SIMULASI *AUTONOMOUS VEHICLE* DI UNIVERSITAS KRISTEN SATYA WACANA SALATIGA

Sandro Angkat[1], Darmawan Utomo[2], Hartanto K. Wardana[3]

[1]Program Studi Teknik Elektro, Fakultas Teknik – UKSW

[2,3]Program Studi Sistem Komputer, Fakultas Teknik – UKSW

[1]sandroangkat@gmail.com, [2]darmawan@staff.uksw.edu, [3]hkwardana@yahoo.com

INTISARI

Fakultas Teknik Jurusan Teknik Elektro Universitas Kristen Satya Wacana (UKSW) Salatiga sedang merancang *autonomous vehicle* (robot mobil) yang berfungsi untuk menghantarkan beberapa barang antar gedung perkuliahan seperti dokumen. Untuk itulah perlu dirancang sebuah simulasi untuk memberikan gambaran mengenai pergerakan dan karakteristik dari robot tersebut sesuai dengan algoritma yang dirancang.

Perancangan dilakukan dengan menggunakan Netbeans IDE 7.1, JDK 1.7 dan dengan menggunakan *engine* GTGE. Perancangan juga menggunakan algoritma *if-then rules* yang dibuat berdasarkan pola pikir manusia (*human behaviour based*) dalam hal ini adalah operator yang merancang algoritma.

Simulasi telah berhasil dirancang dengan tampilan dua dimensi dimana terdapat tiga objek robot, satu objek motor, satu objek mobil, tujuh objek manusia, 17 objek gedung dan dua objek lapangan. Ada juga menu masukan yaitu delapan DIRECTION, Posisi Awal gedung dan Posisi Akhir gedung. Dari hasil pengujian algoritma yang dirancang sudah berhasil membuat robot melakukan kemampuannya seperti bergerak secara otomatis menuju posisi akhir, memberi peringatan kepada objek-objek yang berada pada jalur robot dan menghindar dari objek penghalang sehingga tidak terjadi tubrukan. Ada dua kelemahan dari algoritma yaitu jika robot dikepung dari tiga sisi (depan, kanan dan kiri) akan membuat robot sulit bergerak, untuk itulah robot juga diberikan sistem kendali manual sehingga masalah tersebut dapat teratasi, kelemahan kedua yaitu pada skenario tertentu jarak terpendek yang digunakan oleh robot masih kurang optimal.

Kata Kunci : *autonomous, vehicle,* robot, simulasi

1. LATAR BELAKANG

Robot dalam segala bentuk dan fungsinya adalah salah satu dari kemajuan teknologi yang banyak membantu aktifitas manusia Penggunaannya mencakup bidang militer, medis sampai rumah tangga. Salah satu bentuk pengembangannya adalah robot mobil. Untuk dapat menjalankan tugasnya maka robot mobil yang dirancang haruslah mampu melakukan hal-hal sebagai berikut [1, h.213] :

1. Melakukan pergerakan secara *autonomous* (tanpa dikendalikan operator) dengan sistem navigasi yang dimilikinya.
2. Mencari atau mendeteksi objek atau benda yang menjadi bagian dari tugasnya.
3. Melakukan tindakan terhadap objek sesuai dengan fungsi dari robot tersebut.

Navigasi sebagai salah satu kemampuan dasar untuk robot tersebut telah menjadi topik yang diangkat dalam penelitian robotika. Walaupun ada kemungkinan untuk merancang sistem navigasi robot *autonomous* tanpa sistem kecerdasan buatan, namun robot jenis ini mensyaratkan adanya kondisi tertentu pada lingkungannya agar dapat beroperasi, misalnya saja robot yang mengikuti garis (*line follower*). Jadi sistem kecerdasan buatan adalah suatu hal yang mutlak ada dalam robot yang akan digunakan pada suatu daerah yang belum direkayasa untuk penggunaan robot mobil.

Berbagai penelitian sebelumnya menunjukkan bahwa sistem kecerdasan buatan dapat diwujudkan dengan berbagai macam algoritma [2, h.25-31] seperti algoritma genetik, *Artificial Potencial Field*, logika *Fuzzy*, *Roadmap Methodology* dan yang lainnya. Kendali logika *Fuzzy* bekerja berdasarkan aturan linguistik yang dapat dibuat mirip dengan seorang operator ahli dalam melakukan proses kendali [3, h.2]. Dengan pendekatan algoritma tersebut dibuatlah aturan-aturan tertentu (*if-then rules*) yang dirancang sedemikian rupa sehingga robot mobil tersebut dapat berpikir seperti seorang manusia dalam proses pengambilan keputusan (*human behavior based*) dalam hal ini adalah operator yang merancang algoritma tersebut. Pengambilan keputusan yang dimaksud adalah perilaku robot dalam melakukan kemampuan-kemampuannya seperti yang telah disebutkan sebelumnya.

2. KAJIAN PUSTAKA

Di dalam proses perancangan digunakan *engine* GTGE karena simulasi yang dirancang memiliki kesamaan dengan konsep sebuah permainan (*game*). Pada proses perancangan juga digunakan *if-then rules* sebagai algoritma untuk mencari jarak terdekat.

2.1. GTGE

GTGE merupakan *engine* yang mampu membuat sebuah permainan (*game*) berbasis bahasa pemrograman *Java* [4,h.15]. Dalam bahasa pemograman *Java*, setiap fungsi berada di dalam kelas (*class*), demikian juga dengan fungsi-fungsi GTGE. Untuk memudahkan pencarian dan penggunaan fungsi dari suatu kelas, *Java* mendukung pengelompokan kelas ke dalam suatu paket (*package*). GTGE memanfaatkan hal ini seoptimal mungkin untuk membuat GTGE *Application Programming Interface* terstruktur dengan baik dan memudahkan pencarian serta penggunaan fungsi GTGE [4, h.19].

Local Path Planning

Ada 2 jenis sistem navigasi yang biasa digunakan pada robot mobil [2], yaitu:

a. *Global Path Planning*

Pada metode ini robot telah memiliki informasi tentang keseluruhan lingkungan termasuk objek-objek diam yang berada pada lingkungan tersebut. Pada pendekatan ini robot telah memiliki informasi lengkap jalur yang akan dilewati bahkan sebelum dia bergerak.

b. *Local Path Planning*

Sedangkan pada metode robot tidak mengetahui informasi mengenai lingkungan dimana dia ditempatkan namun dengan pendekatan ini robot dapat membuat sendiri jalur yang baru untuk bisa ke tempat tujuan.

Pada perancangan ini digunakan metode yang kedua karena pada pendekatan yang ke dua robot akan dapat beradaptasi dengan cepat dan mudah dengan perubahan lingkungan.

2.2. *If-Then Rules*

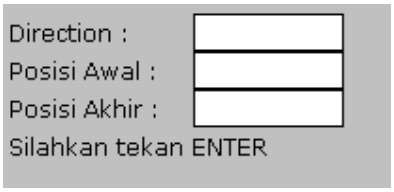
Pada perancangan selain *engine* GTGE, juga terdapat algoritma sebagai komponen pendukung dan dasar pergerakan dari robot mobil. Algoritma yang digunakan adalah *if-then* rules. Algoritma *if-then rules* dibuat berdasarkan pola pikir

manusia (*human behaviour based*) dalam hal ini adalah operator yang merancang algoritma. Fungsi utama dari algoritma adalah untuk mencari jarak terdekat untuk pergerakan robot dari posisi awal menuju posisi tujuan.

3. PERANCANGAN

3.1. Menu Input

Sebelum robot muncul maka terlebih dahulu pengguna (*user*) harus mengisi menu yang telah tersedia, untuk tampilannya dapat dilihat pada Gambar 1.



Gambar 1. Tampilan Menu

Ada tiga data penting yang wajib diisi pada menu tersebut, yaitu DIRECTION, POSISI AWAL dan POSISI AKHIR. Data yang dimasukkan berupa string.

3.2. Latar Belakang Simulasi

Semua gedung utama dan beberapa tempat penting lainnya yang berada dalam ruang lingkup UKSW disertakan ke dalam simulasi ini untuk menggambarkan kondisi UKSW. Untuk lebih jelasnya dapat dilihat pada Gambar 2.

Gambar 2. Latar belakang simulasi

Gedung-gedung tersebut merupakan berkas gambar yang digunakan sebagai *sprite* dan diatur letaknya berdasarkan koordinat (x,y).

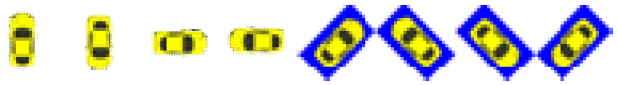
3.3. Robot Mobil/ *Agent*

Robot mobil atau yang bisa disebut juga dengan *agent* merupakan objek utama dalam simulasi ini. Pada perancangan simulasi ini robot memiliki tiga parameter, untuk lebih jelasnya dapat dilihat pada Tabel 1.

Tabel 1. Parameter pada robot

Parameter	Keterangan
State	DIAM, LAMBAT, CEPAT
Direction	ATAS, BAWAH, KANAN, KIRI, SERONG_KANAN_ATAS, SERONG_KIRI_ATAS, SERONG_KANAN_BAWAH dan SERONG_KIRI_BAWAH
Movement	MAJU, MUNDUR, KEKANAN, KEKIRI, SERONGKANAN dan SERONGKIRI.

Untuk lebih jelas tentang *direction* robot dapat dilihat pada Gambar 3.



Gambar 3. Objek robot dengan DIRECTION yang berbeda

Dari Gambar 3 terlihat bahwa dengan *direction* yang berbeda akan memberikan bentuk objek yang berbeda juga.

3.4. Kemampuan Robot

Untuk dapat melakukan tugasnya dengan baik pada simulasi ini, robot dilengkapi dengan beberapa kemampuan pendukung. Kemampuan-kemampuan tersebut adalah sebagai berikut :

1. Mengatur posisi awal dan posisi akhir robot
2. Mendefinisikan posisi akhir
3. Bergerak secara *autonomous* menuju posisi akhir
4. *Obstacle Detection* (sensor)

5. *Collision Warning*
6. *Collision Avoidance*

3.5. Objek Pendukung

Untuk memaksimalkan simulasi ditambahkan tiga jenis objek pendukung seperti yang terlihat pada Tabel 2.

Tabel 2. Daftar objek bergerak

Manusia	Mobil	Motor
		

Objek-objek tersebut ditambahkan untuk melihat bagaimana respon robot terhadap objek bergerak.

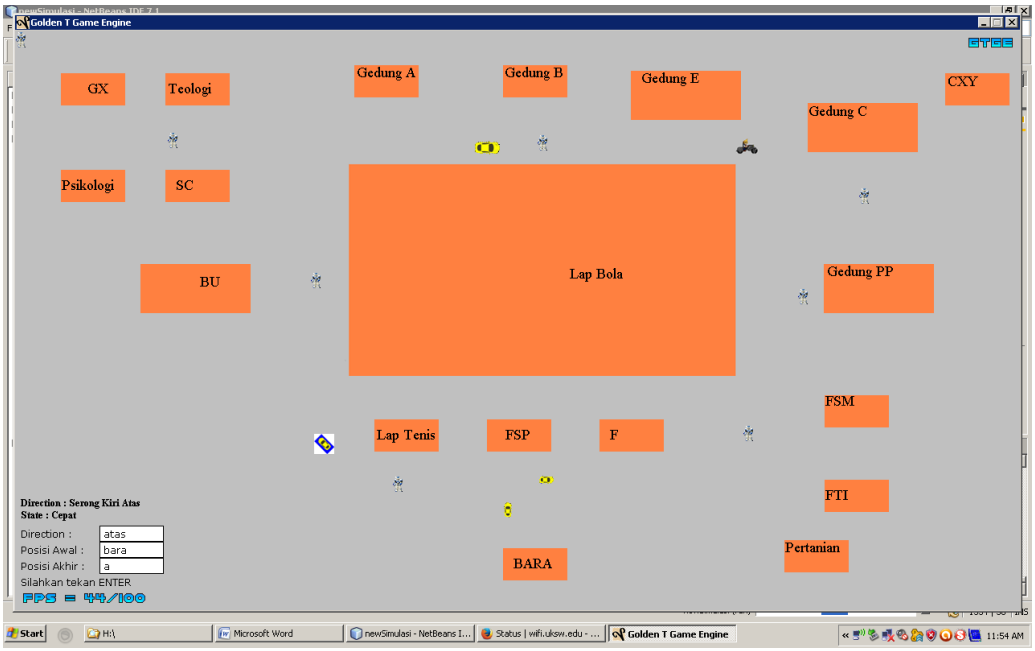
4. HASIL PENGUJIAN

4.1. Pengujian

Untuk mengukur tingkat keberhasilan simulasi dilakukan beberapa pengujian untuk melihat apakah hasil perancangan sudah sesuai dengan spesifikasi yang telah disusun.

4.1.1 Pengujian Jumlah Objek

Pengujian ini dilakukan untuk melihat jumlah maksimal dari objek apakah sudah sesuai dengan spesifikasi awal. Dari hasil pengujian didapat jumlah maksimal objek yang terdapat pada simulasi adalah tiga objek robot, tujuh objek manusia, satu objek motor, satu objek mobil dan 19 objek berupa gedung atau lapangan. Untuk hasilnya dapat dilihat pada Gambar 4.



Gambar 4. Pengujian jumlah objek

4.1.2 Pengujian Posisi Awal dan *Direction* Robot

Tujuan dari pengujian ini adalah untuk melihat apakah posisi awal dan *direction* awal dari robot sudah sesuai dengan masukan dari pengguna pada menu awal. Contoh hasil pengujian dapat dilihat pada Gambar 5.

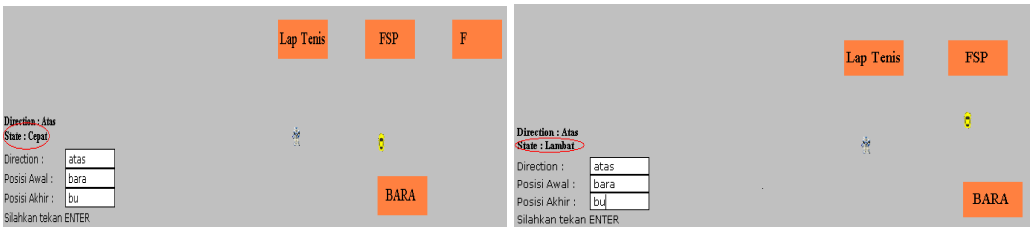
Gambar 5. Posisi dan DIRECTION awal robot

Dari Gambar 5 dapat dilihat bahwa robot telah berhasil muncul pertama kali di gedung awal dan bergerak dengan DIRECTION awal sesuai dengan masukan dari pengguna. Dengan menggunakan getX() dan getY() pada GTGE didapatlah koordinat dari gedung awal yaitu di kiri atas dari gedung tersebut.

4.1.3 Pengujian Perubahan *State* dari Robot

Pengujian ini bertujuan untuk melihat apakah robot sudah dapat mengubah kecepatan / s*tate* yang secara otomatis melambat saat mendekati objek sekitarnya.

Pada Gambar 6 diperlihatkan *state* dari CEPAT menjadi LAMBAT saat robot mendekati gedung FSP.

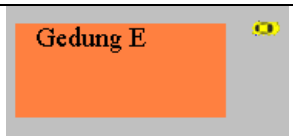


Gambar 6. Pengujian *state* dari CEPAT ke LAMBAT

4.1.4 Pengujian *Movement* Robot

Pengujian dilakukan untuk melihat apakah pergerakan robot sudah sesuai dengan aturan-aturan yang telah ditentukan atau belum. Untuk hasilnya dapat dilihat di Tabel 3.

Tabel 3. Hasil pengujian *movement* untuk contoh *direction* ATAS.

 MAJU	 KEKIRI	 KEKANAN	 MUNDUR – KENDALI MANUAL
Berhasil	Berhasil	Berhasil	Berhasil

4.1.5 Pengujian Sensor, *Collision Warning* dan *Obstacle Avoidance*

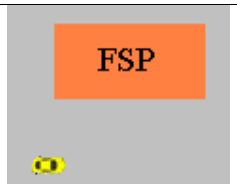
Robot akan mengeluarkan suara tertentu sebagai bentuk peringatan apabila terdapat objek yang mendekati robot supaya objek tersebut dapat menghindar dan tidak terjadi tubrukan. Ada dua jenis suara yang dikeluarkan. Untuk hasil pengujiannya dapat dilihat pada Tabel 4.

Tabel 4. Contoh hasil pengujian *collision warning*

ATAS – BARA – BU	 Mode suara satu	 Mode suara dua

Mode suara satu apabila terdapat objek di dekat robot dan mode suara dua akan dikeluarkan apabila objek tersebut tidak bergerak dan jaraknya semakin dekat. Pengujian pada Tabel 4 dilakukan dengan posisi awal BARA, posisi akhir BU dan *direction* awal ATAS. Ketika peringatan dihiraukan maka robot akan memutuskan untuk menghindari objek tersebut. Contoh dari hasil pengujian dapat dilihat pada Tabel 5.

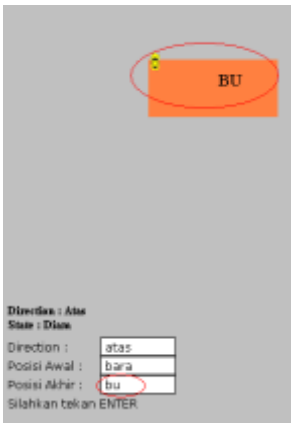
Tabel 5. *Obstacle avoidance*

Menghindari objek di atas (ATAS – BARA - BU)		

Dari Tabel 5 terlihat robot dapat mengindari Gedung FSP dengan membelok ke kanan.

4.1.6 Pengujian Robot Sampai di Posisi Akhir

Tujuan dari pengujian ini adalah untuk melihat apakah robot sudah dapat bergerak menuju posisi akhir sesuai dengan masukan pengguna secara otomatis. Untuk hasil percobaanya dapat dilihat pada Gambar 7.



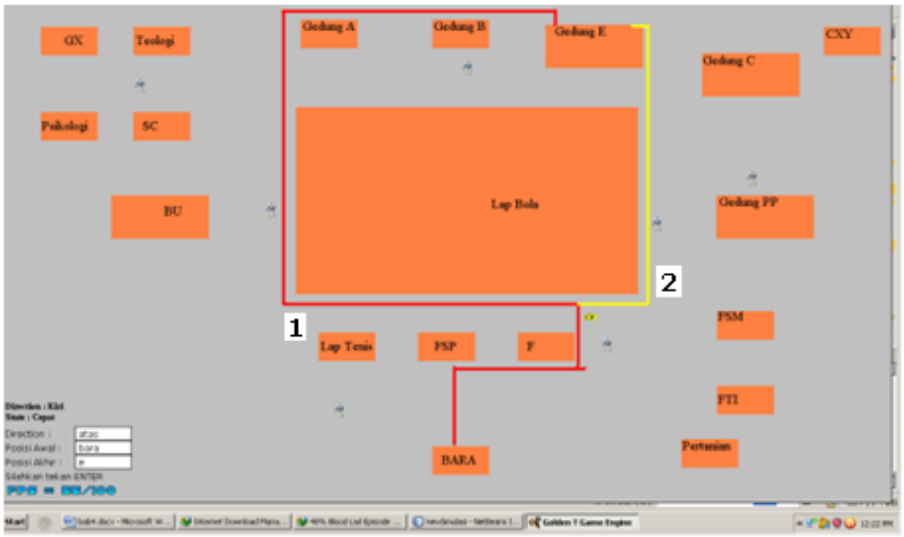
Gambar 7 Pengujian robot sampai di posisi akhir

Dari Gambar 7 dapat dilihat robot telah berhasil berhenti di posisi akhir yang dimasukkan oleh pengguna yaitu tepat di kiri atas dari gedung.

4.2. Kelemahan dari Algoritma

4.2.1 Jarak Terpendek yang Kurang Optimal

Dari beberapa skenario pengujian ditemukan kondisi dimana jarak terpendek yang diambil oleh robot kurang optimal yaitu Posisi Awal Bara dan Posisi Akhir Gedung E dan *direction* awal ATAS, untuk lebih jelasnya dapat dilihat pada Gambar 8.



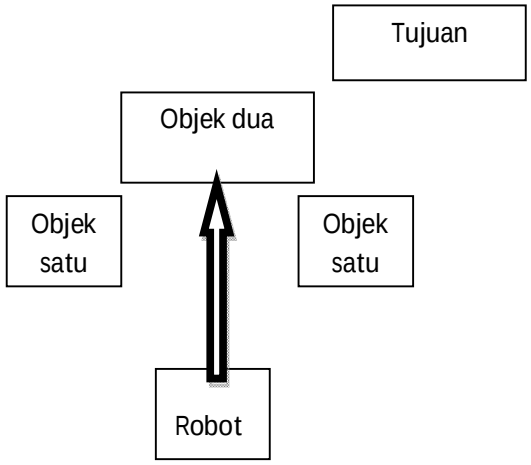
Gambar 8. Kondisi khusus satu

Dari Gambar 8 terlihat ada satu kondisi dimana robot menemukan penghalang yaitu Lapangan Bola. Jalur 1 merupakan jalur yang dilewati robot sesuai dengan algoritma

yang telah dirancang, sedangkan jalur 2 merupakan jalur terdekat menuju Gedung E. Jalur merah dipilih oleh robot karena ketika robot menemukan penghalang yaitu Lapangan Bola, definisi posisi akhir adalah KIRI_ATAS sehingga sesuai dengan algoritma yang dirancang ketika posisi akhir di KIRI_ATAS dan di depan terdapat penghalang maka robot akan bergerak ke kiri.

4.2.2 Robot Dikelilingi oleh Tiga Penghalang dari Arah yang Berbeda

Arah yang dimaksud adalah depan, kiri dan kanan. Robot dikurung sedemikian rupa sehingga tidak dapat bergerak ke arah serong kiri atau serong kanan. Untuk lebih jelasnya dapat dilihat pada Gambar 9.



Gambar 9. Kondisi khusus dua

Dari Gambar 9 terlihat bahwa robot akan melewati jalur hitam. Objek satu bukan dideteksi sebagai penghalang karena tidak berada pada jalur yang akan dilewati oleh robot, tapi terlihat bahwa robot akan menemukan objek dua sebagai penghalang dan saat tersebut terdapat pula kedua objek satu yang menghalangi robot untuk bisa mencari jalur alternatif. Oleh karena itu pada simulasi ini juga ditambahkan fasilitas untuk menggerakkan robot secara manual. Pada kondisi manual, *if-then rules* tidak akan berfungsi karena kendali sepenuhnya dipegang oleh operator. Setelah robot berada pada jalur yang aman, kendali manual dapat dihilangkan dan robot kembali bergerak sesuai *if-then rules*.

5. KESIMPULAN

Pada simulasi yang direalisasikan memiliki tampilan dua dimensi yang terdiri dari tiga objek robot, satu objek motor, tujuh objek manusia, satu objek mobil, 17 objek gedung dan dua lapangan. Ada tiga menu input yaitu DIRECTION, Posisi Awal dan Posisi Akhir. Algoritma yang telah dirancang memberikan robot beberapa kemampuan sebagai berikut :

1. Robot dapat muncul tepat di posisi awal dan memiliki *direction* awal sesuai dengan masukan pengguna, memiliki tiga jenis *state*, memiliki enam jenis *movement*, mengubah *movement* secara otomatis, dan memiliki delapan jenis *direction*.
2. Robot dapat mengetahui posisi akhir dan mendefinisikan letaknya yang terdiri dari 10 jenis, enam jenis sensor, bergerak menuju posisi akhir secara otomatis, mendeteksi adanya objek yang berada di sekitarnya, memberi peringatan pada objek tersebut untuk mencegah terjadinya tubrukan dan bergerak menghindari objek tersebut apabila objek tersebut tetap menjadi penghalang.
3. Algoritma yang digunakan adalah dengan *if-then rules.*

DAFTAR PUSTAKA

[1] Widodo,N.S.2009. “Penerapan Multi-Kontroller Pada Model Robot Mobil Berbasis Logika Fuzi”. Telkomnika Vol. 7, No.3 Desember 2009 : 213-218.

[2] Prases. K, Dayal. R.2012. “Controlling the Motion of an Autonomous Mobile Robot Using Various Techniques.” Journal of Advance Mechanical Engineering [2013] 1 : 24-39 doi:10.7726./jame.2013.1003.

[3] Setiawan. I .Darjat. Septiaji.A. “Perancangan Robot Mobil Penjejak Dinding Koridor Menggunakan Kendali Logika *FUZZY*”

[4] Taru, Andi. 2010. Pemrogaman Game dengan Java dan GTGE. Yogyakarta : ANDI.