



TUGAS AKHIR - EE 184801

**RANCANG BANGUN MOBILE ROBOT PENGHISAP DEBU
DENGAN PENGOPTIMALAN WAKTU PEMBERSIHAN**

Muhammad Faris
NRP 0711134000008

Dosen Pembimbing
Dr. Ir. Djoko Purwanto, M.Eng
Ronny Mardiyanto, ST., MT., Ph.D.

DEPATERMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020



TUGAS AKHIR - EE 184801

**RANCANG BANGUN MOBILE ROBOT PENGHISAP DEBU
DENGAN PENGOPTIMALAN WAKTU PEMBERSIHAN**

Muhammad Faris
NRP 0711134000008

Dosen Pembimbing
Dr. Ir. Djoko Purwanto, M.Eng
Ronny Mardiyanto, ST., MT., Ph.D.

DEPATERMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020



FINAL PROJECT - EE 184801

***DESIGNING OF VACUUM CLEANER MOBILE ROBOT
WITH CLEANING TIME OPTIMATION***

Muhammad Faris
NRP 0711134000008

Supervisor(s)
Dr. Ir. Djoko Purwanto, M.Eng
Ronny Mardiyanto, ST., MT., Ph.D.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2020

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan tugas akhir saya dengan judul **“Rancang Bangun Mobile Robot Penghisap Debu Dengan Pengoptimalan Waktu Pembersihan”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2020

Muhammad Faris
NRP. 0711134000008

**RANCANG BANGUN *MOBILE ROBOT*
PENGHISAP DEBU DENGAN PENGOPTIMALAN
WAKTU PEMBERSIHAN**

TUGAS AKHIR

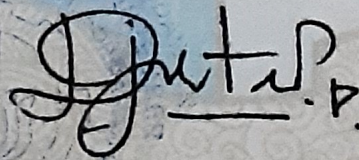
Diajukan Guna Memenuhi Sebagian Persyaratan Untuk
Memperoleh Gelar Sarjana Teknik

Pada

Bidang Studi Teknik Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro dan Informatika Cerdas

Menyetujui:

Dosen Pembimbing I



Djoko P.

Dr. Ir. Djoko Purwanto, M.Eng
NIP. 196512111990021002

**SURABAYA
JULI, 2020**

**RANCANG BANGUN *MOBILE ROBOT*
PENGHISAP DEBU DENGAN PENGOPTIMALAN
WAKTU PEMBERSIHAN**

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan Untuk
Memperoleh Gelar Sarjana Teknik

Pada

Bidang Studi Teknik Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro dan Informatika Cerdas

Menyetujui:

Dosen Pembimbing I



Dr. Ir. Djoko Purwanto, M.Eng
NIP. 196512111990021002

**SURABAYA
JULI, 2020**

**RANCANG BANGUN *MOBILE ROBOT*
PENGHISAP DEBU DENGAN PENGOPTIMALAN
WAKTU PEMBERSIHAN**

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan Untuk
Memperoleh Gelar Sarjana Teknik

Pada

Bidang Studi Teknik Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro dan Informatika Cerdas

Menyetujui:

Dosen Pembimbing II



Ronny Mardiyanto, S.T., M.T., Ph.D
NIP. 198101182003121003

**SURABAYA
JULI, 2020**

RANCANG BANGUN *MOBILE ROBOT* PENGHISAP DEBU DENGAN PENGOPTIMALAN WAKTU PEMBERSIHAN

Nama : **Muhammad Faris**
Pembimbing I : **Dr. Ir. Djoko Purwanto, M.Eng**
Pembimbing II : **Ronny Mardiyanto, S.T., M.T., Ph.D.**

ABSTRAK

Robot penghisap debu adalah jenis robot yang melakukan proses penyedotan menggunakan penghisap debu mini, meskipun bisa saja menggunakan *vacuum cleaner* biasa namun akan mengurangi nilai estetikanya.

Salah satu sensor yang umum digunakan pada robot penghisap debu adalah sensor jarak, yang terbagi dalam empat jenis: sensor jarak induktif, kapasitif, ultrasonic dan fotolistrik. Pada pengerjaan tugas akhir ini menggunakan sensor LiDAR untuk mendeteksi adanya halangan disekitar robot sehingga sistem akan memutuskan pergerakan robot kea rah mana yang lebih aman untuk jelajah robot nantinya. Selanjutnya adalah sensor IMU yang digunakan sebagai referensi *attitude* robot yang kemudian digabungkan dengan sistem navigasinya. Lalu seluruh data sensor akan dikirim ke pusat *processor* yang menggunakan Arduino Due 32 bit dan Raspberry Pi. Pemetaan dari hasil sensor menggunakan *genetic algorithm* berbasis *Robot Operating System* (ROS) untuk selanjutnya dilakukan pemetaan jalur.

Hasil dari pengujian yang dilakukan pada tugas akhir ini merupakan perbandingan waktu pembersihan pada luasan area 8 meter persegi. Dengan jumlah kromosom sebanyak 40 didapatkan hasil bahwa menggunakan *genetic algorithm* memiliki efisiensi waktu sebesar 3 menit dibandingkan dengan metode *random walk*.

Kata kunci: *mobile robot*, LiDAR, ROS, *genetic algorithm*, perencanaan jalur, Raspberry Pi.

halaman ini sengaja dikosongkan

DESIGNING OF VACUUM CLEANER MOBILE ROBOT WITH CLEANING TIME OPTIMATION

Name : Muhammad Faris
Advisor I : Dr. Ir. Djoko Purwanto, M.Eng
Advisor II : Ronny Mardiyanto, S.T., M.T., Ph.D.

ABSTRACT

Vacuum cleaner robot is a type of robot that performs a vacuum process using a mini vacuum cleaner, although it can use a regular vacuum cleaner but it will reduce its aesthetic value.

One of the sensors commonly used in vacuum cleaner robots is the proximity sensor, which is divided into four types: inductive, capacitive, ultrasonic and photoelectric proximity sensors. In this final project, the LiDAR sensor is used to detect obstacles around the robot so that the system will decide which robot is moving towards which is safer to explore the robot later. Next is the IMU sensor which is used as a reference to the attitude of the robot which is then combined with the navigation system. Then all sensor data will be sent to the central processor using Arduino Due 32 bit and Raspberry Pi. Mapping of the sensor results using a genetic algorithm based on the Robot Operating System (ROS) for further mapping pathways.

The results of the tests carried out in this thesis is a comparison of cleaning time in an area of 8 square meters. With the number of chromosomes as much as 40, the results show that using a genetic algorithm has a time efficiency of 3 minutes compared to the random walk method.

Keywords: mobile robot, LiDAR, ROS, genetic algorithm, path planning, Raspberry Pi.

halaman ini sengaja dikosongkan

KATA PENGANTAR

Alhamdulillah, penulis ucapkan kepada Allah SWT atas rahmat dan hidayah-Nya, penulis dapat menyelesaikan laporan tugas akhir ini dengan judul: **“Rancang Bangun Mobile Robot Penghisap Debu Dengan Pengoptimalan Waktu Pembersihan”** guna memenuhi syarat kelulusan pada Bidang Studi Teknik Sistem Pengaturan Departemen Teknik Elektro Fakultas Teknologi Elektro Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Penulis juga mengucapkan terimakasih kepada pihak-pihak yang berjasa dalam pengerjaan tugas akhir ini yaitu:

1. Allah SWT yang telah memberikan kelancaran dan keselamatan selama pengerjaan tugas akhir.
2. Kepada kedua orang tua penulis yang telah memberikan dukungan, semangat dan do'a tanpa henti demi keberhasilan penulis.
3. Bapak Dedet Candra Riawan, ST., M.Eng., Ph.D selaku Kepala Departemen Teknik Elektro ITS 2020-2025.
4. Bapak Dr. Ir. Djoko Purwanto, M.Eng sebagai dosen pembimbing I
5. Bapak Ronny Mardiyanto, S.T., M.T., Ph.D sebagai dosen pembimbing II

Laporan Tugas Akhir ini masih jauh dari sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran dari pembaca. Semoga buku laporan Tugas Akhir ini dapat memberikan manfaat bagi pembaca sebagai acuan penelitian selanjutnya.

Surabaya, 9 Juni 2020

Penulis

halaman ini sengaja dikosongkan

DAFTAR ISI

ABSTRAK	i
<i>ABSTRACT</i>	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Metodologi	3
1.6 Sistem Penulisan	4
1.7 Relevansi	5
SISTEM NAVIGASI <i>MOBILE ROBOT</i>	7
2.1 Mobile Robot	7
2.2 <i>Robotic Mapping</i>	8
2.3 <i>Simultaneous Localization and Mapping (SLAM)</i>	8
2.4 IMU	9
2.4.1 Akselerometer	11
SISTEMATIKA PERANCANGAN DAN PENGOLAHAN DATA	15
3.1 Identifikasi masalah	15
3.2 Perancangan Algoritma <i>Path Planning</i> berbasis GA	16
3.2.1 Inialisasi Populasi	17

3.2.2	Fitness Function dan Seleksi.....	18
3.2.3	Operator Genetika	18
3.2.4	Uji <i>Feasibility</i> Kromosom	19
3.2.5	<i>Elitism</i>	19
3.2.6	Alur Kerja <i>Path Planning</i> Statis	20
3.3	<i>Forward Kinematic</i> dari <i>Mobile Robot</i>	20
3.4	Perancangan Perangkat Keras.....	25
3.4.1	Komponen yang digunakan.....	26
3.5	Blok Diagram.....	28
HASIL DAN ANALISA		29
4.1	Perumusan Algoritma Genetika	29
4.2	Pengujian Sistem	30
4.3	Pengujian Gerak Robot pada Luasan Tertentu.....	33
PENUTUP.....		35
5.1	Kesimpulan	35
5.2	Saran	35
DAFTAR PUSTAKA.....		37
BIODATA PENULIS.....		39

DAFTAR GAMBAR

Gambar 2.1 (a) IMU gimbaled. (b) IMU strap-down	10
Gambar 2.2 Diagram block inertial measurement unit	10
Gambar 2.6 Model akselerometer dengan proff mass	11
Gambar 2.7 Akselerometer open loop	12
Gambar 2.8 Potongan dari akselerometer piezoresistive	12
Gambar 2.9 Kapasitif akselerometer	13
Gambar 2.10 (a) Desain dari akselerometer piezoelektrik. (b) FotoScanning Electron Microscopy dari elemen pendeteksi	13
Gambar 2.11 Multi axis akselerometer	14
Gambar 3.1, Posisi robot pada bidang cartesian	22
Gambar 3.2 Diagram posisi robot.....	23
Gambar 3.3 Mitsuyama MS-SR3	25
Gambar 3.4 Hasil desain pada Google Sketchup.....	25
Gambar 3.5 Raspberry Pi 4.....	26
Gambar 3.6 Arduino Mega 2560.....	26
Gambar 3.7 SZCM 25GA370.....	26
Gambar 3.8 VNH3ASP30 30 A	27
Gambar 3.9 LM2569	27
Gambar 3.10 Hasil dari realisasi perancangan <i>hardware</i>	27
Gambar 3.11 (a) Genetic Algortihm (b) Komponen mekanis .	28
Gambar 4.1 <i>Final Distance</i> 44 (Jumlah Kromosom100).....	32
Gambar 4.2. <i>Final Distance</i> 44 (Jumlah Kromosom220).....	32
Gambar 4.3 Arena uji coba pergerakan robot penghisap debu	33

halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 4.1 Representasi Kromosom.....	29
Tabel 4.2 Hasil Pengujian.....	31
Tabel 4.3 Hasil uji coba waktu pembersihan.....	33

halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Robot adalah perangkat yang dapat digunakan sebagai alat bantu manusia dalam menangani berbagai masalah yang telah ditemui sehingga manusia dapat menyesuaikan masalah tersebut secara efisien. Dalam melakukan tugasnya robot dapat bergerak secara manual yaitu dengan cara dikendalikan oleh manusia dari jarak tertentu guna menggantikan manusia dalam melaksanakan suatu tugas yang sulit dijangkau oleh manusia, maupun bergerak secara otomatis dengan cara menanamkan program yang dapat menyelesaikan suatu tugas pada robot^[1]. Salah satu alasan terciptanya robot untuk membantu pekerjaan manusia, bisa meringankan ataupun diambil alih sepenuhnya. Karena banyak pekerjaan manusia yang berbahaya ataupun sederhana akan tetapi sangat menguras tenaga.

Peneliti (King, 2001) mengemukakan bahwa mobile robot memiliki potensi besar dalam membantu kehidupan manusia yang berlaku pada masyarakat di masa depan. Fungsinya tidak akan lagi dibatasi untuk menyelesaikan tugas dalam perakitan dan manufaktur pada posisi tetap. Dalam rangka untuk menyelesaikan tugasnya sebagai mobile robot, maka mobile robot harus didesain dengan navigasi yang baik dan lancar untuk mengatasi perubahan medan dan lingkungan yang tak terduga.

Wheeled mobile robot (WMR) konvensional dibatasi dalam gerakan mobile robot karena tidak bisa bergerak kesamping tanpa manuver awal. Berbagai mekanisme telah dikembangkan untuk meningkatkan manuver WMR (King, 2002). Penelitian mendalam mengenai mobile robot terus mengalami kemajuan, hal ini didasari adanya inovasi dan ide kreatif dari para pengembang dan peneliti dibidang robotika. Contoh yang paling mendasar menurut Ribeiro (2002), adalah adanya RoboCup, yang merupakan kompetisi robot internasional yang didirikan pada tahun 1997. Tujuannya adalah untuk mengembangkan sepakbola robot otonom dengan tujuan untuk mempromosikan penelitian dan pendidikan di bidang inteligensi buatan, Hal ini menjadi salah satu tantangan ilmiah yang dibuat untuk mendorong penelitian dan pembangunan di bidang seperti *mobile autonomous robotic, automation, electronics, computer vision* dan

image processing. Dimana hal yang paling mendasar yang diperlukan dalam kompetisi RoboCup ini adalah bahwa sebuah *mobile robot* harus mampu bergerak ke segala arah atau bermanuver ke segala arah dengan seketika, Hal inilah yang mendorong banyak penelitian mengenai rancangan roda khusus yang lebih kompleks dalam hal pergerakan roda^[2].

Pengembangan *mobile robot* kali ini diterapkan pada *home appliance*, salah satunya adalah robot penghisap debu. Sudah banyak pengembangan pada robot *penghisap debu* dewasa ini, dengan berbagai metode untuk menjalankan tugasnya Pada beberapa penelitian terdahulu menggunakan *path planning* atau *3D mapping* akan tetapi dengan teknologi seberat itu sangatlah kompleks dan sulit untuk dilepas ke pasaran dikarenakan harganya tidak sebanding dengan kinerjanya. Oleh karena itu pada tugas akhir ini membuat rancang bangun *mobile robot* penghisap debu dengan menggunakan metode *genetic algorithm*, dimana perencanaan jalur dari robot ini akan berdasarkan pemetaan ruang dari hasil pindai sensor LiDAR.

1.2 Perumusan Masalah

Permasalahan yang dibahas dalam tugas akhir ini adalah:

1. Perancangan mekanik *mobile robot* untuk menjadi kerangka utama dari *mobile robot* penghisap debu.
2. Integrasi sensor terhadap mekanik *mobile robot* penghisap debu sebagai variabel pemetaan jalur.
3. Perancangan jalur *mobile robot* penghisap debu dengan pengoptimalan waktu pembersihan

1.3 Batasan Masalah

Batasan masalah dalam tugas akhir ini adalah:

1. Robot penghisap debu hanya bergerak pada bidang datar.
2. Objek yang dibersihkan ialah debu / pasir.
3. Memakai sistem pergerakan *differential mobile robot*.
4. Pergerakannya menggunakan prinsip non-holonomic.

1.4 Tujuan

Tujuan dari tugas akhir ini adalah sebagai berikut:

1. *Mobile robot* dapat berjalan bebas dan dapat menghindari dari halangan.
2. *Mobile robot* dapat menghisap debu atau pasir dengan baik.

3. Sistem kontrol dapat membuat jalur yang paling efektif dan efisien dari segi waktu pembersihan.

1.5 Metodologi

Langkah-langkah yang dikerjakan pada tugas akhir ini adalah sebagai berikut:

- **Studi Literatur**

Pada tahap ini dilakukan pengumpulan dasar teori yang menunjang dalam pembuatan Tugas Akhir ini. Dasar teori ini dapat diambil dari buku literatur, jurnal, artikel baik melalui media cetak ataupun melalui publikasi web.

- **Perancangan Perangkat Lunak**

Perancangan perangkat lunak dilakukan dengan pembuatan *source code* yang meliputi beberapa program untuk melakukan percobaan bertahap dari sistem yang ada. Sehingga apabila terjadi kesalahan dan memerlukan perbaikan (*troubleshooting*) maka akan mempermudah prosesnya. Beberapa contohnya adalah program baca sensor jarak, program baca sensor IMU, program baca sensor odometri, program kendali roda, dll. Kemudian setiap bagian program tersebut akan disatukan dalam program utama sehingga sistem akan terintegrasi dengan sempurna dalam pengoperasiannya. Tahapan yang pertama adalah sistem pemetaan (*mapping*), membaca *attitude* dari *mobile robot* lalu menentukan arah gerakan dari robot tersebut.

- **Perancangan Perangkat Keras**

Perancangan perangkat keras meliputi pembuatan badan robot. Robot yang dibuat merupakan robot non-holonomic dengan dua roda yang dikendalikan secara terpisah serta dua roda kecil sebagai penyangga. Robot ini menggunakan sensor LiDAR sebagai sensor jarak untuk pemetaan ruang dan sensor IMU sebagai monitoring *attitude* dari *mobile robot* tersebut. Aktuator yang dipakai adalah driver motor sebagai penggerak utama dan pengatur kecepatan roda serta sistem penghisap debu yang terpasang pada badan *mobile robot*.

- **Pengujian Sistem**

Pengujian sistem dilakukan untuk menguji keandalan dari sistem yang telah dirancang. Pengujian dilakukan untuk melihat apakah perangkat lunak dan perangkat keras yang telah dirancang dapat berfungsi secara optimal. Pengujian dapat dilakukan dengan beberapa tahap. Pertama adalah pengujian algoritma yang telah dibuat dengan pembuatan simulasi pemetaan ruang. Kedua adalah pengujian implementasi algoritma yang telah dibuat kedalam gerak robot yang sebenarnya.

- **Analisa**

Analisa dilakukan terhadap hasil dari pengujian sehingga dapat ditentukan karakteristik dari perangkat lunak dan perangkat keras yang telah dibuat. Apabila karakteristik dari algoritma yang telah disimulasikan dan diimplementasikan masih belum sesuai, maka perlu dilakukan perancangan ulang pada sistem dan kemudian diuji kembali.

- **Penyusunan Laporan Tugas Akhir**

Tahap penulisan laporan tugas akhir adalah tahapan terakhir dari proses pengerjaan tugas akhir ini. Laporan tugas akhir berisi seluruh hal yang berkaitan dengan tugas akhir yang telah dikerjakan yaitu meliputi pendahuluan, tinjauan pustaka dan teori penunjang, perancangan sistem, pengujian, dan penutup.

1.6 Sistem Penulisan

Dalam buku tugas akhir ini, pembahasan mengenai sistem yang dibuat terbagi menjadi lima bab dengan sistematika sebagai berikut:

- **BAB I : PENDAHULUAN**

Pada bagian ini menjelaskan beberapa sub bagian yang antara lain berisi Latar Belakang, Permasalahan, Tujuan, Penelitian pada tugas akhir ini bertujuan untuk merancang perencanaan gerak pada robot penghisap debu untuk mencapai titik target yang diinginkan. Serta metodologi, sistematika penulisan, dan relevansi penulisan tugas akhir ini.

- **BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI**

Pada bagian ini berisi mengenai landasan teori yang digunakan dalam pelaksanaan tugas akhir yang meliputi robot non-holonomic, pemetaan jalur dan *genetic algorithm*. Bagian ini memaparkan mengenai beberapa teori penunjang dan beberapa literatur yang berguna bagi pembuatan Tugas Akhir ini.

- **BAB III : PERANCANGAN SISTEM**

Bab ini menjelaskan tentang perencanaan sistem baik perangkat keras maupun perangkat lunak untuk sistem perencanaan pemetaan jalur robot penghisap debu.

- **BAB IV : PENGUJIAN**

Pada bagian ini akan menjelaskan hasil uji coba sistem beserta analisisnya.

- **BAB V : PENUTUP**

Bagian ini merupakan bagian akhir yang berisikan kesimpulan yang diperoleh dari pembuatan Tugas Akhir ini, serta saran-saran untuk pengembangannya

1.7 Relevansi

Hasil yang diharapkan dari tugas akhir ini ialah sebuah platform robot penghisap debu berbasis mobile robot yang dapat langsung diaplikasikan dalam kehidupan nyata sehingga dapat membantu tugas manusia serta platform ini akan menjadi platform open source bagi pengembangan selanjutnya karena dengan menggunakan sistem mobile robot holonomic berbasis three wheel omni bidirectional maka robot ini bisa diaplikasikan lebih kreatif dan bermanfaat kedepannya.

halaman ini sengaja dikosongkan

BAB II

SISTEM NAVIGASI *MOBILE ROBOT*

2.1 Mobile Robot

Mobile Robot adalah robot yang memiliki daya penggerak. *Mobile robot* biasanya dianggap sebagai subbidang robotika dan teknik informasi^[3].

Mobile robot memiliki kemampuan untuk bergerak di lingkungan mereka dan tidak terikat pada satu lokasi fisik. *Mobile robot* bersifat "otonom" (AMR - *autonomous mobile robot*) yang berarti mereka mampu menavigasi lingkungan yang tidak terkontrol tanpa memerlukan perangkat panduan fisik atau elektro-mekanis. Dalam kondisi tertentu, *Mobile robot* dapat mengandalkan perangkat panduan yang memungkinkan mereka melakukan rute perjalanan yang telah dinavigasi sebelumnya pada ruang yang relatif terkontrol (AGV - *autonomous guided vehicle*). Sebaliknya, robot industri biasanya kurang lebih bersifat statis, yang terdiri dari lengan bersendi (*multi-linked manipulator*) dan penggenggam rakitan (atau *end effector*), melekat pada permukaan yang tetap.

Mobile robot telah menjadi hal yang biasa dalam bidang komersial dan industri. Rumah sakit telah menggunakan AMR untuk memindahkan material selama bertahun-tahun. Gudang telah memasang sistem *mobile robot* untuk memindahkan material secara efisien dari rak penyimpanan ke zona pemenuhan pesanan. *Mobile robot* juga merupakan fokus utama penelitian saat ini dan hampir setiap universitas besar memiliki satu atau lebih laboratorium yang berfokus pada penelitian *mobile robot* ^[4]. *Mobile robot* juga ditemukan pada bidang industri, militer dan keamanan.

Komponen *mobile robot* adalah pengontrol, sensor, aktuator, dan sistem tenaga. Pengontrol umumnya mikroprosesor, mikrokontroler tertanam atau komputer pribadi (PC). Sensor yang digunakan tergantung pada persyaratan robot. Syaratnya bisa *dead reckoning*, pengindraan sentuhan dan jarak, triangulasi jarak, penghindaran tabrakan, lokalisasi posisi dan aplikasi spesifik lainnya ^[5]. Aktuator biasanya merujuk pada motor yang menggerakkan robot yang bisa beroda atau berkaki. Untuk memberi daya pada *mobile robot* biasanya kami menggunakan catu daya DC (yang merupakan baterai) alih-alih AC.

2.2 Robotic Mapping

Pemetaan robot adalah disiplin yang terkait dengan pengindraan komputer ^[6] dan kartografi. Tujuan dari robot otonom adalah untuk dapat membangun (atau menggunakan) pemetaan (penggunaan luar ruangan) atau denah lantai (penggunaan dalam ruangan) serta untuk lokalisasi diri dan basis pengisian baterai atau suar di dalamnya. Pemetaan robot adalah cabang yang berhubungan dengan studi dan penerapan kemampuan untuk lokalisasi diri dalam sebuah peta serta denah dan terkadang untuk membangun peta atau denah lantai oleh robot otonom.

2.3 Simultaneous Localization and Mapping (SLAM)

Dalam geometri komputasi, *simultaneous localization and mapping* (SLAM) adalah masalah komputasi untuk membangun atau memperbarui peta lingkungan yang tidak diketahui sambil secara bersamaan melacak lokasi agen di dalamnya. ^{[7] [8] [9] [10]} Meskipun permasalahan ini awalnya tampak dilematis, namun ada beberapa algoritma untuk menyelesaikannya, setidaknya mendekati, untuk jangka waktu yang dapat dilacak dalam lingkungan tertentu. Metode pendekatan yang populer termasuk filter partikel, filter Kalman lanjutan, persimpangan Covariance, dan GraphSLAM. Algoritma SLAM digunakan dalam navigasi, pemetaan robot, dan odometri untuk realitas virtual atau augmented reality.

Algoritma SLAM disesuaikan dengan sumber daya yang tersedia, karenanya tidak ditujukan untuk kesempurnaan, tetapi pada kepatuhan operasional. Pendekatan yang dipublikasikan digunakan dalam mobil yang dapat dikendarai sendiri, kendaraan udara tak berawak, kendaraan bawah air otonom, penjelajah planet, robot domestik yang lebih baru dan bahkan di dalam tubuh manusia. ^[11]

Diberikan serangkaian kontrol u_t dan pengamatan sensor σ_t atas langkah waktu diskrit t , masalah SLAM adalah untuk menghitung perkiraan keadaan agen x_t dan peta lingkungan m_t . Semua kuantitas biasanya probabilistik, jadi tujuannya adalah untuk menghitung:

$$P(m_{t+1}, x_{t+1} | \sigma_{1:t+1}, u_{1:t}) \quad (1)$$

Menerapkan aturan Bayes memberikan kerangka kerja untuk memperbarui posisi lokasi secara berurutan, mengingat peta dan fungsi transisi $P(x_t | x_{t-1})$

$$P(x_1|\sigma_{1:t}, u_{1:t}, m_t) = \sum_{m_{t-1}} P(\sigma_t|x_t, m_t, u_{1:t}) \sum_{x_{t-1}} \frac{P(x_t|x_{t-1})P(x_{t-1}|m_t, \sigma_{1:t-1}, u_{1:t})}{Z} \quad (2)$$

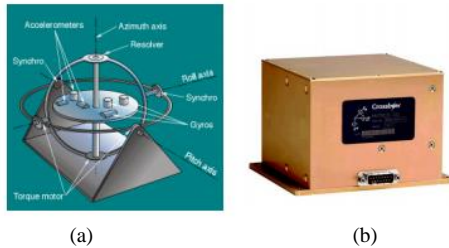
Demikian pula peta dapat diperbarui secara berurutan oleh

$$P(m_1|x_t, \sigma_{1:t}, u_{1:t}) = \sum_{x_t} \sum_{m_t} P(m_t|x_t, m_{t-1}, u_{1:t})P(m_{t-1}, x_t|\sigma_{1:t-1}, m_{t-1}, u_{1:t}) \quad (3)$$

Seperti banyak masalah inferensi, solusi untuk menyimpulkan dua variabel bersama-sama dapat ditemukan, untuk solusi optimal lokal, dengan bergantian pembaruan dari dua penyelesaian dalam bentuk algoritma EM.

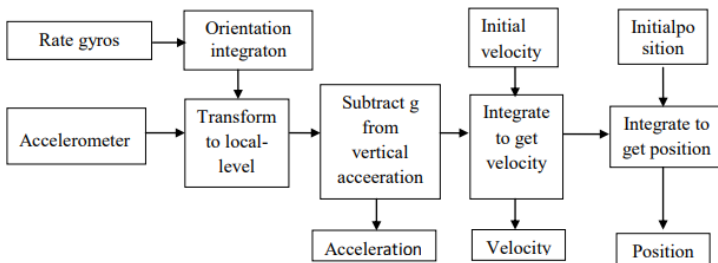
2.4 IMU

Inertial Measurement Unit (IMU) adalah alat yang memanfaatkan sistem pengukuran seperti gyroskop dan akselerometer untuk memperkirakan posisi relatif, kecepatan, dan akselerasi dari gerakan motor. IMU adalah bagian dari navigasi system yang dikenal sebagai Inertial Navigation System atau INS. Pertama kali didemonstrasikan oleh C.S. Draper tahun 1949, IMU menjadi komponen navigasi umum dari bidang dan kapal. Ada beberapa macam IMU yang biasa digunakan yaitu IMU gimbaled (Gambar 2.4(a)) dan IMUstrap-down (Gambar 2.4 (b)). IMU strap-down lebih umum dipakai saat ini. IMU mempertahankan 6-degree-of-freedom (DOF) yang memperkirakan gerakan yaitu posisi (X Y Z) dan orientasi (roll, pitch, yaw)[5]. Sistem seperti IMU hanya mempertahankan perhitungan terus menerus dari orientasi yang dikenal sebagai Attitude and Heading Reference System(AHRS) dan dipergunakan dalam cara yang sama sebagai IMU tetapi mempertahankan representasi tidak menyeluruh. Sebagai tambahan untuk mempertahankan sikap motor 9-DOF, komersial IMU juga secara khas mempertahankan perkiraan dari kecepatan dan akselerasi.



Gambar 2.1 (a) IMU gimbaled. (b) IMU strap-down

Perhitungan dasar dari IMU menggunakan tiga akselerometer orthogonal dan tiga gyroskop orthogonal. Data gyroskop (ω) diintegrasikan untuk mempertahankan perkiraan orientasi motor (θ) secara terus menerus. Di waktu yang bersamaan, tiga akselerometer digunakan untuk memperhitungkan akselerasi motor (a) dengan segera. Data ini kemudian ditransformasikan melalui perhitungan orientasi motor relatif terhadap gravitasi, sehingga vector gravity dapat dihitung lalu diekstrak dari pengukuran. Hasil akselerasi kemudian diintegrasikan untuk memperoleh kecepatan motor, kemudian diintegrasikan lagi untuk mendapatkan posisi (r)



Gambar 2.2 Diagram block inertial measurement unit

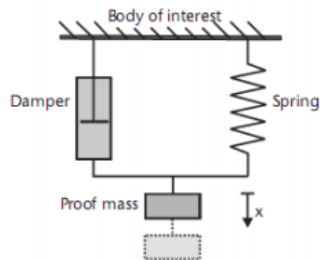
Inertial Measurement Unit (IMU) sangat sensitif untuk mengukur kesalahan di gyroskop dan akselerometer yang mendasar. Penyimpangan gyroskop mengarahkan pada kesalahan perhitungan dari orientasi motor, relatif terhadap gravitasi, menghasilkan kegagalan yang tidak tepat dari vektor gravitasi, seperti data akselerometer yang diintegrasikan dua kali, sisa vektor gravitasi akan menghasilkan kesalahan quadratic dalam posisi. Hampir tidak

mungkin untuk menghilangkan vektor gravitasi dan eror yang lain diintegrasikan dari waktu ke waktu sepenuhnya, karena penyimpangan merupakan isu pokok dari IMU.

2.4.1 Akselerometer

Sensor inertial akselerometer digunakan untuk mengukur akselerasi pada benda. Satu faktor penting mengenai akselerometer adalah bahwa akselerometer sensitif pada semua aksi paksaan eksternal termasuk gaya berat. Akselerometer menggunakan satu dari sejumlah mekanisme yang mengubah gaya menjadi sinyal yang dapat dibaca komputer.

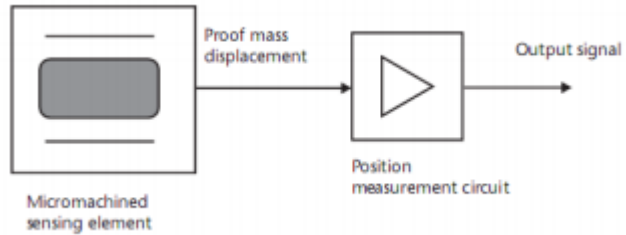
1. *Micromachined* akselerometer. Banyak jenis *micromachined* akselerometer yang telah dikembangkan dan pada umumnya element pendeteksi mekanis terdiri dari massa yang bebas bergantung pada suspensi mekanis terhadap reference frame seperti pada gambar 2.6^[7].



Gambar 2.6 Model akselerometer dengan proff mass

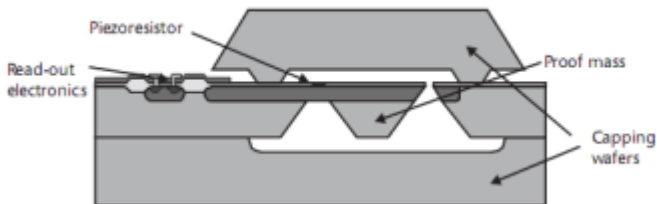
Faktor yang umum untuk semua akselerometer *micromachined* adalah perpindahan (*displacement*) dari massa bebas yang harus diukur dengan sirkuit alat ukur posisi dan dikonversi menjadi sinyal elektrik. Banyak tipe dari mekanisme pendeteksi telah diketahui seperti kapasitif, *piezoresistive*, *piezoelectric*, optis, dan *tunneling current*. Setiap mekanisme tersebut memiliki keunggulan dan kekurangan^[7]. Karakteristik dan performa dari akselerometer banyak dipengaruhi dari alat pengukur posisi dan kebutuhan utama dari alat tersebut. Karakteristik tersebut adalah *low noise*, linearitas yang tinggi, respon dinamik yang bagus, dan konsumsi daya rendah. Jika sinyal

output elektrik dari alat pengukuran posisi digunakan langsung sebagai sinyal dari Akselerometer ini disebut open loop akselerometer yang secara konsep ditunjukkan gambar 2.7. Akselerometer *micromachines* yang paling umum digunakan adalah openloop karena alat tersebut mudah digunakan dan murah.



Gambar 2.7 Akselerometer open loop

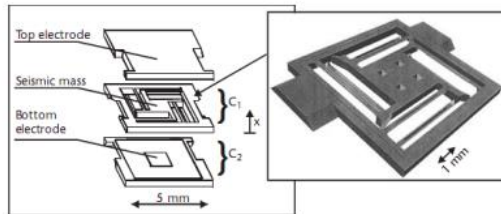
2. *Piezoresistive akselerometer micromachined* pertama menggunakan *bulk-micromachined sensing element*, terdiri dari silikon inti yang membentuk massa bebas dan *cantilever* sebagai sistem suspensi. Silikon *warper* dihubungkan (*bonded*) diantara dua *glasswafers* di mana rongga dietsa sehingga memperbolehkan massa bergerak terhadap reaksi terhadap akselerasi. Gerakan dari massa bebas dideteksi menggunakan *piezo* resistor. *Piezo* resistor dibuat darimenanamkan boron langsung pada batang (*beams*) dari suspensi silikon pada sistem massa bebas, seperti ditunjukkan pada gambar 2.8.



Gambar 2.8 Potongan dari akselerometer piezoresistive

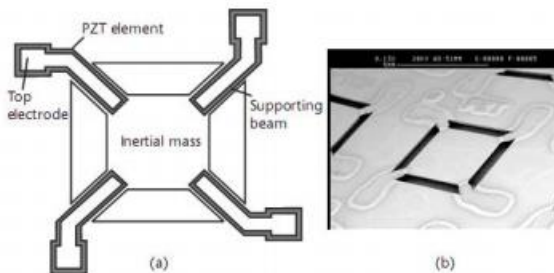
Pengukuran perpindahan (*displacement*) dari massa bebas secara kapasitif memiliki keuntungan bawaan terhadap pengambilan

sinyal *piezoresistive*. Kapasitif akselerometer memberikan sinyal output yang besar, respon keadaan tunak yang baik, dan sensitivitas yang lebih baik karena performa yang sedikit gangguan (*low noise*). Kelemahan utama adalah sensor kapasitif dapat terganggu karena medan elektromagnetis dari lingkungan, karena itu harus dilindungi dengan baik. Kelemahan yang lain adalah tidak dapat dihindarkan kapasitansi parasit pada input kepada alat amplifiier akan mendegradasi sinyal^[6]. Awal sensor kapasitif dibuat *assembly* beberapa wafer dengan teknik penggabungan (*bonding*). Cap wafer berada pada atas dan bawah yang menyediakan damping membentuk elektroda untuk deteksi kapasitif ditunjukkan pada gambar 2.9.



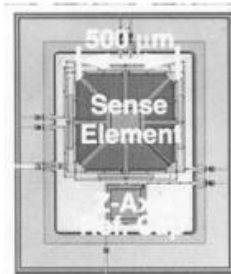
Gambar 2.9 Kapasitif akselerometer

Akselerometer makroskopik umumnya menggunakan *piezoelektrik* material untuk medeteksi massa bebas. Keuntungannya adalah *bandwith* yang lebih lebar dan sensor dapat mencapai puluhan kilohertz. Kelemahan utamanya adalah *piezoelektrik* tidak merespon kepada sinyal akselerasi statis dan frekuensi rendah karena kebocoran arus yang tidak dapat dihindari^[6]. Desain dari elemen pendeteksi dan *Scanning Electron Microscopy* (SEM) fotografi ditunjukkan dengan gambar 2.10.



Gambar 2.10 (a) Desain dari akselerometer piezoelektrik. (b) Foto Scanning Electron Microscopy dari elemen pendeteksi

Inovasi yang relatif baru untuk akselerometer *micromachined* adalah sensor yang mampu mengukur akselerasi pada dua atau tiga sumbu secara bersamaan. Walaupun mungkin untuk memasang tiga akselerometer sumbu tunggal yang tegak lurus terhadap masing-masing sumbu, akan tetapi akselerometer dua atau tiga sumbu memiliki keunggulan pada harga, ukuran, dan pemasangan inersial pada sumbu pendeteksi, karena elemen pendeteksi didefinisikan dengan akurat oleh metode *photolithographic*. Gambar 2.11 menunjukkan gambar multi *axis* akselerometer menggunakan *surface micromachined* dengan massa bebas satu buah yang bergerak pada tiga sumbu



Gambar 2.11 Multi axis akselerometer

BAB III

SISTEMATIKA PERANCANGAN DAN PENGOLAHAN DATA

3.1 Identifikasi masalah

Pada tugas akhir ini, permasalahan yang akan dibahas adalah permasalahan *path planning* yang bertujuan menemukan sebuah *path* terpendek antara dua titik yanerada dalam sebuah *map* yang merupakan representasi. *Path planning* sendiri merupakan salah satu permasalahan yang paling mendasar dalam permasalahan navigasi *mobile robot*. *Dynamic path planning* berbasis *genetic algorithm* akan diterapkan dalam tugas akhir ini untuk menyelesaikan permasalahan navigasi dalam lingkungan yang memiliki *moving obstacles*.

Map yang digunakan sebagai representasi area kerja robot memiliki *grid* yang seragam dengan ukuran 20x20 *grid*. *Grid* tersebut membatasi titik yang mungkin menjadi *waypoint*, sehingga *waypoint* selalu merupakan bilangan bulat. Titik awal gerak robot (*starting point*) diletakkan di ujung kiri bawah, sedangkan titik tujuannya (*end point*) diletakkan di ujung kanan atas. *Obstacle* yang digunakan selalu berbentuk lingkaran dengan radius yang seragam namun tersebar pada titik sebarang didalam *map*. Gambar 3.1 menggambarkan contoh konfigurasi *obstacle* yang mungkin terjadi. *Obstacle* dalam kasus ini melambangkan sebuah area yang tidak boleh dilewati oleh robot, dengan kata lain, koordinat tempat obstacle berada tidak mungkin terdaftar sebagai titik yang mungkin menjadi kandidat *waypoint*.

Akan tetapi, seperti yang telah dicantumkan dalam batasan masalah, dalam tugas ini tidak dibahas mengenai permasalahan estimasi tabrakan *obstacle* dinamis, dengan demikian, untuk menunjukkan kemampuan sistem untuk mencari *path* baru, tabrakan tersebut akan disimulasikan. Simulasi yang dimaksud adalah setelah mencapai *waypoint* pertama, akan dicari sebuah rute baru karena akan terjadi tabrakan. Setelah itu, rute baru tersebut akan langsung diterapkan. Hal ini menimbulkan masalah baru, dimana pencarian rute baru harus ditemukan dalam waktu yang cepat agar bisa ditemukan *waypoint* dari *path* yang bebas dari *obstacle*.

Mobile robot yang akan digunakan dalam penelitian ini adalah robot beroda, karena akan dioperasikan dalam sebuah bidang dua dimensi. Selain itu, robot beroda memiliki keunggulan dalam kemudahan untuk memodelkan dibandingkan *mobile robot* yang lain. Dalam dinamika kendaraan di dunia nyata, sebuah kendaraan perlu menggunakan sistem suspensi untuk mengkompensasi gerakan vertikal yang timbul setelah kendaraan bergerak dengan kecepatan yang cukup tinggi.

Robot beroda yang digunakan merupakan *differential drive robot*, yakni robot yang memiliki dua *drive wheels* yang bisa bergerak secara independen. Gerak roda yang independen tersebut memungkinkan robot untuk melakukan gerak melingkar (rotasi) dengan mengatur kecepatan roda sehingga roda berputar dengan arah berlawanan. Kemudian, berdasarkan model matematika *differential drive robot*, dirancang simulasi dan implementasi untuk membuat mengikuti (*path following*) *waypoint* yang telah ditemukan menggunakan *path planning* berbasis *genetic algorithm*.

3.2 Perancangan Algoritma *Path Planning* berbasis GA

Algoritma genetika (*Genetic algorithm*) merupakan teknik pencarian (*search technique*) yang meniru operator genetika natural. Langkah awal yang dilakukan untuk merancang algoritma *path planning* berbasis algoritma genetika ini diawali dengan inisialisasi populasi. Ukuran populasi yang dibangkitkan dalam proses pencarian *path* menentukan kecepatan program dan banyaknya kromosom. Setiap kromosom merupakan *path* yang mungkin menjadi jawaban dari permasalahan *path planning* (tidak menabrak *obstacle*). Oleh karena itu, diperlukan informasi mengenai letak *obstacle* secara global agar bias memastikan bahwa kromosom yang dibangkitkan telah memenuhi kriteria tersebut. Setelah selesai membangkitkan populasi awal hingga sesuai nilai populasi yang telah ditetapkan, setiap kromosom akan dievaluasi menggunakan *fitness function* untuk mengetahui kromosom yang memiliki jarak terpendek pada generasi tersebut. Setelah itu, dilakukan proses seleksi menggunakan seleksi *roulette wheel* untuk memilih kromosom yang akan dikawinsilangkan (*crossover*). Setelah dilakukan *crossover*, proses berikutnya adalah mutasi. Lalu, setelah diperoleh sebuah kromosom baru yang masih harus dicek terlebih dahulu apakah ia memenuhi

syarat bebas *obstacle*. Demikian penjelasan mengenai proses *path planning* dalam lingkungan yang statis.

Untuk permasalahan *path planning* dalam sebuah lingkungan dengan *obstacle dinamis*, timbul masalah baru dimana proses pencarian *path* baru harus dilakukan dengan waktu yang cukup cepat. Untuk mengatasi masalah tersebut, kromosom terbaik yang diperoleh untuk mencari *path* pada permasalahan *path planning* dalam lingkungan statis digunakan dalam proses pembangkitan populasi awal sehingga bisa memotong waktu komputasi. Alur Kerja *dynamic path planning* berbasis *genetic algorithm* terlampir pada subbab-subbab berikutnya.

3.2.1 Inisialisasi Populasi

Inisialisasi populasi adalah proses dimana kromosom yang mewakili *path* yang perlu dilalui untuk mencapai titik tujuan pada map yang telah didefinisikan. Di sebuah *map* yang memiliki dua dimensi, untuk memperoleh posisi sebuah titik dalam *map* tersebut, maka diperlukan pengetahuan akan posisinya terhadap sumbu x dan sumbu y. Oleh karena itu, setiap kali proses pembuatan kromosom, diperlukan dua kromosom yang mewakili posisi terhadap sumbu x dan sumbu y.

Di dalam masing-masing kromosom, terdapat gen yang mewakili *waypoint* dalam suatu *path*. Dalam kasus ini, gen tetap menggunakan pengkodean desimal.

Dalam proses inisialisasi, terdapat beberapa parameter yang harus diketahui seperti: ukuran populasi, ukuran kromosom (panjang *waypoint*), dan posisi setiap *obstacle* di dalam *map*. Parameter-parameter tersebut harus diketahui karena dalam setiap proses pembuatan kromosom (*chromosome generation*) dibangkitkan kromosom (dengan ukuran tertentu) sejumlah ukuran populasi. Setelah itu, kromosom yang dibangkitkan harus dipastikan untuk tidak melewati *obstacle*. Jika kromosom yang baru ternyata melewati *obstacle* yang telah ditentukan, maka akan dicari lagi sebuah kromosom hingga akhirnya ditemukan yang bebas dari halangan.

Dalam proses pemilihan titik yang mungkin terpilih menjadi sebuah kromosom, digunakan *weighted sampling*. *Weighted sampling* adalah metode sampling yang memberi suatu nilai (*weight*) pada setiap titik yang mungkin, sehingga terdapat daerah tertentu yang memiliki probabilitas yang lebih besar dibanding daerah lain.

Pada titik awal dan titik akhir yang telah ditetapkan [(0,0) dan (20,20)], jika tidak terdapat *obstacle*, maka kira-kira *waypoint* yang dihasilkan akan berupa garis lurus dari titik awal ke akhir. Dengan demikian, untuk penentuan *weight*, pada daerah yang dekat (*radius obstacle* + 2) diukur dari garis $x=y$, akan diberi probabilitas 0.55 dibagi dengan jumlah titik yang ada dalam daerah tersebut. Sedangkan daerah yang lain akan memiliki probabilitas sebesar 0.45 dibagi dengan jumlah titik yang ada dalam daerahnya.

3.2.2 Fitness Function dan Seleksi

Fitness function adalah fungsi yang digunakan untuk mengukur kedekatan suatu kandidat solusi dengan tujuannya. Dalam permasalahan *path planning*, *fitness function*-nya adalah jarak terpendek dari posisi awal ke posisi tujuan yang diinginkan. Oleh karena itu, *fitness function* diharapkan meningkat seiring dengan menurunnya jarak, dengan kata lain, *fitness function* memiliki hubungan terbalik dengan persamaan jarak dari dua titik (*distance*).

Oleh karena dalam proses pembuatan kromosom telah dipastikan bahwa semua kromosom pasti tidak melewati *obstacle* (*feasible*), maka nilai *fitness function* pada saat *path*-nya melewati *obstacle* tidak perlu ditentukan.

Setelah setiap kromosom telah diketahui nilai *fitness*-nya, dilakukan proses seleksi yang bertujuan untuk memilih kromosom yang akan dimasukkan ke dalam proses *crossover* (memilih *parent*). Proses seleksi *roulette wheel* sering kali disebut sebagai *fitness proportionate selection*, dimana proses seleksi dilakukan secara acak, namun kromosom dengan nilai *fitness* besar diberi probabilitas yang besar juga (proporsional). Proses tersebut dilakukan sebanyak *parent* yang diperlukan untuk membangkitkan generasi berikutnya yang memiliki jumlah populasi yang sama. Persamaan mengenai probabilitas sebuah kromosom terpilih sebagai *parent* melalui *roulette wheel selection* adalah sebagai berikut: (dengan N sebagai total populasi.)

3.2.3 Operator Genetika

Operator genetika yang digunakan dalam kasus ini adalah *crossover* dan mutasi. *Crossover* (kawin silang) merupakan proses rekombinasi 2 *parent* untuk memperoleh dua kromosom baru. Metode *crossover* yang digunakan adalah metode *single point*

crossover, dimana gen dalam kedua kromosom *parent* setelah *point crossover* ditukar. Nilai P_c (*crossover probability*) merupakan probabilitas bahwa sepasang kromosom akan melalui proses *crossover*.

Operator genetika mutasi digunakan pada kromosom yang dihasilkan setelah melalui proses *crossover*. Mutasi diperlukan agar dimungkinkan muncul gen (*waypoint*) baru yang tidak ada di populasi sebelumnya. Di dalam kasus ini, semua kromosom melalui proses mutasi. Proses ini diawali dengan memilih secara acak (*weighted sampling*) gen di dalam sebuah kromosom yang akan melalui proses mutasi menggunakan nilai probabilitas P_m (*mutation probability*). Setelah terpilih, maka akan dipilih secara random sebuah angka yang diambil dari database tentang titik-titik yang tidak melewati *obstacle*. Baru setelah itu, dilanjutkan dengan uji *feasibility* untuk menentukan apakah garis yang dibentuk kedua titik menyinggung *obstacle*.

3.2.4 Uji *Feasibility* Kromosom

Pada tahap ini, kromosom yang telah ditemukan dari proses-proses sebelumnya dites *feasibility*-nya, dimana kriteria *feasible*-nya adalah baik titik belok maupun *path* yang berupa garis lurus antara kedua titik tidak boleh melewati *obstacle*. Tes dilakukan menggunakan persamaan matematika untuk mengecek apakah *path* (garis lurus) bersinggungan dengan suatu *obstacle* (lingkaran)..

Proses pengujian ini dilakukan setiap selesai proses pembentukan kromosom dan proses mutasi. Setelah proses *crossover* tidak dicek karena selain proses ini menggunakan *computational cost* yang besar, proses *crossover* juga langsung diikuti oleh proses mutasi yang membuat proses ini menjadi *redundant*.

3.2.5 *Elitism*

Salah satu bentuk upaya untuk mempertahankan kromosom terbaik dalam suatu generasi adalah dengan menerapkan *elitism*. Dalam proses *elitism*, kromosom yang memiliki nilai fitness paling tinggi tidak akan mengikuti proses *crossover* dan mutasi, melainkan akan langsung menjadi kromosom berikutnya. Dalam tugas akhir ini, *elitism* diterapkan dengan hanya mengambil 1 kromosom terbaik agar populasi yang lain bisa berkesempatan mencari solusi yang lebih baik.

3.2.6 Alur Kerja *Path Planning* Statis

Alur kerja *path planning* statis adalah:

1. Tentukan parameter-parameter yang diperlukan seperti: ukuran populasi, ukuran kromosom, nilai P_c dan P_m , banyak generasi, apakah *elitism* berlaku, posisi dan jari-jari *obstacle* dan lain-lain.
2. Inisialisasi populasi pertama
3. Pastikan baik titik maupun garis *path* tidak menyinggung *obstacle*.
4. Untuk generasi pertama hingga ke-N, lakukan:
 - a. Hitung nilai *fitness* masing-masing kromosom
 - b. Seleksi
 - c. *Crossover*
 - d. Mutasi
 - e. Pastikan baik titik maupun garis tidak menyinggung *obstacle*.

3.3 *Forward Kinematic* dari *Mobile Robot*

Robot memiliki dua perbedaan dasar, yaitu: robot memiliki sensor dan dapat bergerak bebas secara mandiri. Karena gerakan mereka sangat berbeda dari lengan manusia, kita perlu berbicara tentang gaya kinematika baru: *Differential Drive*. *Differential Drive Robot* adalah robot bergerak yang gerakannya didasarkan pada dua roda yang digerakkan secara terpisah yang diletakkan di kedua sisi tubuh robot. Dengan demikian dapat mengubah arahnya dengan memvariasikan tingkat relatif rotasi roda-rodanya dan karenanya tidak memerlukan gerakan kemudi tambahan. Ketika kedua roda berputar pada kecepatan yang sama di arah yang sama, robot bergerak lurus ke arah itu. Ketika satu roda berputar lebih cepat dari yang lain, robot berputar melengkung ke arah roda lebih lambat. Ketika roda berbelok ke arah yang berlawanan, robot berputar di tempat. Pergerakan robot secara umum dapat dijabarkan sebagai berikut:

- a. Jika robot bergerak dalam kurva, ada pusat kurva itu pada saat itu, dikenal sebagai Pusat Kelengkungan Instan (atau ICC). Kami berbicara tentang pusat sesaat, karena kita akan menganalisis ini pada waktu tertentu karena kurva mungkin akan berubah pada waktu berikutnya.

- b. Jika r adalah jari-jari kurva (diukur ke tengah robot) dan l adalah jarak antar roda, maka laju rotasi (ω) di sekitar ICC adalah terkait dengan kecepatan roda dengan:

$$\omega \left(r + \frac{1}{2} \right) = v_r \quad (1)$$

$$\omega \left(r - \frac{1}{2} \right) = v_l \quad (2)$$

Mengapa? Kecepatan sudut didefinisikan sebagai kecepatan posisi dibagi dengan jari-jari:

$$\frac{d\theta}{dt} = \frac{V}{r} \quad (3)$$

Ini seharusnya masuk akal secara intuitif: semakin jauh dari pusat rotasi, semakin cepat robot perlu bergerak untuk mendapatkan kecepatan sudut yang sama. Jika robot bergerak dengan kecepatan π radian per detik selama 1 detik, maka robot akan menempuh jarak setengah keliling, atau πr . Karena ini dalam satu detik, kecepatannya adalah πr per detik. Dengan π radian per detik sama dengan kecepatan πr , maka $v = \omega r$. Setelah kita memiliki persamaan itu, kita bisa menyelesaikannya untuk r atau ω :

$$v_r = \omega \left(r + \frac{1}{2} \right) \quad (4)$$

$$= \omega r + \omega \frac{1}{2} \quad (5)$$

$$v_l = \omega \left(r - \frac{1}{2} \right) \quad (6)$$

$$= \omega r - \omega \frac{1}{2} \quad (7)$$

melalui proses eliminasi

$$v_r - v_l = \frac{2\omega l}{2} \quad (8)$$

$$\omega = \frac{v_r - v_l}{l} \quad (9)$$

melalui proses penambahan

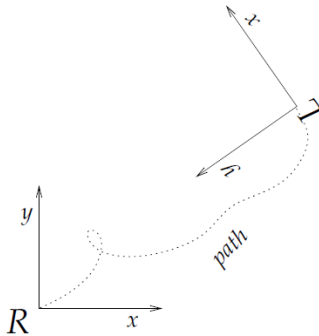
$$2\omega r = v_r + v_l \quad (10)$$

$$r = \frac{l(v_r + v_l)}{2(v_r - v_l)} \quad (11)$$

Hal yang perlu diperhatikan:

- Kecepatan sudut adalah perbedaan dalam kecepatan roda berdasarkan jarak antar roda.
- Jika $v_r = v_l$, maka $\omega = 0$, robot bergerak lurus.
- Jika $v_r = -v_l$, maka $r = 0$, robot berputar di tempat.

Robot itu, pada suatu waktu di suatu lokasi x ; y , dan menghadap ke arah yang membentuk sudut dengan sumbu x dari kerangka referensi. Mendefinisikan $\theta = 0$ menjadi robot yang menghadap sepanjang sumbu x positif membuat kita konsisten dengan tradisi matematika tetapi memiliki konsekuensi tambahan. Ketika robot bergerak, frame lokalnya bergerak bersamanya, begitu pula sudut antara frame referensi x -axis dan frame lokal x -axis. Triple x ; y ; disebut pose robot.



Gambar 3.1, Posisi robot pada bidang cartesian

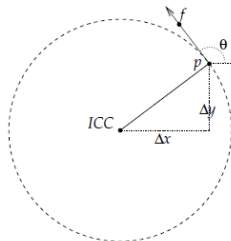
Permasalahan pada *forward kinematic* adalah: diberikan robot pada beberapa posisi, serta bergerak pada kecepatan sudut ω selama periode waktu t , tentukan posisi baru untuk robot.

- Pertama, perhatikan bahwa semua elemen ini adalah fungsi waktu: $x(t)$; $y(t)$; $\pi(t)$; $V(t)$; $\theta(t)$.

- b. Selanjutnya, mari kita menghitung di mana ICC, diberikan r . Dalam gambar (saat ini tidak ada), robot menghadap ke arah yang ditunjukkan oleh arah \vec{pf} . Arah tersebut bersinggungan dengan kurva yang dilalui oleh robot pada saat itu, sehingga segmen dari p ke ICC tegak lurus terhadap \vec{pf} . Variabel x dan y membentuk segitiga siku-siku, dan:

$$\Delta x = -r \sin \theta$$

$$\Delta y = r \cos \theta$$



Gambar 3.2 Diagram posisi robot

Kami akan menggambarkan posisi robot sebagai vektor kolom $\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$.

Perhatikan bahwa v_l dan v_r benar-benar fungsi waktu $v_l(t)$ dan $v_r(t)$ (mereka berubah seiring waktu ketika robot bergerak dengan cara yang berbeda) dengan demikian, r dan! keduanya juga fungsi waktu $r(t)$ dan! (t) . Karena kita tidak memiliki ekspresi yang bagus untuk fungsi-fungsi ini, apa yang biasanya kita lakukan adalah memecah aksi robot ke periode waktu di mana v_l dan v_r konstan, dan dengan demikian dapat diganti dengan angka tunggal. Setelah kita memiliki angka-angka tunggal itu, rumus di atas mudah untuk dihitung.

(B) Tapi, memikirkan parameter ini sebagai fungsi memberi kita cara lain untuk mendapatkan persamaan ini; cara yang dirasakan banyak orang adalah lebih sederhana. Dalam metode ini, kami mencatat bahwa fungsi x ; y ; tergantung pada fungsi kecepatan $V(t)$ dan rotasi sudut! (t) :

$$x(t) = \int_0^t V(t) \cos[\theta(t)] dt \quad (12)$$

$$(13)$$

$$y(t) = \int_0^t V(t) \sin[\theta(t)] dt$$

$$\theta(t) = \int_0^t \omega t dt \quad (14)$$

Dalam kasus robot penggerak diferensial, $V(t) = \frac{v_r(t)+v_l(t)}{2}$, rata-rata dari dua roda, dan sebelumnya telah diketahui $\omega(t)$. Dengan demikian dengan substitusi, persamaan menjadi:

$$x(t) = \frac{1}{2} \int_0^t [v_r(t) + v_l(t)] \cos[\theta(t)] dt \quad (15)$$

$$y(t) = \frac{1}{2} \int_0^t [v_r(t) + v_l(t)] \sin[\theta(t)] dt \quad (16)$$

$$\theta(t) = \frac{1}{2} \int_0^t \omega(t) dt \quad (17)$$

Sekali lagi, kita tidak memiliki deskripsi fungsi-fungsi ini, tetapi jika kita mengasumsikan bahwa v_r dan v_l adalah konstan, maka kita dapat menggantikan:

$$\theta(t) = \int_0^t \frac{v_r - v_l}{l} dt = \frac{v_r - v_l}{l} t \quad (18)$$

Lalu kita bisa mensubstitusi ini kembali:

$$x(t) = \frac{1}{2} \int_0^t [v_r + v_l] \cos \left[\frac{v_r - v_l}{l} t \right] dt \quad (19)$$

$$x(t) = \frac{v_r + v_l}{v_r - v_l} \frac{1}{2} \sin \left[\frac{v_r - v_l}{l} t \right] \quad (20)$$

dan

$$y(t) = \frac{1}{2} \int_0^t [v_r + v_l] \sin \left[\frac{v_r - v_l}{l} t \right] dt \quad (21)$$

$$y(t) = -\frac{v_r + v_l}{v_r - v_l} \frac{1}{2} \cos \left[\frac{v_r - v_l}{l} t \right] + \frac{v_r + v_l}{v_r - v_l} \frac{1}{2} \quad (22)$$

Dalam semua analisis ini, diasumsikan bahwa $v_r \neq v_l$. Kalau tidak r akan bernilai tak hingga dan persamaannya tidak valid. Ketika keduanya sama, kita perlu menurunkan persamaan yang berbeda. Mulai dari persamaan (4), kami mencatat bahwa $\theta(t)$ bukan lagi fungsi waktu, karena $\theta(t)$ tidak pernah berubah. Karena kita menurunkan ini dalam kerangka local, maka $\theta(t) = 0$.

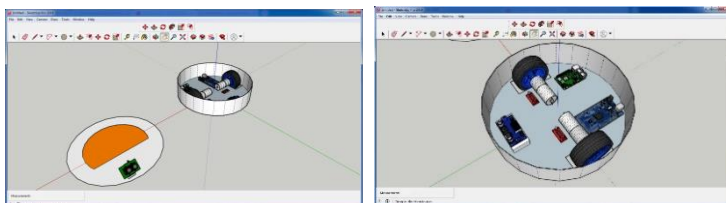
3.4 Perancangan Perangkat Keras

Dalam melakukan perancangan perangkat keras, digunakan *software* Google Sketchup untuk membantu mendesain tata letak serta dimensi dari robot penghisap debu. Dalam perancangan ini, dilakukan modifikasi dari *vacuum cleaner* yang sudah ada di pasaran, yaitu Mitsuyama MS-SR3.



Gambar 3.3 Mitsuyama MS-SR3

Adapun hasil desain dari *software* Google Sketchup dapat dilihat pada gambar berikut:



Gambar 3.4 Hasil desain pada Google Sketchup

3.4.1 Komponen yang digunakan

Komponen yang digunakan dalam perancangan perangkat keras robot penghisap debu antara lain:

1. Raspberry Pi

Raspberry Pi yang digunakan dalam pengerjaan tugas akhir ini adalah Raspberry Pi 4. Raspberry Pi 4 berfungsi sebagai komponen untuk memproses *genetic algorithm* berbasis *Robot Operating System (ROS)*.



Gambar 3.5 Raspberry Pi 4

2. Arduino Mega 2560

Arduino Mega 2560 berfungsi sebagai media komunikasi antara sensor, Raspberry Pi 4 dan aktuator. Dalam perancangan perangkat keras digunakan Arduino karena komunikasi analog antara sensor dan pin GPIO Raspberry Pi 4 tidak cukup mumpuni dalam pengerjaan sistem.



Gambar 3.6 Arduino Mega 2560

3. SZCM 25GA370

Merupakan motor penggerak roda robot dengan kapasitas 100 rpm yang dilengkapi dengan encoder.



Gambar 3.7 SZCM 25GA370

- VNH3ASP30 30A
Motor Driver sebagai aktuator dari sistem.



Gambar 3.8 VNH3ASP30 30 A

- LM2569
Buck converter sebagai penyesuai tegangan input dari baterai sehingga tidak menyebabkan tegangan berlebih pada komponen.



Gambar 3.9 LM2569

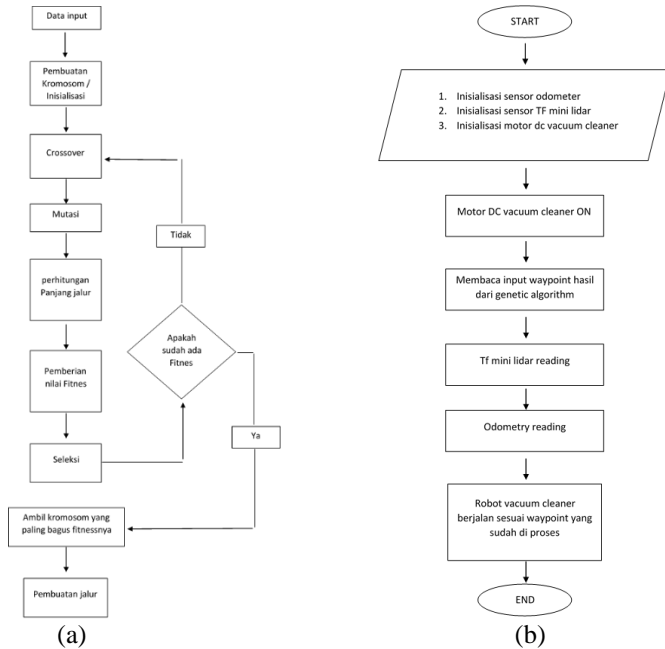
Adapun tampilan perangkat keras setelah selesai dilakukan modifikasi dan perangkaian tampak seperti gambar dibawah ini:



Gambar 3.10 Hasil dari realisasi perancangan *hardware*

3.5 Blok Diagram

Adapun blok diagram dari sistem yang dirancang dalam Tugas Akhir ini dapat dilihat pada gambar berikut ini:



Gambar 3.11 (a) Genetic Algoritim (b) Komponen mekanis

BAB IV HASIL DAN ANALISA

4.1 Perumusan Algoritma Genetika

Algoritma genetika yang digunakan memiliki karakteristik sebagai berikut:

1. Representasi kromosom yang digunakan adalah permutasi dari node-node tujuan. Sebagai contoh, apabila node-node tujuan adalah n_1, n_2, n_3, n_4, n_5 , maka salah satu kromosom yang dapat digunakan adalah n_3, n_5, n_1, n_2, n_4 . Hal ini menunjukkan bahwa rute yang terbentuk adalah $n_3 \rightarrow n_5 \rightarrow n_1 \rightarrow n_2 \rightarrow n_4$

Tabel 4.1 Representasi Kromosom

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

9	8	7	6	5	4	3	2	1
---	---	---	---	---	---	---	---	---

					6	7	8	
--	--	--	--	--	---	---	---	--

9	8	7	6	5	4	3	2	1
---	---	---	---	---	---	---	---	---

2. Besar populasi yang digunakan adalah 50.
3. Fungsi fitness yang digunakan adalah

$$\sum_{i=1}^m \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad (1)$$

dimana x_i dan y_i adalah absis dan ordinat dari node tujuan ke- i

4. Metode seleksi yang digunakan adalah metode *elitism*. Metode seleksi *elitism* memilih kromosom-kromosom untuk dipakai pada generasi selanjutnya didasarkan pada urutan nilai

fitness. Semakin baik nilai fitnessnya maka kromosom tersebut akan dipertahankan

5. Operator mutasi yang digunakan adalah *swap mutation*. Dengan *swap mutation* 2 node dalam rute dipilih secara acak, kemudian posisinya ditukar. Sebagai contoh, jika kita terapkan *swap mutation* pada *list* berikut, [1,2,3,4,5], kemungkinan hasil swap yang didapat adalah [1,2,5,4,3]. Disini posisi 3 dan 5 ditukar sehingga membuat list yang dengan nilai-nilai yang persis sama, hanya berbeda urutan. Karena *swap mutation* hanya menukar nilai yang sudah ada, maka tidak akan terjadi pembuatan list yang mempunyai nilai yang hilang atau ganda ketika dibandingkan dengan yang asli.
6. Operator *crossover* (kawin silang) yang digunakan adalah *crossover* terurut. Dalam metode ini dipilih subset dari *parent* pertama, dan kemudian subset tersebut ditambahkan ke *offspring*. Jika ada nilai yang hilang maka kemudian ditambahkan ke *offspring* dari *parent* kedua dengan urutan ketika ditemukan. Disini subset dari rute diambil dari *parent* pertama (6,7,8) dan kemudian ditambahkan ke rute *offspring*. Berikutnya, lokasi rute yang hilang ditambahkan dalam urutan dari *parent* kedua. Node pertama dari rute *parent* kedua adalah 9 yang mana tidak terdapat pada rute *offspring* jadi ditambahkan di posisi pertama yang tersedia. Posisi berikutnya pada rute *parent* adalah 8 yang mana terdapat di rute *offspring* maka dilewatkan. Proses ini dilanjutkan hingga *offspring* tidak mempunyai nilai yang kosong lagi.
7. Kondisi pemberhentian yang digunakan adalah ketika banyaknya generasi telah mencapai 1000.

4.2 Pengujian Sistem

Pengujian proses algoritma genetika dilakukan dengan melakukan perubahan nilai parameter yang digunakan, yaitu nilai mutation rate, jumlah kromosom perpopulasi serta penggunaan metode seleksi elitism atau tidak. Bentuk rute pengujian dapat dilihat pada Gambar 5. Pengujian yang dilakukan adalah mencari rute paling

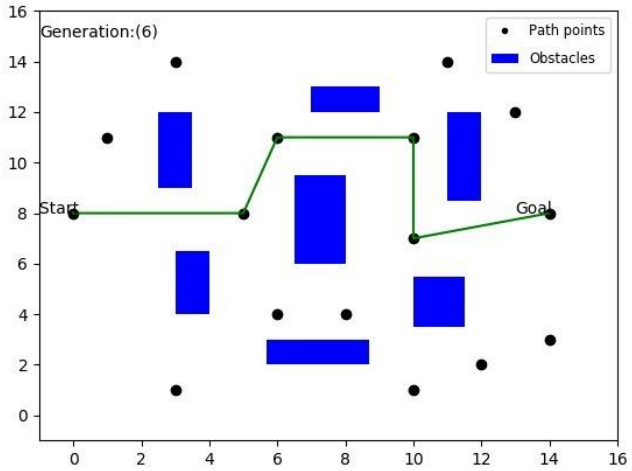
optimum. Rute paling optimum didapat dengan nilai final distance terkecil. Secara ringkas, hasil pengujian dapat dilihat pada Tabel berikut ini.

Tabel 4.2 Hasil Pengujian

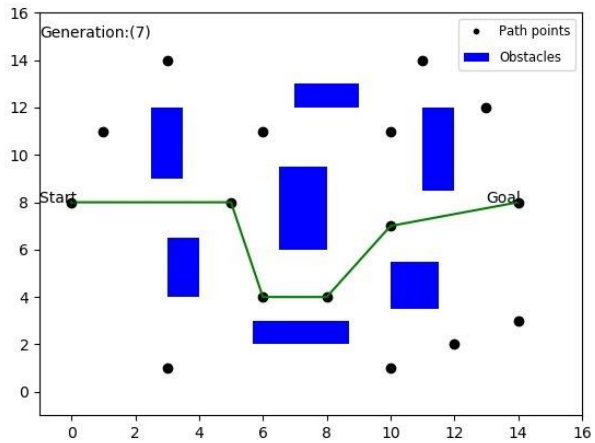
Jumlah Kromosom	Elitism	Mutation Rate	Final Distance
100	True	0,08	44
100	False	0,08	53
100	True	0,02	63
100	False	0,02	105
220	True	0,08	44
220	False	0,08	52
220	True	0,02	62
220	False	0,02	106

Dari hasil pengujian didapatkan bahwa jumlah kromosom tidak berpengaruh signifikan terhadap hasil yang didapat. Hasil rute paling optimal dari pengujian adalah dengan *final distance* 44 yang didapat dari pengujian dengan jumlah kromosom perpopulasi adalah 100 dan 220, *elitism true* dan *Mutation rate* 0.08. Meskipun didapat rute dengan jarak optimal yang sama, tetapi menghasilkan rute yang berbeda seperti yang terlihat pada Gambar 6 dan Gambar 7. Hal ini dikarenakan jumlah kromosom perpopulasi yang berbeda. Tetapi keduanya dapat dikatakan rute yang paling optimal. Rute paling jauh didapat dari kombinasi jumlah kromosom 220, *elitism falsedan mutation rate* 0.2 dengan *final distance* 106. Rute yang didapat dapat dikatakan “tidak teratur” berbeda dengan kondisi teratur sesuai Gambar 6 dan Gambar 7. Hal ini dikarenakan metode seleksi *elitism* tidak diterapkan sehingga kromosom-kromosom yang memiliki nilai fitness terbaik tidak dipertahankan sehingga meningkatkan kemungkinan untuk mendapatkan kromosom yang kurang baik.

Pada generasi selanjutnya didasarkan pada urutan nilai fitness. Semakin baik nilai fitnessnya maka kromosom tersebut akan dipertahankan . Oleh karena itu, hasil yang didapat lebih optimal. Dari tabel 1 juga dapat dilihat bahwa nilai *mutation rate* 0.08 lebih menghasilkan hasil yang lebih baik daripada nilai *mutation rate* 0.2



Gambar 4.1 *Final Distance 44* (Jumlah Kromosom100)



Gambar 4.2. *Final Distance 44* (Jumlah Kromosom220)

4.3 Pengujian Gerak Robot pada Luasan Tertentu

Setelah melakukan simulasi *genetic algorithm* pada Spyder, dilakukan uji arena pada area seluas 8 meter persegi.



Gambar 4.3 Arena uji coba pergerakan robot penghisap debu

Dengan area seperti gambar diatas dan diberikan haling rintang sebesar 20 cm x 40 cm, maka didapatkan lama waktu pembersihan sebagai berikut:

Tabel 4.3 Hasil uji coba waktu pembersihan

No	Metode	Jumlah Kromosom	Luas Area (m ²)	Waktu Pembersihan
1	<i>Random Walk</i>	0	8	10 menit
2	Algoritma Genetika menggunakan 100 iterasi	40	8	7 menit
3	Algoritma Genetika menggunakan 220 iterasi	40	8	7 menit

halaman ini sengaja dikosongkan

BAB V

PENUTUP

Dari perancangan dan analisa yang telah dilakukan, dapat dirangkum dalam sebuah kesimpulan. Untuk kekurangan dan kendala yang dihadapi, dituliskan dalam bagian saran, untuk membantu penelitian selanjutnya.

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari penelitian ini adalah dengan menggunakan jumlah populasi yang tepat, metode seleksi *elitism* dan *mutation rate* yang tepat, algoritma genetika dapat dapat diimplementasikan dalam permasalahan penentuan rute optimal. Nilai *mutation rate* sebesar 0.08 memberikan hasil paling optimal.

Dengan menggunakan metode *genetic algorithm* didapatkan perbedaan waktu pembersihan sebesar 3 menit untuk luasan area pembersihan sebesar 8 meter persegi.

5.2 Saran

Setelah melakukan penelitian ini, untuk pengembangan penelitian, disarankan beberapa hal, antara lain:

1. Penggunaan sensor LiDAR dengan cakupan pindai yang lebih luas untuk memberikan perancangan jalur yang lebih baik.
2. Pemutakhiran penerapan ROS pada *mobile robot* sehingga didapatkan hasil komputasi yang lebih baik dan waktu pembersihan yang lebih efisien.
3. Penerapan dan pengembangan algoritma agar bisa diterapkan melakukan pemetaan secara *real-time* dalam lingkup ruang yang lebih luas

halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

1. Fahmizal, et al. (2019). Trajectory Tracking pada Robot Omni dengan Metode Odometry. *JNTETI*, 8(1), 35-44.
2. Tzafestas, Spyros G. (2018). Mobile Robot Control and Navigation: A Global Overview. *Journal of Intelligent & Robotic Systems*. doi:10.1007/s10846-018-0805-9
3. Mouad, Mehdi et al. (2012). Mobile Robot Navigation and Obstacles Avoidance based on Planning and Re-Planning Algorithm. *IFAC Symposium on Robot Control*, 10. 622-628.
4. Erlangga, Dwiky et al. (2019). Sistem Navigasi Mobile Robot dalam Ruangan Berbasis Autonomous Navigation. *Journal of Mechanical Engineering and Mechatronics*, 4(2). 78-86.
5. Zhang, Xiaolei. (2016). Dynamic Path Planning Algorithm for a Mobile Robot Based on Visible Space and an Improved Genetic Algorithm. *International Journal of Advanced Robotic Systems*, 13. 1-17.
6. Gao, Xueshan. (2007). A Floor Cleaning Robot Using Swedish Wheels. *IEEE International Conference on Robotics and Biomimetics*, 2069-2073.
7. M.-C. Kang et al. (2014). A Robust Obstacle Detection Method for Robotic Vacuum Cleaners. *IEEE Transactions on Consumer Electronics*, 60(4). 587-595.
8. J. Wang et al. (2011). Path Planning of Mobile Robot Based on Improving Genetic Algorithm. *International Conference on ICCE*, 112. 535-542.
9. Jianwei Ma et al. (2020). Robot Path Planning Based on Genetic Algorithm Fused with Continuous Bezier Optimization. *Hindawi Computational Intelligence and Neuroscience*, 2020. 1-10.
10. Lamini, Chaymaa et al. (2018). Genetic Algorithm Based Approach for Autonomous Mobile Robot Path Planning. *Procedia Computer Science*, 127. 180-189
11. Gul, Faiza et al. (2019). A Comprehensive Study for Robot Navigation Techniques. *Cogent Engineering*, 6:1, 1632046.
12. Moghadam, P., Wijesoma, W.S., & Feng, D.J. (2008). Improving Path Planning and Mapping Based on Stereo Vision and Lidar. *Intl. Conf. on Control, Automation, Robotics and Vision*, 10. 384-389.

13. Choueiry, Samia et al. (2019). Mobile Robot Path Planning Using Genetic Algorithm in a Static Environment. *International Conference on Advances in Computational Tools for Engineering Applications*, 4.
14. Achour, Nouara & Chaalal, M. (2011). Mobile Robots Path Planning using Genetic Algorithms. *International Conference on Autonomic and Autonomous Systems*, 7. 111-115.
15. Chi Zhang et al. (2017). 2D Map Building and Path Planning Based on LiDAR. *International Conference on Information Science and Control Engineering*, 4. 783-787.

BIODATA PENULIS



Penulis bernama Muhammad Faris. Penulis merupakan putra pertama dari Bapak Ir. Moh. Wahyudin Isbandi, M.Si dan Ibu Sri Purnama Dewi, A.Md dari dua bersaudara. Penulis dilahirkan pada tanggal 18 November 1996 di Padang, Sumatera Barat. Penulis menempuh pendidikan sekolah dasar di SD Kartika I-9 Pekanbaru, kemudian dilanjutkan dengan pendidikan menengah di SMP Negeri 04 Pekanbaru selama dua tahun dan di SMA Negeri 08 Pekanbaru selama 3 tahun. Penulis kemudian melanjutkan pendidikan perguruan tinggi di Institut Teknologi Sepuluh Nopember Surabaya pada tahun 2013. Selama masa perkuliahan, penulis aktif sebagai pengurus di salah satu organisasi sosial ekstra kampus yaitu Swayanaka Indonesia Regional Surabaya pada tahun 2015 sampai 2017. Selain itu, penulis juga pernah menjadi pengurus salah satu paguyuban mahasiswa, yaitu Ikatan Pelajar dan Mahasiswa Riau (IPMR) dalam kurun waktu 2013-2015. Pada tahun 2019, penulis juga sempat melaksanakan kerja praktek di PT. PLN (Persero) Unit Pelaksana Pengatur Distribusi (UP2D) Sumatera Barat selama kurang lebih 1 bulan.

E-mail : farismuhammad384@gmail.com

Whatsapp : +62 822 9777 6907