dijalankan bersama dengan dengan program lain seperti *fuzzy logic* pada *line follower*, *Radio Frequency Identifier* (RFID) untuk informasi posisi, dan perintah dari *user* yang dikirim melalui jaringan *bluetooth*. Sehingga *Automatic Guided Vehicle* dapat beroperasi dalam lingkup yang lebih besar dan menghasilkan hasil produksi yang lebih optimal tanpa memiliki banyak pekerja.

2. Dasar Teori

2.1 *Automatic Guided Vehicle*

Automatic Guided Vehicle (AGV) adalah sebuah *mobile robot* yang mampu bergerak atau melakukan pekerjaan tertentu secara mandiri.[4] AGV dapat diprogram sehingga bekerja secara otomatis dan digunakan sebagai perangkat manipulasi. AGV memiliki pergerakan aktif karena dapat bergerak dan berpindah posisi dari satu titik ke titik yang lain secara dinamis, oleh karena itu AGV digunakan hampir di setiap industri manufaktur untuk memindahkan berbagai macam produk. AGV pada industri manufaktur digunakan untuk memindahkan bahan baku atau hasil produksi dari satu lokasi ke lokasi yang lain.

2.2 Algoritma Dijkstra

Algoritma dijsktra merupakan algoritma pencarian jalur terpendek dari verteks *a* ke verteks *z* dalam graf berbobot, bobot tersebut merupakan bilangan positif dan tidak boleh bernilai negarif, bobot merupakan panjang jalan yang akan dilalui.[2] Algoritma Dijkstra mencari lintasan terpendek dalam sejumlah langkah. Untuk setiap simpul sumber (*source*) dalam graf, algortima ini akan mencari jalur dengan *cost* minimum antara simpul tersebut dengan simpul lainnya. Algoritma ini juga dapat digunakan untuk mencari total biaya (*cost*) dari lintasan terpendek yang dibentuk dari sebuah simpul ke sebuah simpul tujuan. Sebagai contoh, bila simpul pada graf merepresentasikan kota dan bobot sisi merepresentasikan jarak antara 2 kota yang mengapitnya, maka algoritma dijkstra dapat digunakan untuk mencari rute terpendek antara sebuah kota dengan kota lainnya.

2.3 *Fuzzy Logic*

Logika *Fuzzy* merupakan logika dalam pengambilan keputusan yang digunakan untuk memecahkan masalah dengan sistem yang sulit untuk dimodelkan, teori tentang *fuzzy set* pertama kali diperkenalkan dan dikembangkan oleh Dr. Lotfi A. Zadeh dari Universitas California, Berkeley pada tahun 1965.[3] Dalam kehidupan banyak masalah dengan informasi yang sulit direpresentasikan ke dalam sebuah model rumus atau angka yang pasti karena informasi tersebut bersifat kualitatif (tidak bisa dihitung secara kuantitatif). Fuzzy logic dibagi menjadi tiga bagian proses, yaitu:

1. ***Fuzzyfication***

   Fuzzyfication merupakan proses pengubahan data masukan yang berupa nilai kebenaran bersifat pasti (*crisp* input) menjadi masukkan *fuzzy* yang berupa nilai linguistik dengan cara pemetaan crisp input pada himpunan *fuzzy*.

2. ***Inference***

   Suatu aturan *fuzzy* dituliskan sebagai: *IF antecendent THEN consequent.* Dalam suatu sistem berbasis aturan fuzzy, proses *inference* memperhitungkan semua aturan yang ada dalam basis pengetahuan. Hasil dari proses *inference* direpresentasikan oleh suatu *fuzzy set* untuk setiap untuk setiap *variable* bebas (pada *consequent*). Derajat keanggotaan untuk setiap nilai *variable* tidak bebas menyatakan ukuran kompatibilitas terhadap *variable* bebas (pada *antecendent*).

3. *Defuzzification*

Terdapat berbagai metode *defuzzification* yang dapat diaplikasikan untuk berbagai macam masalah. Metode *Weighted Average* mengambil nilai rata-rata dengan menggunakan pembobotan berupa derajat keanggotaan. Sehingga y* didefinisikan sebagai :

$$\frac{(\ )}{(\ )} \quad \text{.....(2.1)}$$

Dimana y adalah nilai crisp, μ(y) adalah derajat keanggotaan dari nilai crisp y. Kelemahan dari metode ini hanya bisa digunakan bila fungsi keanggotaan dari keluaran *fuzzy* memiliki bentuk yang sama.

2.4 *Radio Frequency Identification* (RFID)

RFID adalah proses identifikasi seseorang atau objek dengan menggunakan frekuensi transmisi radio.[1] RFID menggunakan frekuensi radio untuk membaca informasi dari sebuah divais yang bernama *tag* atau *transponder* (*Transmitter + Responder*). *Tag* RFID akan mengenali diri sendiri ketika mendeteksi sinyal dari divais yang disebut pembaca RFID (RFID *Reader*).

RFID adalah teknologi identifikasi yang fleksibel, mudah digunakan, dan sangat cocok untuk operasi otomatis.[1] RFID mengkombinasikan keunggulan yang tidak tersedia pada teknologi identifikasi yang lain. RFID tidak memerlukan kontak langsung maupun jalur cahaya untuk dapat beroperasi, dapat berfungsi pada berbagai variasi kondisi lingkungan, dan menyediakan tingkat integritas data yang tinggi. Sebagai tambahan, karena teknologi ini sulit untuk dipalsukan, maka RFID dapat menyediakan tingkat keamanan yang tinggi.

3. Perancangan Sistem

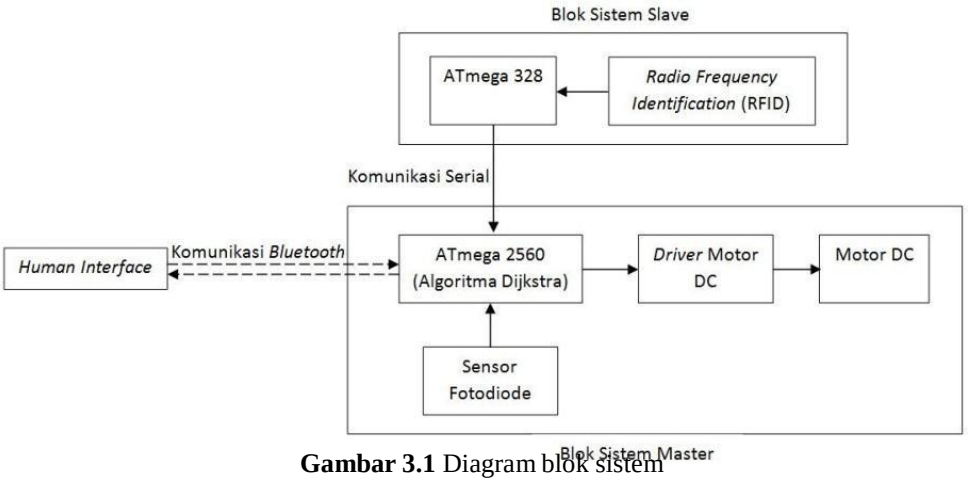
Perancangan sistem *Automatic Guided Vehicle* terdiri dari beberapa blok sistem yang terintegrasi menjadi satu sistem. Pembagian blok sistem dibagi menjadi blok *user input* dan sensor, blok pengolahan data, dan blok keluaran.

Pada bagian *user input*, masukan berasal dari *user* yang memasukkan data tujuan berdasarkan informasi data yang sudah dikirim oleh mikrokontroler menggunakan *serial monitor*. Sedangkan sensor menggunakan sensor fotodiode untuk membaca jalur yang berupa garis dan menggunakan modul RFID untuk informasi posisi dari percabangan serta membaca tujuan yang dituju, untuk pengiriman data antara user dengan mikrokontroler menggunakan komunikasi *wireless serial (bluetooth).*

Pada bagian pengolahan data menggunakan dua mikrokontroler, pembagian dalam pengolahan data dibedakan menjadi pembacaan sensor garis dengan logika untuk mengikuti garis, pembacaan module RFID serta data dari *user input* untuk diolah menggunakan logika jalur terpendek sehingga mengetahui apakah keluaran sistem sudah sesuai, komunikasi antar mikrokontroler menggunakan komunikasi I2C dan *serial*.

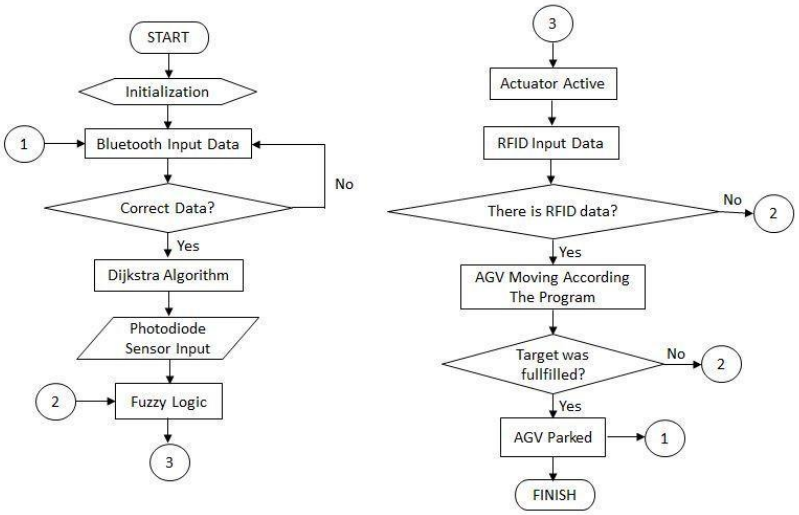
Pada bagian keluaran merupakan pengaturan arah pergerakan dan kecepatan dua motor DC yang dipasang secara diferensial dengan menggunakan *driver* mosfet. Data yang dihasilkan dari bagian pengolahan akan digunakan untuk mengatur arah gerak dari kedua motor serta mengatur kecepatannya menggunakan *duty cycle*.

Berikut adalah diagram blok dari sistem yang digunakan dalam pengerjaan tugas akhir :

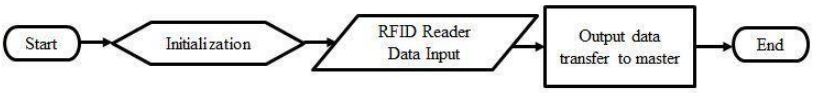


Gambar 3.1 Diagram blok sistem

Sedangkan untuk flowchart sistem digambarkan pada gambar 3.2 untuk *master* sistem dan gambar 3.3 untuk *slave* sistem.



Gambar 3.2 *Master* sistem *flowchart*



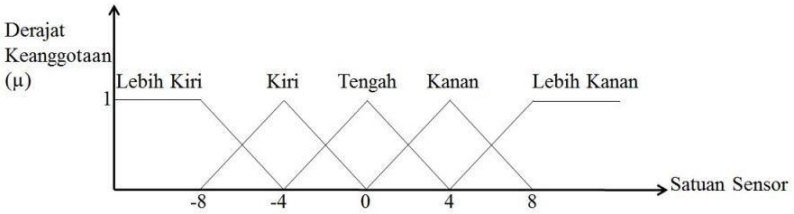
Gambar 3.3 *Slave* sistem *flowchart*

3.1 Perancangan *Fuzzy Logic*

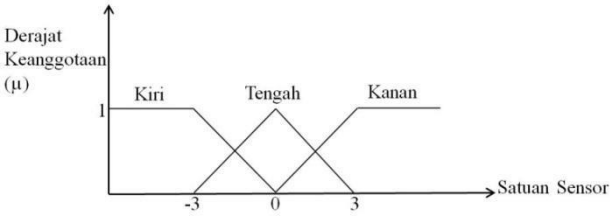
Sistem kendali yang digunakan untuk mengikuti garis pada AGV menggunakan *Fuzzy Logic*. Pada sistem ini memiliki dua baris sensor masukan berupa sensor garis yang berjumlah 16 buah (10 depan dan 6 belakang) dan dipasang secara sejajar. Hasil dari pembacaan sensor digunakan sebagai masukan untuk *fuzzy logic* dengan keluaran berupa kecepatan motor DC yang berupa nilai PWM, berikut akan dijelaskan proses-proses yang terjadi pada suatu sistem *fuzzy logic* yang digunakan pada robot.

3.1.1 *Fuzzyfication*

Masukan dari sensor garis ini memiliki delapan nilai linguistik yaitu lima untuk baris depan **LEBIH KIRI, KIRI, TENGAH, KANAN, dan LEBIH KANAN** dan tiga untuk baris belakang **KIRI, TENGAH, dan KANAN** dengan fungsi keanggotaan bahu trapesium dan segitiga. Gambar himpunan *fuzzy* dari *input* sistem pada gambar 3.4 dan gambar 3.5.



Gambar 3.4 Fungsi keanggotaan masukan sensor garis bagian depan

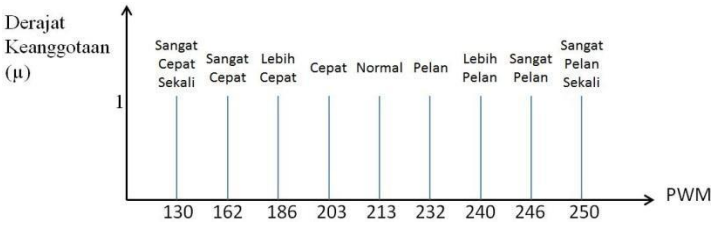


Gambar 3.5 Fungsi keanggotaan sensor garis bagian belakang

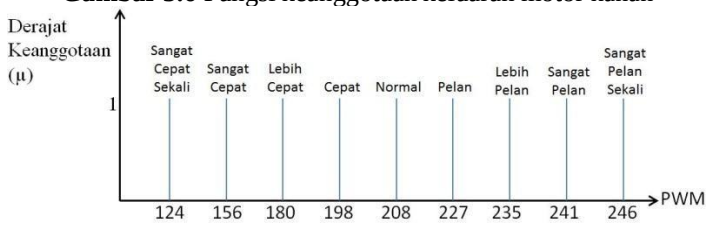
Pada keluaran sistem menggunakan model sugeno, pembentukan fungsi keluaran pada model sugeno memiliki fungsi yang lebih sederhana dengan respon lebih cepat dari model yang lain. Bentuk keluaran fungsi keanggotaan pada model sugeno mempunyai bentuk *singletone*,

bentuk dengan derajat keanggotaan satu pada suatu nilai *crisp* tunggal dan nilai nol pada suatu *crisp* yang lain.

Karena keluaran dalam bentuk *singletone* maka fungsi pada setiap nilai linguistik bernilai satu dan nol pada diluar nilai linguistik. Keluaran pada sistem yang dibuat ada dua, yaitu kecepatan motor dc kanan dan kiri. Untuk keluaran sistem yang berupa kecepatan memiliki sembilan nilai linguistik, yaitu: **SANGAT CEPAT SEKALI, SANGAT CEPAT, LEBIH CEPAT, CEPAT, NORMAL, PELAN, LEBIH PELAN, SANGAT PELAN, SANGAT PELAN SEKALI.** Gambar fungsi keanggotaan dari output *fuzzy* pada gambar 2.6 dan gambar 2.7.



Gambar 3.6 Fungsi keanggotaan keluaran motor kanan



Gambar 3.7 Fungsi keanggotaan keluaran motor kiri

3.1.2 *Rule Inference*

Pada *rule inference*, terjadi proses pengolahan data masukan fuzzyfikasi dengan hasil keluaran yang dikehendaki dengan aturan-aturan tertentu. Dari aturan-aturan yang dibentuk inilah yang nantinya akan menentukan respon dari sistem terhadap berbagai kondisi *set point* dan gangguan yang terjadi pada sistem yang akan dibuat. Rule inference sistem tertulis pada tabel 3.1 dan tabel 3.2.

Tabel 3.1 *Rule Inference* motor kiri

Depan / Belakang	LEBIH KIRI	KIRI	TENGAH	KANAN	LEBIH KANAN
KIRI	**LC**	**C**	**P**	**SP**	**SPS**
TENGAH	**SC**	**LC**	**N**	**LP**	**SP**
KANAN	**SCS**	**SC**	**C**	**P**	**LP**

Tabel 3.2 *Rule Inference* motor kanan

Depan / Belakang	LEBIH KIRI	KIRI	TENGAH	KANAN	LEBIH KANAN
KIRI	**LP**	**P**	**C**	**SC**	**SCS**
TENGAH	**SP**	**LP**	**N**	**LC**	**SC**
KANAN	**SPS**	**SP**	**P**	**C**	**LC**

Keterangan : SPS=Sangat Pelan Sekali, SP=Sangat Pelan, LP=Lebih Pelan, P=Pelan, N=Normal, C=Cepat, LC= Lebih Cepat, SC= Sangat Cepat, SCS=Sangat Cepat Sekali.

Berdasarkan tabel, maka sistem mempunyai 15 aturan fuzzy, yaitu:
1 *IF* Sensor Depan = **LEBIH KIRI** *AND* Sensor Belakang = **KIRI** *THEN* Motor DC kanan= **LEBIH CEPAT** *AND* Motor DC kiri = **LEBIH PELAN**.
2 *IF* Sensor Depan **KIRI** *AND* Sensor Belakang = **KIRI** *THEN* Motor DC kanan = **CEPAT** *AND* Motor DC kiri = **PELAN**.
.
.

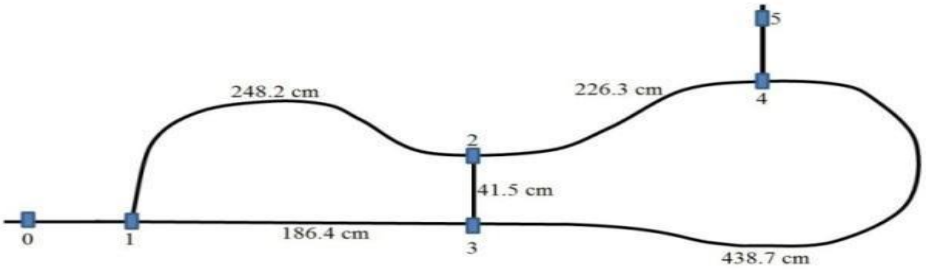
14 *IF* Sensor Depan = **LEBIH KANAN** *AND* Sensor Belakang = **TENGAH** *THEN* Motor DC kanan= **SANGAT PELAN** *AND* Motor DC kiri = **SANGAT CEPAT.**
15 *IF* Sensor Depan = **LEBIH KANAN** *AND* Sensor Belakang = **KANAN** *THEN* Motor DC kanan= **LEBIH PELAN** *AND* Motor DC kiri = **LEBIH CEPAT.**

3.1.3 *Deffuzyfication*

Deffuzifikasi merupakan pemetaan bagi nilai-nilai *fuzzy* keluaran yang dihasilkan pada tahap *rules inference* ke nilai-nilai keluaran kuantitatif. Pada perancangan robot mobil ini proses deffuzifikasi menggunakan metode *Weigth Average* dan keluaran dari proses deffuzifikasi berupa nilai PWM yang nantinya digunakan untuk mengontrol kecepatan motor DC.

3.2 Perancangan Sistem RFID

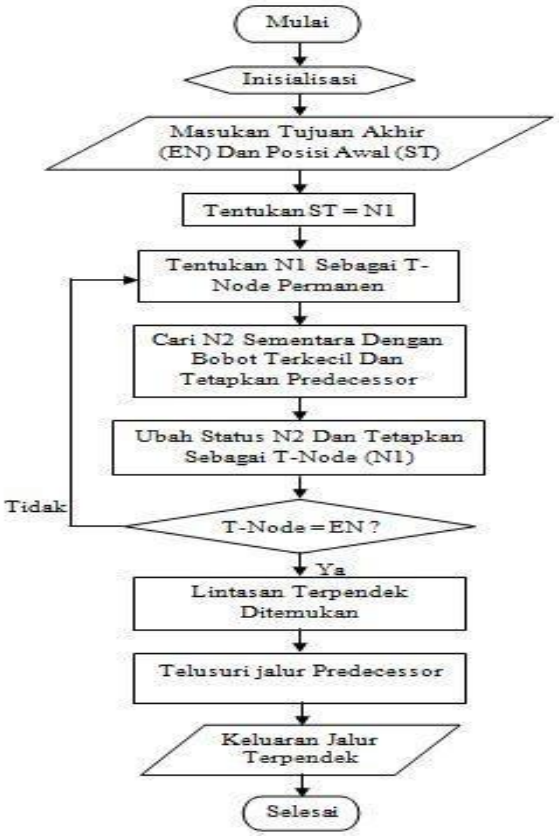
Perancangan RFID *reader* dipasang menjadi satu dengan sensor fotodiode dan diletakkan pada bagian bawah AGV. Sedangkan RFID *tag* diletakkan pada setiap terminal dan setiap persimpangan jalan dari lintasan AGV. Berikut gambar dari simpul peletakkan *tag* RFID.



Gambar 3.8 Peletakkan *tag* RFID

3.3 Perancangan Algoritma Dijkstra

Algoritma untuk menentukan rencana perjalan terpendek pada sistem ini menggunakan Algoritma Dijkstra, masukan pada sistem ini terdiri dari banyaknya simpul, jarak antar simpul, posisi awal dan tujuan akhir. Flowchart algoritma Dijkstra pada gambar 3.9.



Gambar 3.9 *Flowchart* sistem algoritma Dijkstra

4. Pengujian

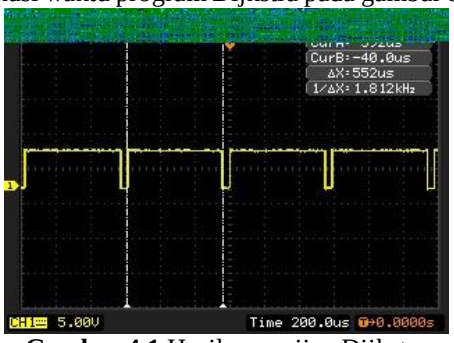
Pada bagian pengujian dibagi menjadi tiga yaitu waktu eksekusi *fuzzy logic***,** waktu eksekusi algoritma Dijkstra beserta hasil perhitungan dan perpindahan AGV dari terminal satu ke terminal empat dan sebaliknya.

Hasil pengujian untuk *fuzzy logic* menghasilkan kecepatan eksekusi dari *fuzzy logic* dapat dilihat pada tabel 4.1.

Tabel 4.1 Hasil pengujian kecepatan eksekusi *Fuzzy Logic*

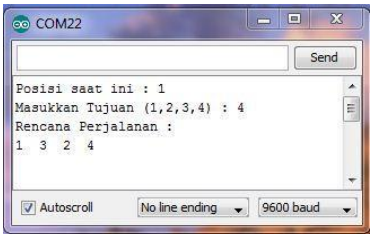
Pengujian	Nilai Sensor Depan	Nilai Sensor Belakang	Waktu Eksekusi
1	0	0	2.80ms
2	9	3	2.60ms
3	-3	5	2.60ms
4	5	0	2.80ms
5	7	-3	2.40ms
6	0	-5	2.60ms

Hasil pengujian eksekusi waktu program Dijkstra pada gambar berikut :

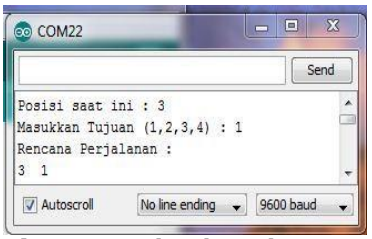


Gambar 4.1 Hasil pengujian Dijkstra

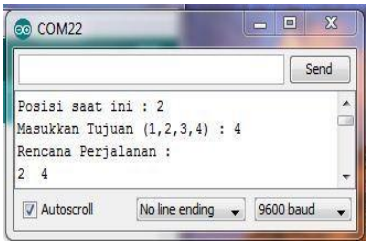
Hasil pengujian terhadap perhitungan jalur terdapat pada gambar di bawah ini :



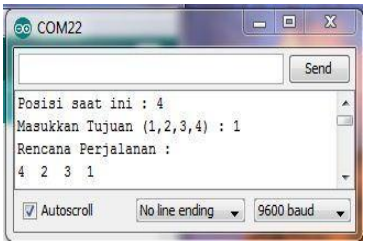
Gambar 4.2 Hasil Dijkstra dengan posisi awal terminal 1



Gambar 4.4 Hasil Dijkstra dengan posisi awal terminal 3



Gambar 4.3 Hasil Dijkstra dengan posisi awal terminal 2



Gambar 4.5 Hasil Dijkstra dengan posisi awal terminal 4

Hasil pengujian dari pergerakan AGV berdasarkan perintah yang diinginkan.

Tabel 4.2 Hasil pengujian antar terminal

No	DESTINATION			STATUS
	1 - 3	3 - 2	2 - 4	
1	OK	OK	OK	BERHASIL
2	OK	OK	OK	BERHASIL
3	OK	OK	OK	BERHASIL
4	OK	OK	OK	BERHASIL
5	OK	OK	OK	BERHASIL
6	OK	OK	OK	BERHASIL
7	OK	OK	OK	BERHASIL
8	OK	OK	OK	BERHASIL
9	OK	OK	OK	BERHASIL
10	OK	OK	OK	BERHASIL

No	DESTINATION			STATUS
	4 - 2	2 – 3	3 - 1	
1	OK	OK	OK	BERHASIL
2	OK	OK	OK	BERHASIL
3	OK	GAGAL	GAGAL	GAGAL
4	GAGAL	GAGAL	GAGAL	GAGAL
5	OK	GAGAL	GAGAL	GAGAL
6	OK	OK	OK	BERHASIL
7	OK	GAGAL	GAGAL	GAGAL
8	OK	OK	OK	BERHASIL
9	OK	OK	OK	BERHASIL
10	OK	OK	OK	BERHASIL

Berdasarkan pengujian, rata-rata waktu yang dibutuhkan untuk eksekusi *fuzzy logic* adalah 2.63 ms, waktu yang digunakan untuk eksekusi *fuzzy logic* tidak membebani kerja program dari mikrokontroler. Sedangkan lama waktu eksekusi algoritma Dijkstra hanya membutuhkan 552us, sehingga tidak mengganggu kerja program.

Berdasarkan hasil pengujian perjalanan dari terminal satu menuju terminal empat memiliki keberhasilan 100% dan perjalanan dari terminal empat menuju terminal satu memiliki keberhasilan 60% dengan jarak tempuh 454.2 cm.

5. Kesimpulan

Automatic Guided Vehicle memerlukan waktu 2.63 ms untuk mengolah *fuzzy logic* dan 552 us untuk eksekusi algoritma Dijkstra. Pergerakan *Automatic Guided Vehicle* dapat berjalan dari terminal satu menuju terminal dua dengan jalan 1 – 3 – 2 – 4 dan dari terminal empat menuju terminal satu dengan jalan 4 – 2 – 3 – 1 memiliki total panjang jalur 454.2 cm dengan tingkat keberhasilan membaca *tag* RFID adalah 80%.

6. Daftar Pustaka

[1] Waldy, Ibnu. 2014. RANCANG BANGUN SISTEM AUTOMATIC GUIDED VEHICLE (AGV) MENGGUNAKAN RFID UNTUK INFORMASI POSISI. Bandung : Universitas Telkom.

[2] Kurnia, Albert. Angelina, Friska. Dwiparaswati, Windy. 2012.”PENERAPAN METODE DIJKSTRA DALAM PENCARIAN JALUR TERPENDEK PADA PERUSAHAAN DISTRIBUSI FILM”.Depok.Universitas Gunadarma.

[3] Priyono, Agung. 2014. Perancangan dan Implementasi One Steered Traction Wheel Robot dengan Circular Line Sensor menggunakan Kontrol Logika Fuzzy. Bandung : Universitas Telkom.

[4] Romdon, Arizal. 2014. RANCANG BANGUN SISTEM NAVIGASI AUTOMATIC GUIDED VEHICLE (AGV) MENGGUNAKAN SENSOR GARIS BERBENTUK LINGKARAN DAN FUZZY LOGIC. Bandung : Universitas Telkom.