

Kontrol Otomatis pada Robot Pengantar Barang dengan Parameter Masukan Jarak dengan Objek dan Posisi Robot

Ricky Jeconiah¹, Darmawan Utomo², Saptadi Nugroho³

Program Studi Sistem Komputer,
Fakultas Teknik Elektronika dan Komputer,
Universitas Kristen Satya Wacana, Salatiga
¹ricky_jeconiah@yahoo.co.id, ²darmawan@staff.uksw.edu, ³saptadi_nugroho@yahoo.com

Ringkasan

Pada sistem robot pengantar barang otomatis tentu robot harus dapat menghindari halangan yang ada disekitarnya agar tidak menimbulkan kerugian. Oleh karena itu dirancang sebuah kecerdasan buatan dengan mengandalkan *simple reflex agent* dan *goal based agent* yang dapat berkomunikasi dengan robot. *Simple reflex agent* yang memiliki kemampuan cepat dalam mengambil keputusan dapat digunakan untuk menghindari objek yang menghalangi Untuk dapat menghindari halangan tentunya dibutuhkan sensor yang dapat mendeteksi jarak objek yang berada disekitarnya oleh karena itu digunakan sensor kinect yang dapat mendeteksi objek yang berada satu sampai empat meter dari sensor. *Goal based agent* yang sesuai dengan namanya dapat memastikan robot dapat mencapai tujuan. Dari hasil pengujian diperoleh data bahwa kinect memiliki ralat berkisar antara 10 mm sampai dengan 262 mm. Pada pengujian *goal based agent*, algoritma DFS dan A* dibandingkan. Menggunakan delapan titik yang merepresentasikan jalur yang akan dilalui didapat hasil yaitu kedua algoritma berjalan kurang dari seratus mikro detik sehingga tidak memberikan perbedaan pada jalannya sistem. Tetapi karena pada prosesnya algoritma DFS menggunakan memori yang lebih sedikit maka sistem ini menggunakan algoritma DFS. *Goal based agent* dapat mengarahkan robot untuk mencapai tujuan. *Simple reflex agent* dapat menghindari benda dengan tinggi lebih dari 50 cm.

Kata kunci: robot otomatis, kecerdasan buatan, kinect

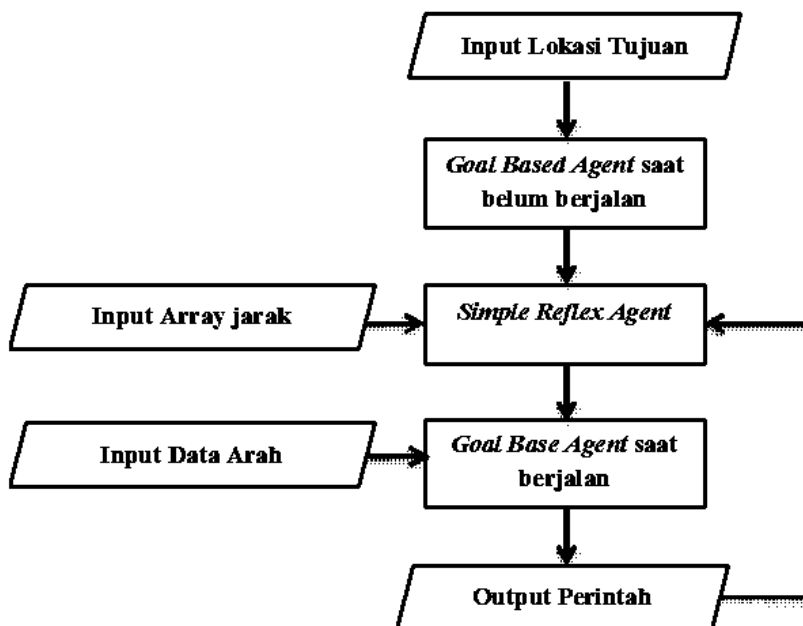
1. Pendahuluan

Di era yang moderen ini sudah banyak otomatisasi sistem yang dapat membantu manusia, salah satunya adalah robot pengantar barang. Pada robot pengantar barang sistem dibuat agar robot dapat bergerak tanpa kendali manusia sama sekali, operator manusia hanya perlu memasukkan barang dan memasukkan tujuan pengantaran. Di Indonesia sendiri sudah banyak orang yang meneliti dan membuat sistem untuk otomatisasi robot pengantar barang. Tetapi sistem yang dibuat atau dirancang masih membutuhkan jalur berupa garis yang telah dibuat sebelumnya. Sistem otomatisasi yang menggunakan garis memiliki beberapa kekurangan seperti hanya dapat berjalan secara otomatis dengan mengikuti garis yang telah dibuat. Oleh karena dibuat pengotomatisan robot yang tidak menggunakan garis sehingga bergerak selama masih terdapat ruang bergerak untuk menghindari objek yang menghalangi tanpa tergantung dengan garis yang ada.

Sistem ini dirancang menggunakan kecerdasan buatan yang berdasar pada simple reflex agent dan goal based agent. Simple reflex agent memiliki tujuan mengambil keputusan dari input yang diperoleh dan goal based agent akan mengambil keputusan agar dapat mencapai tujuan dengan memperhatikan perubahan pada lingkungannya [1, h. 40-44]. Simple reflex agent akan mendapatkan masukan data jarak dari sensor Kinect, yang digunakan untuk mendeteksi dan menghindari halangan. Sedangkan goal based agent akan menerima masukan berupa data kompas digital dan posisi robot. Pada goal based agent untuk dapat menentukan jalur yang akan dilalui digunakan algoritma pencarian jalur. Algoritma pencarian jalur DFS dan A* dibandingkan terlebih dahulu untuk menentukan algoritma yang lebih baik pada peta yang dibuat.

2. Perancangan Sistem

Perancangan yang dilakukan yaitu perancangan perangkat keras, perancangan perangkat lunak untuk kinect, dan perancangan agen cerdas yaitu goal based agent dan simple reflex agent. Cara kerja sistem dapat digambarkan seperti pada Gambar 1. Ketika mendapat informasi lokasi tujuan, informasi lokasi tujuan diolah oleh goal based agent sehingga memperoleh jalur yang harus dilalui. Setelah memiliki jalur yang dilalui maka simple reflex agent akan mengolah data jarak agar dapat menghindar jika terdapat halangan. Kemudian goal based agent yang dirancang untuk bekerja saat berjalan akan mengoreksi arah jika robot keluar jalur dan memeriksa apakah robot sudah sampai pada tujuan.



Gambar 1. Urutan Cara Kerja Sistem

2.1. Perancangan Perangkat Keras

Robot pengangkut barang yang digunakan merupakan robot milik FTEK UKSW yang telah dirancang sebelumnya. Robot pengangkut barang yang telah ada sebelumnya[2], diberi perangkat tambahan berupa kinect dan komputer yang digunakan

untuk menjalankan sistem/program kontrol otomatis. Untuk peletakkannya Kinect diletakkan pada bagian depan robot dengan sudut hadap 15 derajat ke arah bawah.

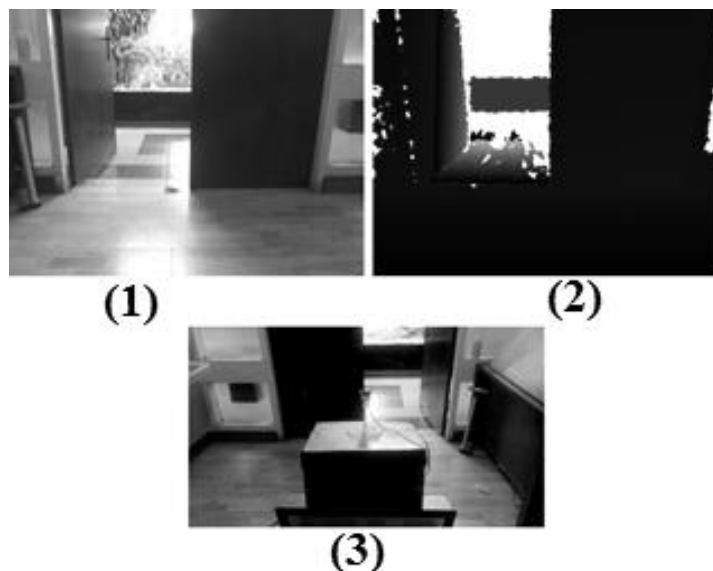
2.2. Perancangan Perangkat Lunak untuk Robot Pengangkut Barang

Perancangan perangkat lunak untuk robot pengangkut barang merupakan perangkat lunak yang telah dirancang sebelumnya agar robot dapat berjalan [2], dengan perubahan program pada Raspberry Pi dan pada kontrol manual. Perubahan pada raspberry yaitu penambahan pembacaan perintah agar dapat meneruskan perintah dari kontrol manual ke sistem kontrol otomatis. Sedangkan penambahan pada kontrol manual yaitu penambahan form untuk mengubah tujuan dan mengubah parameter posisi robot.

2.3. Perancangan Perangkat Lunak untuk Kinect

Pada bagian ini perangkat lunak menggunakan kode sumber yang telah disediakan oleh Microsoft bernama Kinect Toolbox. Terdapat dua class utama pada Kinect Toolbox yang digunakan pada sistem ini yaitu DepthStreamManager dan ColorStreamManager. Kedua class tersebut digunakan untuk menampilkan data gambar yang diterima dari kinect ke layar. Selain itu kedua class tersebut juga dimodifikasi agar dapat digunakan untuk menyimpan data gambar untuk sistem pemosisian lokal.

Perlu diketahui data gambar yang ditampilkan tidak sesuai dengan kondisi asli yang ada di depan kinect. Gambar yang dihasilkan dicerminkan terhadap sumbu y sehingga bagian yang terdapat pada kanan gambar merupakan bagian kiri dari robot, dan bagian kiri dari gambar merupakan bagian kanan dari robot. Untuk koordinat yang digunakan dalam penulisan ini akan mengikuti koordinat sesuai dengan data gambar yang dihasilkan sehingga koordinat dengan nilai x kurang dari 320 (bagian kiri gambar) merupakan bagian kanan dari robot dan sebaliknya, koordinat dengan nilai x lebih dari 320 merupakan bagian kiri dari robot. Sebagai contoh dapat dilihat pada Gambar 2.



Gambar 2. (1) Gambar Berwarna yang Diambil Kinect, (2) Gambar Representasi Data Jarak yang Diperoleh Kinect, (3) Gambar Kamera Normal yang Diambil dari Belakang Robot.

2.4. Perancangan *Goal Based Agent*

Goal based agent pada sistem ini dapat dibagi menjadi dua bagian utama yaitu bagian yang dipanggil pada saat awal mencari tujuan dan pada saat robot mendapat informasi lokasi terbaru ketika berjalan. Pada bagian awal robot harus mampu mencari jalan mengarah ke tujuan, dan pada saat menerima data lokasi baru robot harus dapat memeriksa apakah sudah mencapai lokasi tujuan atau belum. Pada saat menerima lokasi baru robot juga harus mampu mengarahkan robot ke titik tujuan yang harus dilewati sebelum mencapai tujuan.

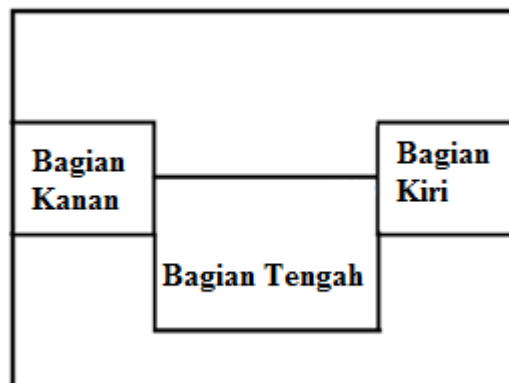
Bagian awal dipanggil ketika ada informasi tujuan dari pusat kontrol. Pada bagian ini sistem akan menentukan jalur yang akan dilalui oleh robot. Penentuan jalur ini menggunakan algoritma pencarian jalur depth first search (DFS). Setelah memperoleh jalur yang akan dilalui maka sistem akan memeriksa apakah jalur yang diberikan berisi lebih dari 1 titik atau tidak, jika tidak lebih dari satu diasumsikan tujuan sama dengan posisi sekarang, jika berisi lebih dari 1 titik maka sistem akan mengarahkan robot ke titik yang harus dicapai yaitu titik ke-2 pada array jalur.

Untuk bagian yang dipanggil pada saat adanya pembaruan informasi lokasi, sistem akan memeriksa apakah lokasi terbaru terdapat pada jalur yang harus dilalui atau tidak, jika tidak maka sistem akan membuat jalur baru dengan lokasi terbaru tersebut sebagai titik awal, jika terdapat pada jalur yang harus dilalui maka sistem akan mengarahkan robot ke tujuan berikutnya sampai mencapai tujuan.

2.5. Perancangan *Simple Reflex Agent*

Seperti yang telah disebutkan sebelumnya simple reflex agent pada sistem ini bertujuan untuk menghindari objek yang menghalangi robot. Pada bagian ini data jarak dibagi menjadi beberapa bagian yaitu bagian kiri, tengah dan kanan. Selain dibagi menjadi tiga bagian data jarak juga dibatasi untuk mempercepat waktu pemrosesan.

Pembatasan untuk setiap bagian antara lain untuk bagian kiri dibatasi sumbu x bernilai 460 sampai 639 dan sumbu y 140 sampai 279, untuk bagian tengah dibatasi sumbu x dengan nilai 180 sampai 459 dan sumbu y bernilai 210 sampai 399, dan bagian kanan dengan batasan sumbu x bernilai 0 sampai 179 dan sumbu y bernilai 139 sampai 279. Untuk lebih jelasnya pembagian data dapat dilihat di Gambar 3. Bagian lain dari data jarak tidak diperiksa dengan asumsi data tersebut berada terlalu jauh dari kinect atau terlalu dekat dengan kinect sehingga sudah direpresentasikan oleh bagian yang diperiksa.



Gambar 3. Bagian yang Diperiksa pada *Simple Reflex Agent*.

Untuk logika yang digunakan dapat dilihat di Tabel 1. Dengan nilai 0 berarti tidak terdapat halangan dan nilai 1 terdapat halangan. Nilai 1 dan 0 pada program diperoleh dari pengecekan data jarak akan dianggap terdapat halangan jika ditemukan sekumpulan data jarak yang kurang dari 1400 ($\pm 1,3m$ dari robot) untuk bagian tengah dan kurang dari 1500 untuk bagian samping (kanan dan kiri) dalam lingkup 20×10 pixel dari titik dengan jarak kurang dari batas yang pertama kali ditemukan. Pada pengecekan juga terdapat kasus khusus untuk jarak yang tidak diketahui, data tersebut hanya akan dianggap halangan jika telah ditemukan titik dengan data jarak kurang dari 1400 untuk bagian tengah dan kurang dari 1500 untuk bagian samping sebelum data -1.

Jalannya program dapat dijelaskan sebagai berikut, setelah memperoleh data jarak, sistem akan memeriksa data jarak pada titik – titik yang telah ditentukan sebelumnya, yaitu titik pada bagian kiri, tengah dan kanan. Pada pemeriksaan data tersebut tidak semua titik diperiksa dengan alasan mempercepat waktu pemrosesan. Dari jumlah titik yang ditandai maka akan ditentukan apakah ada atau tidaknya halangan, dianggap terdapat halangan jika terdapat lebih dari 35 titik yang ditandai dalam area persegi 20×10 pixel. Dari setiap informasi tentang halangan yang telah diproses kemudian ditentukan keputusan yang telah ditentukan dalam array tiga dimensi yang telah ditentukan sebelumnya (Tabel 1).

Tabel 1. Logika Pengambilan Keputusan oleh *Simple Reflex Agent*

Halangan di Bagian Kiri	Halangan di Bagian Tengah	Halangan di Bagian Kanan	Keputusan
0	0	0	Maju
0	0	1	Putar Kiri
0	1	0	Berhenti
0	1	1	Putar Kiri
1	0	0	Putar Kanan
1	0	1	Maju
1	1	0	Putar Kanan
1	1	1	Berhenti

3. Pengujian dan Analisis

Pada bagian ini akan dibahas pengujian yang dilakukan pada sistem. Pengujian dilakukan pada kinect untuk mengetahui batasan yang dimiliki dan dilakukan pada sistem yang telah dibuat antara lain simple reflex agent dan goal based agent. Tujuan dari pengujian adalah memastikan sistem dapat bekerja sesuai dengan yang diharapkan pada perancangan.

3.1. Pengujian Kinect

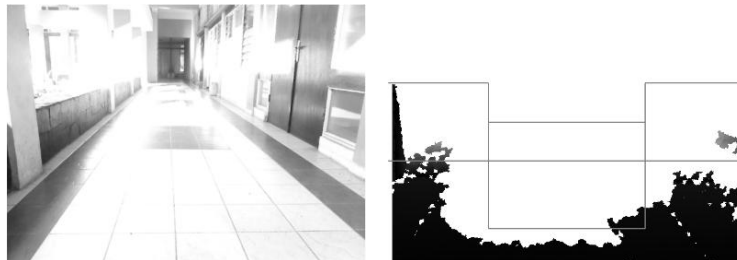
Pengujian yang dilakukan berupa perbandingan data jarak yang dihasilkan kinect dengan jarak sebenarnya. Pengujian dilakukan dengan mode normal kinect pada jarak 80 cm sampai 400 cm, jarak diuji 80 cm dan selanjutnya setiap 30 cm dimulai dari 90 cm. Pengujian ini dilakukan pada kondisi lingkungan ideal dengan area yang terlihat tidak menerima sinar matahari dan pada kondisi permukaan yang datar. Pengukuran jarak dari kinect hingga titik yang diukur, diukur menggunakan penggaris 30 cm jika ditambahkan dengan kesalahan manusia ralat pengukuran bernilai ± 5 cm. Hasil pengujian dapat dilihat pada Tabel 2.

Tabel 2. Hasil Pengujian Data Jarak Kinect terhadap Data Sebenarnya

Jarak sebenarnya (cm)	Data Kinect
<80	-1
80	816
90	910
120	1210
150	1530
180	1826
210	2135
240	2475
270	2799
300	3109
330	3383
360	3745
370	3861
380	4062
>380	4095

Hasil keluaran dari kinect berupa data dengan tipe integer yang menunjukkan korelasi dengan jarak. Dari hasil pengujian data kinect akan bernilai -1 jika jarak tidak dapat diketahui dengan penyebab jarak terlalu dekat. Untuk jarak yang terdeteksi (80 cm sampai 380 cm) data kinect yang diperoleh memiliki perbandingan satu banding sepuluh dengan data kenyataan. Jika 1 satuan data kinect disamakan dengan 1 mm maka data yang dihasilkan memiliki ralat antara 10 mm sampai 262 mm.

Pada percobaan dengan lingkungan di depan kinect menerima sinar matahari, kinect tidak dapat memberikan data jarak karena terganggu oleh sinar matahari. Meskipun sinar matahari tidak langsung mengarah ke kinect tetapi kinect tetap tidak dapat mendeteksi jarak dengan baik. Kondisi dan hasil data jarak ketika terdapat sinar matahari dapat dilihat pada Gambar 4.

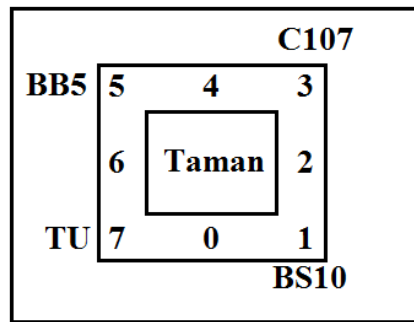


Gambar 4. Data yang Diperoleh pada Lingkungan yang Terdapat Sinar Matahari

3.2. Pengujian *Goal based agent*

3.2.1. Pengujian Algoritma Penyelesaian *Graph*

Pada pengujian ini bagian yang diuji merupakan pengujian pencarian jalur oleh algoritma DFS maupun A*. Kedua algoritma diuji menggunakan graph representasi gedung C yang sederhana dengan delapan titik. Graph yang digunakan merepresentasikan peta seperti pada Gambar 5.



Gambar 5. Representasi Titik yang Digunakan Dalam Pengujian

Untuk lamanya proses dilakukan beberapa pengujian dengan masing masing mengulang kasus yang sama sebanyak 30000 kali. Kasus yang diuji yaitu dari 0 ke 1, 0 ke 2, 0 ke 3 dan 0 ke 4. Pengujian tersebut bertujuan untuk mengetahui lamanya proses dengan jumlah titik yang dikunjungi berbeda - beda. Hasil pengujian dapat dilihat pada tabel 3.

Tabel 3. Waktu yang Dibutuhkan unuk Mencari Sebuah Jalur

Jalur yang dicari	DFS (μ s)	A* (μ s)
0 ke 1	42,23	26,17
0 ke 2	44,10	48,07
0 ke 3	44,47	71,30
0 ke 4	47,60	91,77

Dari hasil pengujian dapat dilihat bahwa kedua algoritma memiliki waktu yang berbeda. Algoritma DFS memiliki waktu yang relatif konstan apabila dibandingkan dengan algoritma A*. Pada percobaan ini waktu yang dibutuhkan oleh algoritma DFS terlihat lebih cepat 4μ s sampai 44μ s pada proses pencarian jalur dari 0 ke 2, 0 ke 3, dan 0 ke 4, karena peta yang digunakan melingkar dan tidak banyak bercabang. Sedangkan algoritma A* yang merupakan pengembangan algoritma BFS lebih cepat ketika titik yang dituju lebih dekat dengan titik awal (titik 0 ke titik 1). Dari hasil percobaan meski terdapat selisih waktu antar kedua algoritma, kedua algoritma masih berjalan kurang dari seratus mikro detik sehingga tidak memberikan perbedaan pada sistem ketika digunakan. Tetapi karena pada prosesnya algoritma DFS menggunakan memori yang lebih sedikit maka sistem ini menggunakan algoritma DFS.

3.2.2. Pengujian Robot untuk Mencapai Tujuan

Pada pengujian ini robot diuji pada gedung C FTEK UKSW dengan titik tumpuan seperti Gambar 5. Beberapa percobaan yang dilakukan antara lain pengujian pada jalur lurus (7 ke 0, 0 ke 1, 1 ke 2 dan lainnya) dan pengujian berbelok (0 ke 2, 0 ke 6 dan lainnya). Pengujian dilakukan dengan menjalankan robot pada lokasi dengan halangan berupa dinding. Perubahan lokasi dilakukan secara manual oleh operator dimana operator harus mengirimkan lokasi terbaru robot menggunakan kontrol manual.

Pada pengujian pada jalur lurus, robot dapat berjalan dengan baik tanpa menabrak. Pada pengujian pada jalur lurus juga diberikan sebuah halangan yang diletakkan di sisi kiri jalan dan robot masih dapat menghindarinya dan sistem masih dapat mengarahkan robot ke arah yang tepat ketika arah hadap robot telah melenceng hingga lebih dari 90 derajat (kesalahan sudut yang terjadi ketika robot berputar ke kiri atau kanan terlalu jauh untuk menghindari halangan).

Pengujian berbelok bertujuan untuk mengetahui apakah sistem dapat mengarahkan robot ke arah yang benar ketika menemui persimpangan. Pada percobaan ini juga diperiksa apakah robot dapat memutar tanpa menabrak. Hasil percobaan yang didapat robot menabrak tiga dari lima pengujian ketika ingin berbelok, dikarenakan ketika perintah rem diberikan robot masih tetap melaju mendekati dinding (ketika berjalan mengarah ke dinding) sehingga tidak menyisakan ruang untuk berputar. Setelah dilakukan pengujian beberapa kali untuk mengurangi momentum ketika berhenti maka didapatkan kecepatan ± 6 km/jam robot masih menyisakan ruang untuk berputar dan mengurangi kemungkinan menabrak ketika berputar. Selain itu didapat bahwa sistem sudah dapat mengarahkan robot untuk memutar ke arah yang tepat.

3.2.3. Pengujian Mencari Jalur Alternatif

Pengujian ini dilakukan dengan cara memberikan perintah berbelok dari titik 2 ke titik 0 (atau 0 ke 6) dan tidak memberikan pembaruan lokasi pada titik 1 sehingga robot berjalan menghadap dinding. Pengujian dikatakan berhasil jika robot tidak menabrak dinding pada titik 1 dan memutar balik menuju titik 2. Pada titik 2 hanya diperiksa jalur yang akan dilalui robot tidak menjalankan robot untuk melalui jalur alternatif karena jalur alternatif yang ada harus memutar gedung C.

Percobaan dilakukan sebanyak 10 dengan nilai kecepatan ± 6 km/jam. Hasil yang diperoleh sistem dapat memutar pada setiap percobaan tanpa menabrak dinding didepannya meski harus memperbaiki posisi beberapa kali sebelum berjalan menuju titik sebelumnya dan setelah memutar jalur yang ditampilkan pada layar sistem menunjukkan rute alternatif yang memungkinkan. Oleh karena itu sistem untuk mencari jalur alternatif dikatakan berhasil.

3.3. Pengujian Simple Reflex Agent

Pada bagian ini akan diuji kemampuan dari simple reflex agent dalam menghindari halangan. Halangan yang akan diuji terdiri dari dua jenis yaitu manusia dan halangan berupa benda. Kedua jenis halangan akan diuji ketika berada di bagian tengah, dan bagian kiri atau kanan.

3.3.1. Pengujian Halangan Berupa Manusia

Pengujian pertama dilakukan ketika robot diam dan di bagian yang telah ditentukan (bagian samping dan tengah sensor), berdiri seseorang untuk menghalangi robot dan kemudian diperiksa apakah sistem dapat mendeteksi hal tersebut sebagai halangan. Jarak yang diambil untuk percobaan yaitu 1 m (karena objek yang dianggap halangan ketika jarak kurang dari 1,3 m). Hasil percobaan dapat dilihat pada Gambar 9.

Pada percobaan, kinect tidak dapat mendeteksi apakah halangan yang ada didepannya merupakan manusia atau benda, karena bagian yang terlihat ketika ada manusia hanya bagian bawah tubuhnya. Untuk dapat mendeteksi manusia di depannya kinect membutuhkan bagian pinggang ke atas, tetapi pada sistem ini kinect diarahkan 15 derajat ke bawah sehingga bagian tubuh bagian atas seseorang tidak tertangkap kamera kinect. Meski tidak dapat membedakan halangan berupa manusia ataupun benda, sistem masih mampu menghindari manusia.



Gambar 6. (a). Pengujian Halangan Berupa Manusia pada Bagian Tengah. (b). Pengujian Halangan Berupa Manusia pada Bagian Kiri Sensor.

Pada Gambar 6.a dapat dilihat bahwa bagian yang diperiksa hanya bagian kaki bagian bawah, untuk perintah yang dihasilkan perintah tetap menganggap hal tersebut sebagai halangan, tetapi ketika orang tersebut mundur mendekati jarak 1,3 m sistem tidak mampu mendeteksi bagian tersebut sebagai halangan karena bagian yang dideteksi terlalu sedikit. Untuk mengatasi masalah tersebut batas atas pada bagian tengah dinaikkan sehingga ketika jarak mendekati 1,3 m sistem masih dapat mendeteksi bagian atas kaki sebagai tambahan area yang dideteksi. Sedangkan untuk bagian kiri seperti gambar 6.b bagian yang terbaca sudah cukup untuk dianggap sebagai halangan.

4.3.2. Pengujian Halangan Berupa Benda yang Disusun Zig - Zag

Percobaan kedua yang dilakukan terhadap benda tidak bergerak. Enam buah benda diletakkan pada bagian kiri dan kanan jalan secara zig – zag dengan jarak antar halangan sejauh 1,5 m, dengan luas jalan 2,4 m, dan tinggi benda yang diletakkan ± 45 cm. Dalam percobaan robot dibiarkan jalan secara otomatis 10 kali ke arah rintangan yang telah dibuat.

Hasil yang diperoleh robot hanya dapat menghindari dengan baik 6 kali dari 10 kali percobaan yang dilakukan. Empat pengujian lainnya, robot masih menabrak satu atau lebih rintangan yang ada. Dari keempat pengujian yang gagal dapat diketahui dua kondisi yang menyebabkan robot menabrak yaitu kondisi ketika robot hendak berhenti dan mengganti arah tetapi momentum yang dimiliki robot menyebabkan robot berhenti terlalu dekat dengan halangan sehingga tidak terlihat pada bagian yang diperiksa atau bahkan tidak terlihat oleh kinect dan kondisi kedua yaitu kondisi ketika robot sudah menghindar tetapi bagian paling luar robot (kerangka roda) masih menabrak halangan.

4.3.3. Percobaan Dengan Benda yang Memiliki Ketinggian Berbeda

Percobaan ini dilakukan untuk mengetahui tinggi minimum yang dibutuhkan oleh sistem agar dapat terdeteksi sebagai halangan. Sebuah benda dengan tinggi tertentu akan diletakkan dengan jarak 1,3 m atau kurang dari robot kemudian akan dilihat apakah sistem mengenali benda tersebut sebagai halangan atau tidak. Pengujian ini dilakukan pada bagian tengah dan bagian samping kiri sensor. Bagian kanan sensor tidak diuji karena dianggap memiliki sifat yang sama dengan bagian kiri. Hasil pengujian dapat dilihat pada Tabel 4.

Tabel 4. Hasil Pengujian Benda dengan Tinggi Tertentu pada Bagian Tengah (kiri) dan pada Bagian Samping (kanan)

Tinggi (cm)	Jarak (m)	Terdeteksi halangan	Tinggi (cm)	Jarak (m)	Terdeteksi halangan
10	1,3	Tidak	10	1,3	Tidak
10	0,9	Ya	26	1,3	Tidak
23	1,2	Ya	33	1,3	Ya
30	1,3	Ya	33	1	Ya
23	0,9	Ya	50	0,60	Ya

Dari tabel 4 pada jarak 1,3 m sistem masih belum dapat mendeteksi halangan dengan tinggi 10 cm dan 23 cm tetapi keduanya masih dapat terdeteksi sebagai halangan dengan jarak yang lebih dekat, untuk tinggi 10 cm dibutuhkan jarak 0,9 dan untuk tinggi 23 cm dibutuhkan jarak 1,2 m. Dari data tersebut dapat dilihat bahwa sistem dapat mendeteksi halangan yang semakin pendek pada jarak yang semakin dekat. Tetapi pada saat mencoba ketinggian 3 cm sistem tidak mampu mendeteksi benda tersebut sebagai halangan, sistem baru mendeteksi ketika tinggi benda 6 cm, sehingga dapat dikatakan tinggi minimum benda yang dapat dideteksi sebagai halangan adalah 6 cm.

Untuk bagian samping, tinggi yang dibutuhkan untuk dianggap sebagai halangan jauh lebih tinggi dibandingkan pada bagian tengah yaitu 33 cm. Hal ini disebabkan karena bagian samping yang diperiksa memiliki sumbu y yang lebih kecil (semakin ke atas) sehingga benda harus lebih tinggi untuk dapat masuk pada bagian yang diperiksa. Jika dijalankan benda yang dapat dideteksi pada bagian tengah seringkali tidak dideteksi kembali pada bagian samping dan menyebabkan benda tertabrak. Sehingga tinggi minimum benda yang dibutuhkan oleh sistem agar dapat dihindari oleh sistem adalah 50 cm.

4. Kesimpulan

Dari hasil percobaan dapat disimpulkan bahwa sistem yang dibuat dapat menggerakkan robot secara otomatis dengan beberapa pembatasan yaitu sistem tidak dapat berjalan pada lingkungan yang terpapar sinar matahari secara langsung, tinggi maksimal benda agar dapat dihindari adalah 50 cm dan kecepatan maksimal robot agar dapat menghindari benda adalah 6 km/jam. Sistem tidak dapat berjalan pada lingkungan yang terpapar sinar matahari secara langsung karena kinect tidak dapat mendeteksi jarak pada lingkungan yang terpapar sinar matahari. Kecepatan 6 km/jam masih dapat ditingkatkan jika robot dilengkapi dengan sistem pengereman untuk mengurangi momentum ketika berhenti.

Daftar Pustaka

- [1] S.J. Russell, P. Norvig. "Artificial Intelligence: A Modern Approach". Prentice Hall. 2009.
- [2] K.A. Kurniawan. 2014. *Sistem Kontrol Wireless dan Realtime Video streaming pada Robot Pengangkut Barang*. Tesis. Salatiga: Fakultas Teknik Elektronika dan Komputer, Universitas Kristen Satya Wacana. 2014.