

Algoritma Koordinasi untuk Prototipe *Automated Guided Vehicle*

Ivan Hendrayanta Susilo

Program Studi Teknik Elektro, Universitas Kristen Petra

Jl. Siwalankerto 121-131, Surabaya 60236, Indonesia

E-Mail: m23412009@john.petra.ac.id

Abstrak— *Automated Guided Vehicle* adalah robot beroda yang berjalan dengan panduan. Robot AGV biasa diperintahkan untuk melakukan pembawaan barang dari suatu tempat ke tempat tertentu. Sistem ini menggunakan dua buah robot AGV yang saling berkoordinasi melalui *server* sehingga tidak terjadi tabrakakan. Tujuan pengambilan barang robot diperintahkan melalui sebuah *smartphone* dan *server* akan memerintah robot untuk berjalan hingga mencapai tujuan yang diperintahkan.

Penggunaan algoritma Wavefront pada *server* dapat membantu robot dalam pencarian rute menuju tujuan dengan keberhasilan 100%. Proses transfer data oleh robot mengalami delay 1 hingga 2 detik. Kecepatan robot 1 dan menggunakan 20 cm/detik dengan keberhasilan robot 1 sebesar 100% dan robot 2 91,7%. Penggunaan 3 buah sensor *ultrasonic* masih terdapat *blindspot*.

Kata Kunci— Arduino, Automated Guided Vehicle, Wavefront

I. PENDAHULUAN

Perkembangan toko *online* di Indonesia beberapa tahun terakhir ini sangat pesat, terbukti dengan munculnya toko – toko *online*, beberapa contoh toko *online* yang sudah ada di Indonesia adalah : Bhinneka.com, Lazada, Jakartanotebook, dan masih banyak yang lain, toko – toko *online* tersebut menyediakan berbagai macam jenis barang, dengan memiliki jenis barang yang banyak maka akan terjadi banyak transaksi, selama ini untuk mengambil barang yang dipesan di gudang masih menggunakan tenaga manusia, hal tersebut menyebabkan membutuhkan banyak tenaga kerja atau karyawan, sedangkan di luar negeri seperti contohnya di Amazon.com, menggunakan AGV untuk pengambilan barang – barang yang dipesan, dengan menggunakan robot AGV akan meningkatkan efisiensi pekerjaan.

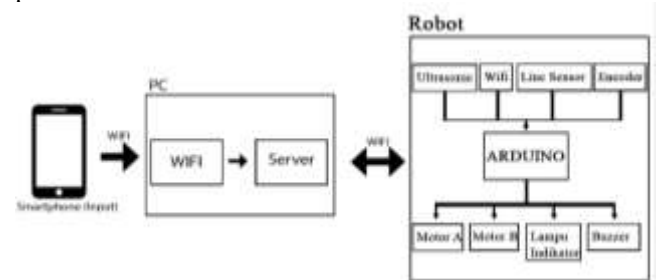
AGV (*Automated Guided Vehicle*), adalah robot beroda yang dapat berjalan berdasarkan jalur yang telah ditentukan [1]. AGV sendiri telah dikembangkan mulai dari tahun 1960 sampai awal 1980, AGV tersebut mendeteksi rel panduan yang diletakkan pada sebuah tempat, contohnya lintasan kabel yang terkubur atau lantai yang diberi garis berwarna [2]. Fungsi dari robot AGV ini sama seperti kereta api yang berjalan pada track dengan mengangkut penumpang dan melakukan pekerjaannya berulang – ulang, pada industri toko *online* adalah untuk mengangkut barang dari satu tempat ke satu tempat tertentu, Robot AGV ini sangat membantu dalam pengangkutan barang dalam jumlah yang cukup banyak dalam toko online yang memiliki gudang cukup besar, dalam gudang yang cukup besar dibutuhkan robot AGV tidak hanya satu buah, dalam jumlah AGV yang cukup besar dalam sebuah *plant* pabrik, dibutuhkan

koordinasi antara robot AGV satu dengan robot AGV yang lain, agar tidak terjadi penumpukan dalam sebuah jalur, dengan digunakan robot AGV ini maka diharapkan biaya operasional pada sebuah toko *online* akan turun meskipun investasi awal yang tinggi, selain dapat mengurangi biaya operasional penggunaan robot tersebut dapat mengurangi kecelakaan kerja, meningkatkan efisiensi pekerja, dan mengurangi permasalahan yang terjadi [3].

II. PERANCANGAN SISTEM

A. Gambaran Umum Sistem

Sistem robot *Automated Guided Vehicle* merupakan sebuah sistem yang memiliki tujuan untuk membantu penggunanya dalam pengambilan barang pada sebuah tempat. Sistem ini terdiri dari *input*, *server* pada komputer, dan *output* yang berupa robot. *Input* dari sistem ini adalah sebuah *smartphone* Android. *Smartphone* ini berfungsi untuk memilih tujuan pengambilan barang. Data tujuan dari *smartphone* dikirimkan melalui *Wifi* kepada komputer atau PC. PC tersebut terdapat *server* yang berfungsi untuk tempat penampungan data yang dikirim oleh *smartphone*. *Server* menerima data tujuan dari *smartphone* dan *server* akan melakukan proses penentuan jalur yang akan dilalui oleh robot dengan menggunakan *wavefront algorithm*, dengan algoritma tersebut didapatkan jalur untuk menuju pada *goal* atau tujuan robot tersebut. Selain *server* melakukan penentuan jalur, *server* juga akan mengkoordinasi posisi robot pertama dengan robot kedua. Dengan data – data tersebut maka *server* akan mengirimkan perintah – perintah kepada kedua robot tersebut. Komunikasi antara *server* dengan kedua robot dilakukan dengan *wifi*. Data diterima oleh Arduino di robot dengan menggunakan modul *wifi*. Setelah Arduino menerima perintah dari *server* maka Arduino akan menjalankan program sesuai perintah dari *server*. Robot ini dilengkapi dengan sensor – sensor yang akan bekerja untuk membantu robot dalam menjalankan perintah dari *server*.



Gambar 1. Desain Sistem.

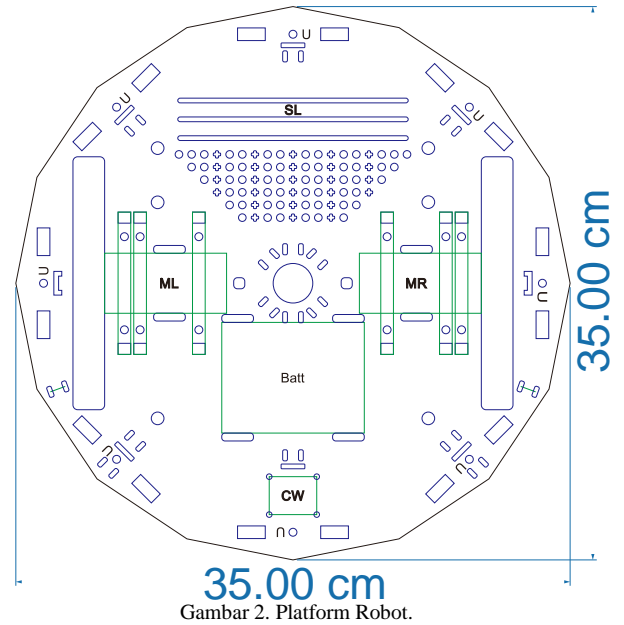
B. Desain Robot AGV

Robot AGV yang digunakan pada sistem ini berjumlah 2 buah yang masing – masing memiliki kontroler. Kontroler yang digunakan adalah Arduino Mega 2560 dan *wireless module* ESP-13. *Wireless* modul ini berfungsi untuk jalur komunikasi antara PC dengan Arduino. Selain *wireless* sensor *ultrasonic*, sensor warna, motor, *buzzer* dan lampu – lampu indikator melengkapi robot AGV tersebut. Daftar pin yang digunakan adalah sebagai berikut :

Tabel 1. Daftar Penggunaan Pin Arduino Mega 2560

Pin	Fungsi
Vin	Sumber tegangan Arduino
GND	Ground sumber
0	Tx Wireless
1	Rx Wireless
2	Encoder Kanan
3	Encoder kiri
6	Pwm driver motor kiri A
9	Pwm driver motor kiri B
10	Pwm driver motor kanan A
11	Pwm driver motor kanan B
13	Lampu Merah
33	Lampu indikator membawa barang
35	Buzzer
22	Trigger ultrasonic kiri
24	Echo ultrasonic kiri
26	Trigger ultrasonic tengah
28	Echo ultrasonic tengah
30	Trigger ultrasonic kanan
32	Echo ultrasonic kanan
39	Line sensor kiri
41	Line sensor kanan
A0	Sensor warna kiri
A1	Sensor warna tengah
A2	Sensor warna kanan

Rangkaian kontrol dan komponen – komponen lainnya yang digunakan diletakkan pada sebuah platform robot AGV, platform robot tersebut terbuat dari akrilik dengan tebal 8 mm. Platform robot memiliki ukuran 35 cm x 35 cm (Gambar 2).



Gambar 2. Platform Robot.

Pada robot tersebut menggunakan penggerak 2 buah roda. Roda tersebut menggunakan akrilik dengan tebal 8 mm dan memiliki diameter 15 cm. Roda tersebut dilengkapi dengan sticker bergaris berjumlah 10 buah garis (Gambar 3). Garis – garis tersebut digunakan untuk membantu arduino dalam membaca pergerakan robot dengan bantuan encoder. Dengan penggunaan 10 buah garis tersebut arduino dapat mengetahui pergerakan robot sebesar 4,71 cm dengan perhitungan menggunakan persamaan berikut :

$$1 \text{ garis encoder} = \frac{\text{Keliling Lingkaran}}{\text{Jumlah Garis}} \quad (1)$$



Gambar 3. Sticker bergaris pada roda.

C. Desain Software Aplikasi Android

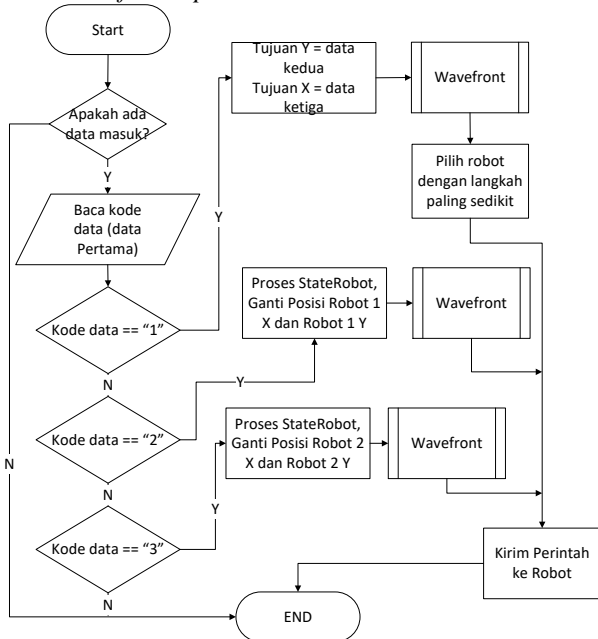
Aplikasi Android yang diprogram menggunakan *software “Android Studio”* berfungsi untuk pengiriman data tujuan pengambilan barang kepada *server* secara *wireless*. Aplikasi ini memiliki fungsi untuk pembacaan penekanan tombol tujuan barang dan tombol kirim. Tombol tujuan barang berfungsi untuk memilih barang yang akan diambil

oleh robot AGV sedangkan tombol kirim untuk mengirimkan data barang yang dipilih kepada *server*.



Gambar 4. Layout pada aplikasi.

D. Desain Software pada Server

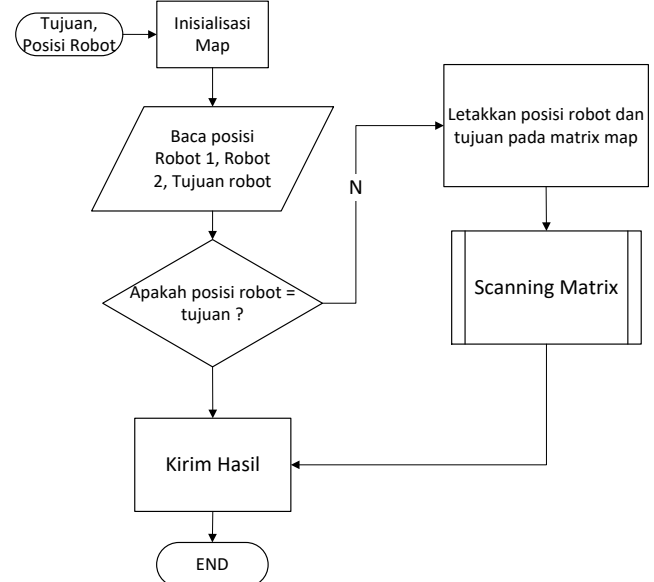


Gambar 5. Flowchart sistem kerja *server*.

Server akan membaca data yang masuk secara terus menerus dan memproses data tersebut dengan membagi dengan kode – kode tertentu. Kode pada sistem ini dibagi menjadi 3, yaitu kode “1” untuk *server* menerima input tujuan, kode “2” dan “3” sebagai data bahwa robot telah mencapai tujuan. Pada saat *server* menerima kode 1, maka *server* akan membaca tujuan Y dan tujuan X dan melakukan kalkulasi dengan memanggil algoritma Wavefront. Ketika dua robot sedang diam atau tidak bekerja, *server* akan mengkalkulasi step yang dibutuhkan robot 1 dan robot 2 untuk mencapai tujuan tersebut. *Server* memilih robot yang memiliki step paling sedikit untuk mencapai tujuan tersebut, lalu mengirimkan perintah kepada robot yang telah dipilih.

Server menerima kode “2”, kode tersebut adalah kode dari robot pertama. Kode tersebut berfungsi untuk memberitahu *server* bahwa dia telah menyelesaikan tugas yang diberikan oleh *server*. Pertama – tama *server* akan melakukan pengecekan *state* pada robot. Pengecekan *state* pada robot tersebut berfungsi untuk mengetahui bahwa robot tersebut sedang dalam tugas mengambil barang, mengantarkan barang, mengembalikan barang atau robot sedang kembali ke rumah. *State* akan berubah ketika robot tersebut mencapai tujuan, perubahan *state* tersebut berfungsi untuk merubah tujuan robot kembali ke rumah atau menuju ke tempat pengambilan barang. Ketika *state* tersebut belum terpenuhi *server* akan memperbarui posisi robot 1 dan melakukan kalkulasi dengan algoritma Wavefront. Hasil dari algoritma tersebut akan dikirimkan kepada robot 1. Kode “3” adalah kode untuk robot kedua memberitahu *server* bahwa, robot tersebut telah melaksanakan tugas yang diberikan oleh *server* kepada robot dan melakukan proses *state* sama seperti robot 1.

E. Algoritma Wavefront



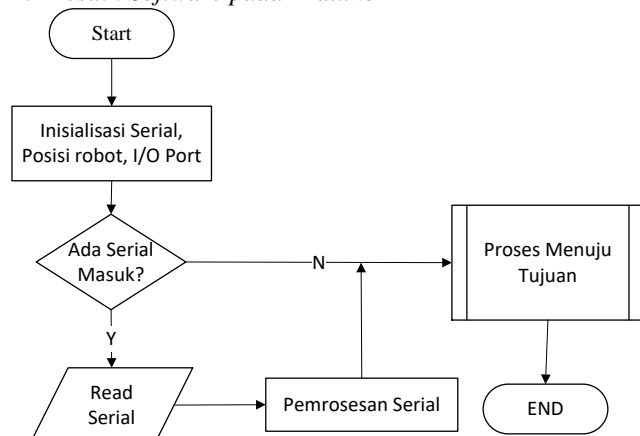
Gambar 6. Flowchart Algoritma Wavefront.

Algoritma ini berjalan ketika dipanggil dan membutuhkan parameter tujuan robot dan posisi robot. Langkah pertama dalam algoritma ini adalah menginisialisasi *map* yang telah dimasukkan oleh *user* sebelum *server* dijalankan dan membaca posisi robot dan tujuan robot. Setelah *map* terinisialisasi maka yang dilakukan selanjutnya adalah membaca apakah posisi robot tersebut sama dengan tujuan, ketika tidak sama maka program akan melakukan peletakan posisi dan tujuan robot pada *map*. Proses selanjutnya program akan melakukan *scanning matrix* hingga posisi robot sama dengan tujuan dan mengirimkan hasil dari proses tersebut.

Proses *Scanning matrix* akan melakukan pengecekan setiap jalan yang dapat dilalui oleh robot menggunakan metode *Von Neumann neighborhood* [4]. Pengecekan jalan yang dapat dilalui dengan melakukan *scanning matrix y* dan *matrix x*. Ketika *scanning* menemukan jalan yang dapat dilalui oleh robot (*node*), program akan melakukan

pengecekan disekitar *node* tersebut. Program tersebut akan melakukan pengecekan di sekitar *node*, dan mengambil angka terkecil yang diperoleh dari sekitar *node* tersebut. Angka tersebut akan ditambah dengan nilai 1 dan dimasukkan ke matrix $map[y][x]$. Program tersebut akan terus berulang hingga menemukan posisi robot dan merubah posisi robot pada *node* terakhir yang dicek hasil perpindahan posisi robot akan dimasukkan pada *list*. Pada saat pengulangan program tersebut mencapai 100 kali, hal tersebut menunjukkan bahwa program tersebut tidak menemukan jalan untuk robot tersebut. Program akan memberikan hasil tidak menemukan jalan dan hasil tersebut akan dikirimkan kepada robot dengan perintah menunggu.

F. Desain Software pada Arduino



Gambar 7. Flowchart program Arduino.

Program pada Arduino akan terus berada pada posisi *standby*, menunggu adanya serial yang didapat. Jika program mendeteksi adanya data serial yang masuk, maka program tersebut akan mengambil data pertama dan akan melakukan pemrosesan serial. Pemrosesan serial dilakukan sesuai dengan data pertama yang didapat. Pada saat data pertama adalah angka “1”, maka tujuan y adalah angka kedua yang diterima dan tujuan x adalah angka ketiga. Data pertama adalah angka “2” maka robot akan menyalakan indikator membawa barang dan merubah tujuan x dan tujuan y. Ketika data pertama adalah angka “3” robot akan mematikan indikator membawa barang dan merubah tujuan x dan tujuan y. Ketika data pertama adalah angka “4” maka robot akan merubah *state* robot menjadi “4”. Setelah melakukan pemrosesan data, program akan menjalankan proses menuju tujuan dan program tersebut akan kembali dalam keadaan siap terima serial hingga arduino ini dimatikan.

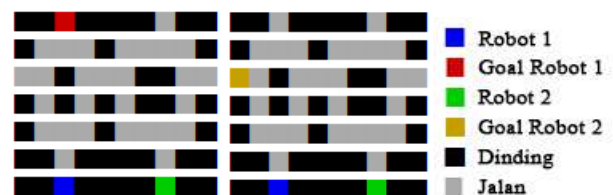
Pada proses menuju tujuan hal pertama yang dilakukan adalah membaca nilai dari sensor warna untuk membaca kotak *silver* pada bawah robot. Setelah melakukan pembacaan, program akan melakukan penentuan arah robot. Penentuan arah robot dilakukan dengan membandingkan posisi robot sekarang dengan tujuan yang akan dituju oleh robot. Pada saat arah yang akan dituju oleh robot berbeda dengan arah robot sekarang, robot akan melakukan perubahan arah hingga arah tersebut sama dengan arah yang akan dituju robot. Perubahan arah dilakukan dengan

menggunakan perintah belok kiri dan belok kanan. Ketika arah robot telah sama dengan arah tujuan maka robot akan berjalan lurus menuju tujuan. Pada saat robot berjalan lurus, robot akan menyalakan sensor *ultrasonic* dan membandingkan nilai dari sensor warna. Nilai sensor warna tersebut dibandingkan dengan nilai dari warna *silver*. Perbandingan tersebut bertujuan agar robot dapat mendeteksi warna *silver*. Pada saat warna *silver* terdeteksi maka robot akan melakukan perubahan posisi robot sesuai dengan arah robot itu berjalan dan mengirim serial sesuai dengan identitas robot tersebut.

III. PENGUJIAN SISTEM

A. Pengujian Proses Penentuan Jalan Robot pada Server

Pengujian ini bertujuan untuk mengetahui tingkat keberhasilan *server*, dalam mengatur robot menuju ke tempat tujuan. Pengujian yang dilakukan dengan menentukan koordinat pengambilan barang, lalu memintahkan robot untuk mengambil barang pada koordinat tersebut. Setelah selesai lakukan hal yang sama pada *map* berbeda



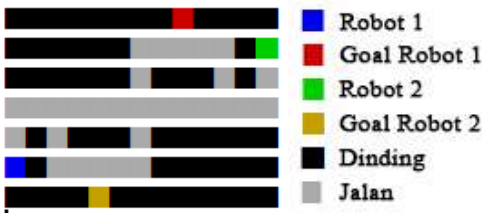
Gambar 8. Titik Tujuan Pengambilan Barang pada Server.

Penentuan tujuan pengambilan barang dilakukan dengan aplikasi android pada contoh pengujian di atas, tujuan pengambilan barang adalah koordinat (0,2), (2,0), (0,7) dan (2,9).

Tabel 2. Hasil Pengujian Server

Pengujian Tujuan	1	2	3	4	5
(0,2) (2,0)	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
(0,2) (0,7)	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
(0,2) (2,9)	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
(2,0) (0,2)	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
(2,0) (0,7)	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
(2,0) (2,9)	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
(0,7) (0,2)	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
(0,7) (2,0)	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
(0,7) (2,9)	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
(2,9) (0,2)	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
(2,9) (2,0)	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
(2,9) (0,7)	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil

Hasil pengujian tersebut membuktikan bahwa *server* dengan algoritma Wavefront berjalan dengan sempurna pada *map* pertama. Hal tersebut terbukti dengan keberhasilan robot 1 dan robot 2 dalam mengambil barang pada semua kombinasi tujuan yang ada. Pengujian tahap kedua dilakukan dengan *map* sebagai berikut :



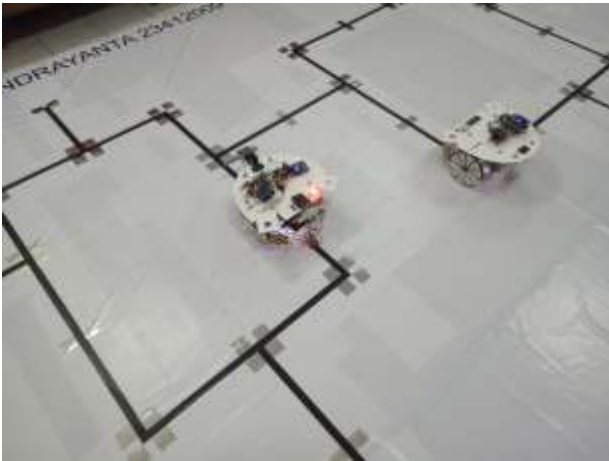
Gambar 9. Map 2 untuk pengujian Server

Pengujian dengan *map* berbeda ini bertujuan untuk membuktikan algoritma ini dapat digunakan pada *map* lain.

Tabel 3. Hasil Pengujian Map 2

Pengujian Tujuan	1	2	3	4	5
(0,8) (6,4)	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil
(6,4) (0,8)	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil

Berdasarkan Tabel 3 pengujian dengan *map* berbeda. Hasil pengujian tersebut membuktikan bahwa dengan *map* yang berbeda. *Server* dapat menentukan jalannya robot sampai ke tujuan dengan keberhasilan 100%. Berikut adalah gambar robot 1 dan 2 pada saat melaksanakan perintah :



Gambar 10. Robot 1 dan Robot 2 pada saat melaksanakan tugas.

B. Pengujian Pengiriman Data dengan Modul Wifi

Pada tugas akhir ini digunakan modul *wifi* untuk melakukan komunikasi antara server dengan robot. Modul *wifi* digunakan untuk mengirim dan menerima data. Pengujian ini dilakukan untuk mengetahui tingkat keakuratan dan kecepatan modul *Wifi* untuk proses pengiriman data dari robot menuju *server* dan sebaliknya, metode yang digunakan adalah metode perbandingan. Pengujian ini dilakukan dengan cara mensinkronkan waktu pada *Arduino* dengan komputer dan ditampilkan pada serial lalu kirim data waktu tersebut kepada *server*. *Server* akan menampilkan data yang dikirim oleh *Arduino* dengan waktu terima sesuai dengan waktu komputer, lakukan hal sebaliknya untuk mengetahui waktu *server* kirim ke *Arduino*.

Tabel 4. Pengujian Pengiriman Data dari Arduino

Data ke	Waktu Arduino	Waktu Server Terima	Selisih Waktu	Kesalahan Data
1	20:50:33	20:50:33	0 Detik	0
2	20:50:34	20:50:34	0 Detik	0
3	20:50:35	20:50:35	0 Detik	0
4	20:50:36	20:50:36	0 Detik	0
5	20:50:37	20:50:37	0 Detik	0
6	20:50:38	20:50:38	0 Detik	0
7	20:50:39	20:50:39	0 Detik	0
8	20:50:40	20:50:40	0 Detik	0
9	20:50:41	20:50:41	0 Detik	0
10	20:50:42	20:50:42	0 Detik	0

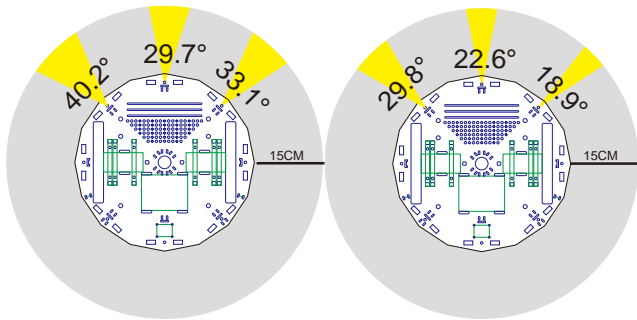
Tabel 5. Pengujian Pengiriman Data dari Server

Data ke	Waktu Server	Waktu Arduino Terima	Selisih Waktu	Kesalahan Data
1	21:33:24	21:33:25	1 Detik	0
2	21:37:46	21:37:47	1 Detik	0
3	21:39:18	21:39:19	1 Detik	0
4	21:46:30	21:46:31	1 Detik	0
5	21:49:52	21:49:54	2 Detik	0
6	21:55:14	21:55:15	1 Detik	0
7	21:58:11	21:58:13	2 Detik	0
8	22:01:14	22:01:15	1 Detik	0
9	22:04:20	22:04:21	1 Detik	0
10	22:12:42	22:12:43	1 Detik	0

Dari kedua hasil pengujian di atas, terlihat bahwa tingkat kesalahan dalam pengiriman data menggunakan *Wifi* dengan data sebanyak 20 tidak ada kesalahan. Hasil waktu pengiriman terdapat perbedaan pada saat *Arduino* mengirim dan menerima data. Pada saat *Arduino* mengirim data kepada *server* tidak terjadi *delay* pengiriman. Hasil berbeda ditunjukkan pada saat *Arduino* menerima data dari *server* terlihat pada Tabel 5. Setiap penerimaan data oleh *Arduino* mengalami *delay*, keterlambatan penerimaan data yang terjadi 1 hingga 2 detik. Keterlambatan penerimaan tersebut terjadi pada modul *Wifi* robot 1 dan robot 2. Keterlambatan dalam penerimaan membuat robot berhenti pada setiap robot meminta perintah pada *server*.

C. Pengujian Sudut Kerja Sensor Ultrasonic

Sensor *ultrasonic* HC-SR04 pada sistem ini menggunakan 3 buah HC-SR04, sensor *ultrasonic* tersebut memiliki keterbatasan dalam sudut pengukuran. Pengujian ini bertujuan untuk mengetahui sudut kerja dari sensor *ultrasonic*, sehingga mengetahui *blindspot* pada robot. Pengujian tersebut dilakukan dengan cara memposisikan benda dengan jarak 15 cm dari *ultrasonic* dan benda tersebut digeser ke kiri dan ke kanan.

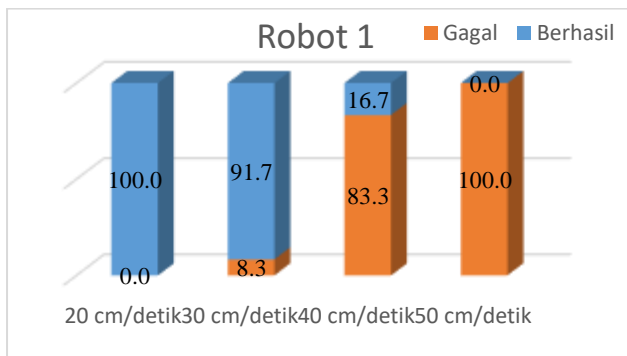


Gambar 11. Sudut Pengukuran dan *Blindspot* robot 1 (kiri) dan robot 2.

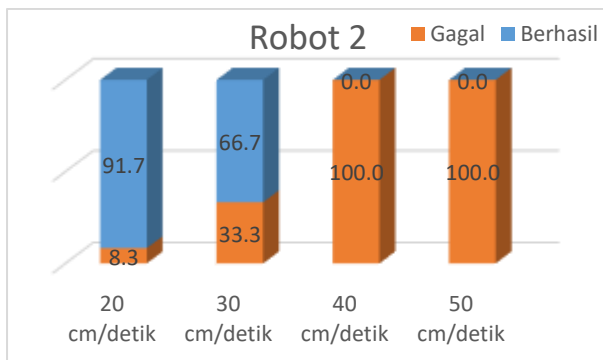
Dari gambar diatas dapat dilihat bahwa sudut pengukuran *ultrasonic* hanya terbatas, dengan maksimal sudut sebesar 40 derajat dan terkecil 18,9 derajat. Sudut dari 3 *ultrasonic* tersebut tidak dapat menjangkau semua bagian depan robot. Robot masih dapat menabrak benda yang berada pada depan robot ketika benda tersebut berada pada *blindspot* robot.

D. Pengujian Robot Menuju Titik Tujuan dengan Kecepatan Berbeda

Pengaturan PWM (*Pulse Width Modulation*) dapat membuat perubahan kecepatan pada *output* motor. Kecepatan tersebut dapat mempengaruhi kinerja robot dalam menuju ke titik tujuan. Pengujian ini bertujuan untuk mendapatkan kecepatan robot dengan *error* terkecil dalam melaksanakan tugas yaitu menuju ke titik pengambilan barang. Pengujian ini dilakukan dengan cara mengatur kecepatan robot 20 cm/detik, 30 cm/detik, 40 cm/detik, 50 cm/detik, lalu memerintahkan robot menuju titik tujuan pengambilan barang.



Gambar 12. Persentase keberhasilan robot 1.



Gambar 13. Persentase keberhasilan robot 2.

Bar chart di atas menunjukkan persentase keberhasilan robot dengan variasi kecepatan. Pada robot 1 memiliki keberhasilan paling besar pada kecepatan 20 cm/detik, sehingga kecepatan yang dipilih untuk robot 1 adalah 20 cm/detik dan pada robot 2 memiliki keberhasilan 91,7% pada kecepatan 20 cm/detik dan keberhasilan 66,7 pada kecepatan 30 cm / detik. Sehingga kecepatan yang dipilih pada robot 2 adalah 20 cm/detik.

IV. KESIMPULAN

Kesimpulan yang didapat dari pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Pencarian rute menggunakan algoritma wavefront dapat berjalan dengan hasil 100% berhasil dan dapat digunakan pada *map* yang berbeda.
2. *Server* akan mencari jalan alternatif ketika jalur yang akan dilalui digunakan oleh robot lain dan robot akan menunggu di tempat ketika jalur alternatif tersebut tidak ada.
3. Area cakupan sensor *ultrasonic* adalah seperti pada pengujian 4.5. Dalam hal ini, sensor *ultrasonic* yang dipakai belum menjangkau semua area depan robot atau masih terdapat *blindspot* .
4. Proses transfer data melalui modul *Wireless* ESP8266 dapat berjalan dengan baik, karena tidak ada kesalahan dalam pengiriman data, namun memiliki kekurangan yaitu adanya *delay* 1 – 2 detik pada saat penerimaan data.
5. Kecepatan optimal robot untuk melakukan tugasnya adalah kecepatan 20 cm/detik untuk robot 1 dengan keberhasilan sebesar 100% dan kecepatan 20 cm/detik untuk robot 2 dengan keberhasilan sebesar 91,7%.

DAFTAR PUSTAKA

- [1] Latief, S. M. (2012). Prototype Automatic Guide Vehicle(AGV).
- [2] Roberts, J. M., Duff, E. S., & Corke, P. I. (2002). Reactive navigation and opportunistic localization for autonomous underground mining vehicles, *145*, 127–146.
- [3] Oliveira, P. B. D. M., Pires, E. J. S., & Novais, P. (2015). Neurocomputing design of posicast PID control systems using a gravitational search algorithm. *Neurocomputing*, *167*, 18–23. <http://doi.org/10.1016/j.neucom.2014.12.101>
- [4] Varshavskaya, P. (2016, 5 19). *Intelligent Robotics*. Retrieved from <http://www.cs.tufts.edu/comp/150IR/labs/wavefront.html>