

Desain Robot Pengantar Obat dengan Algoritma *Dynamic Window Approach*

Muhammad Fakhri Putra Supriyadi ^{*1}, Teguh Muflih Rizky ^{*2}, Alfi Abdurrahman Addachil ^{*3}, Muhammad Rifki Madani ^{*4}, Naufal Syafiqurrahman ^{*5}, Melani Permata Saghira L. ^{*6}

*Teknik Elektro, Telkom University
Terusan Buah Batu, Bandung, Indonesia*

¹ fakhriputra@student.telkomuniversity.ac.id

² teguhmuflih@student.telkomuniversity.ac.id

³ alfiabdr@student.telkomuniversity.ac.id

⁴ mrifkimadani@student.telkomuniversity.ac.id

⁵ naufalsyafiqurrahman@student.telkomuniversity.ac.id

⁶ melanipermata@student.telkomuniversity.ac.id

Abstract

Robot could easily be found anywhere in this time around. The usage of robot is not limited to industrial usage but also to help medical officer. In this report, we will explain the result of drug delivery robot for address frontier region that we have designed. We hope the existence of this robot will be able to help medical officer to supply drugs to all areas that are hard to reach with the help of Dynamic Window Approach (DWA) pathfinding algorithm.

Keywords: robot, DWA, 3T, pathfinding.

Abstrak

Saat ini penggunaan robot mudah sekali kita temui dimana saja. Penggunaan robot tidak terbatas hanya pada robot industri tetapi juga robot yang dapat membantu petugas medis. Pada laporan ini, kami akan menjelaskan hasil robot pengantar obat untuk daerah 3T yang sudah kami desain. Dengan adanya robot ini diharapkan dapat membantu tenaga medis menyuplai obat-obatan ke daerah yang sulit dijangkau dengan menggunakan algoritma pathfinding yaitu *Dynamic Window Approach* (DWA).

Kata Kunci: robot, DWA, 3T, *pathfinding*.

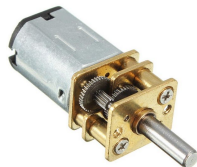
I. DESAIN PLATFORM ROBOT

BANYAKNYA pulau-pulau yang ada di Indonesia menjadikan Indonesia sebagai salah satu negara yang mempunyai wilayah yang sangat luas. Hal tersebut tentunya akan mempengaruhi SDM yang ada di setiap wilayah, khususnya pada wilayah 3T (daerah tertinggal, terdepan dan terluar Indonesia). Keterbatasan aktivitas dalam wilayah 3T dampaknya sangat terasa oleh orang-orang yang menetap disana. Dengan adanya perkembangan teknologi yang semakin pesat pada saat ini tentunya akan mempermudah dan membantu aktivitas yang biasa dilakukan sehari-hari. Untuk saat ini sudah banyak teknologi modern dalam bidang otomasi khususnya robot yang dapat mempermudah dalam pengoperasian suatu alat secara otomatis, dimana dengan adanya robot tersebut mampu meringankan pekerjaan manusia.

Pada dasarnya robot merupakan suatu sistem terintegrasi yang membuat sebuah sistem mampu mengerjakan suatu tindakan berdasarkan input yang diterimanya. Dunia robotika sejauh ini telah memberikan banyak manfaat yang positif bagi aktivitas manusia, salah satunya yaitu diciptakannya robot pengantar obat. Robot ini nantinya akan melaksanakan tugasnya sebagai petugas pengantar obat seperti yang sering dijumpai di rumah sakit, pada umumnya petugas medis akan mengantarkan obat-obatannya ke setiap kamar pasien secara rutin.

Sekilas hampir sama dalam teknis pengantaran obatnya, dimana robot ini akan mengantarkan obat ke setiap kamar pasien, bisa juga mengantarkan obat-obatan hingga ke rumah pasien apabila memang pasien tidak sempat untuk ke rumah sakit. Tentunya robot tersebut akan mempermudah dari kinerja petugas medis, dan juga menjaga protokol kesehatan dan mencegah penyebaran COVID-19. Langkah awal dalam pembuatan robot pengantar obat yakni membuat desain keseluruhan robot sesuai dengan keperluan yang telah ditentukan sebelumnya, nantinya robot tersebut akan dioperasikan pada daerah 3T. Mengingat daerah 3T juga memiliki keterbatasan akses antar wilayahnya (Contoh jalan antar desa) maka dari itu desain robot yang akan dibuat haruslah yang bisa digunakan di segala medan.

A. Komponen Robot



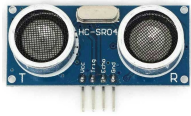
Gambar 1. Motor DC.

1) *Motor DC*: Motor DC yang digunakan adalah 600RPM Mini Metal Gear Motor DC N20 gearbox 6V 12V DIY Arduino Robotic. Dibutuhkan 2 motor DC pada robot ini yang berfungsi sebagai penggerak roda robot.



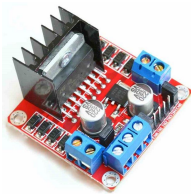
Gambar 2. Roda.

2) *Roda*: Roda berfungsi sebagai penggerak pada robot. Robot ini menggunakan roda yang bergerigi dengan tujuan untuk menyesuaikan dengan medan yang dilewati.



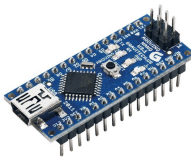
Gambar 3. Sensor Ultrasonik.

3) *Sensor Ultrasonik*: Sensor Ultrasonik pada robot ini berfungsi sebagai pendeteksi tujuan robot atau pendeteksi tempat dimana robot harus mengantarkan obat yang dibawanya.



Gambar 4. Driver Motor L298N.

4) *Driver Motor*: Driver Motor pada robot ini berfungsi sebagai fungsikan pengontrol kecepatan serta arah perputaran motor DC.



Gambar 5. Arduino Nano.

5) *Arduino Nano*: Arduino Nano pada robot ini berfungsi sebagai pengendali robot.



Gambar 6. Baterai Lithium 12V.

6) *Baterai*: Baterai pada robot ini berfungsi sebagai penyimpan daya.



Gambar 7. Powerbank UBEC 5V.

7) *UBEC*: UBEC (Universal Battery Elimination Circuit) pada robot ini berfungsi sebagai rangkaian elektronik yang mengambil daya dari battery pack atau sumber DC lainnya, dan menurunkannya ke level tegangan 5V.



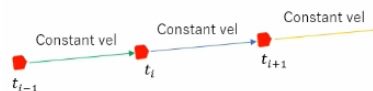
Gambar 8. Motor Servo.

8) *Motor Servo*: Motor Servo pada robot ini berfungsi untuk mendorong atau memutar arah robot.

II. ALGORITMA ROBOT

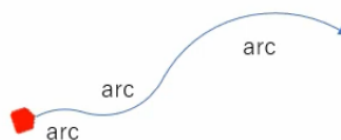
Robot menggunakan algoritma untuk mencari rute/*pathfinding* yang disebut dengan *dynamic window approach* yaitu untuk mempertimbangkan akselerasi yang terbatas oleh motor, keseluruhan search space dikurangi menjadi "Dynamic window" yang hanya berisi kecepatan yang dapat dicapai dalam interval waktu berikutnya.

A. Pendekatan algoritma DWA (*Dynamic Window Approach*)



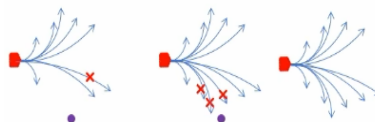
Gambar 9. Kecepatan dalam sebuah lintasan.

1. Kecepatan sedikit demi sedikit konstan dalam interval waktu.



Gambar 10. Lintasan robot.

2. Pendekatan lintasan robot mengikuti urutan busur lingkaran.



Gambar 11. Pencarian ruang.

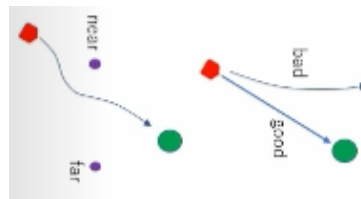
Algoritma bekerja dengan mencari ruang terlebih dahulu yang selanjutnya akan menghasilkan lintasan yang melingkar yang secara unik ditentukan oleh pasangan kecepatan translasi dan rotasi, lalu dengan

memilih kecepatan yang dapat diterima jika robot dapat berhenti sebelum mencapai rintangan, dan membuat *dynamic window* yang akan membatasi kecepatan yang dapat diterima menjadi yang dapat dicapai dalam interval waktu yang singkat mengingat akselerasi robot yang terbatas.

$$G(v, \omega) = \sigma(\alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{dist}(v, \omega) + \gamma \cdot \text{vel}(v, \omega))$$

Gambar 12. Fungsi *Objective*.

Lalu selanjutnya akan dilakukan optimisasi pada algoritma nya dengan melakukan fungsi *objective* untuk mengambil rute terbaik.



Gambar 13. *Target Heading*.

Heading pada fungsi diatas merupakan ukuran progress menuju lokasi tujuan, nilainya akan maksimal jika robot bergerak langsung menuju sasaran, sedangkan *Dist* adalah jarak ke obstacle terdekat di lintasan, semakin dekat jarak ke suatu obstacle maka semakin tinggi keinginan robot untuk bergerak.

III. SPESIFIKASI ROBOT

Robot ini nantinya akan melaksanakan tugasnya sebagai petugas pengantar obat seperti yang sering dijumpai di rumah sakit, pada umumnya petugas medis akan mengantarkan obat-obatannya ke setiap kamar pasien secara rutin.

A. Desain Fisik Robot

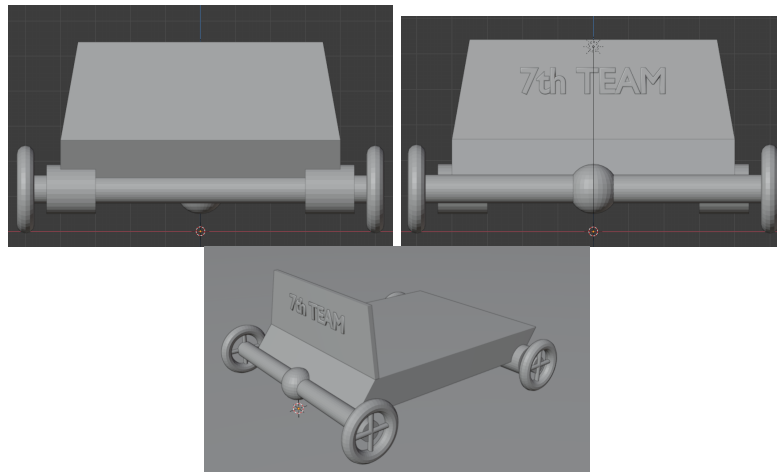
Agar robot ini dapat berjalan dengan semestinya maka kami menggunakan PID sebagai pengatur dari seluruh kerja robotnya. Robot ini gerakan oleh dua buah motor DC yang masing-masing langsung terhubung dengan ban sebagai penggerak utamanya, kecepatan dan arah gerak motor dapat di kendalikan oleh driver motor. Untuk dapat berbelok dengan baik di dalam ruangan sebuah rumah sakit, robot ini dibekali sebuah servo yang berfungsi sebagai steering yang dapat mengarahkan robot untuk berbelok ke kanan maupun ke kiri. Pada saat berbelok, kedua roda depan akan berbelok secara keseluruhan dikarenakan adanya penempatan servo pada batang silinder yang menghubungkan ke dua roda depan. Untuk spesifikasi dimensi dari robot yang kami buat dapat dilihat dari table berikut.

Tabel I
TABEL DIMENSI UKURAN ROBOT

Keterangan	Ukuran
Panjang	160cm
Lebar	90cm
Tinggi	80cm
Diameter Roda	30cm

Pada tahap pertama proses pembuatan robot ini kami menggunakan software blender sebagai pembuatan

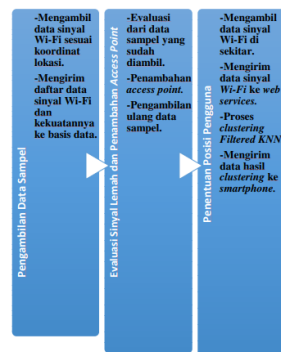
3D modelling nya untuk menentukan bentuk fisik robot yang nanti akan dibuat. Hasil dari 3D modelling software blender dapat dilihat sebagai berikut:



Gambar 14. Gambar bentuk robot.

Dengan adanya sketsa 3D modelling robot diatas dan juga ukuran robot yang telah ditentukan sebelumnya maka beban maksimum yang dapat dibawa oleh robot tersebut yaitu 7 Kg. Beban maksimum tersebut didapat dari asumsi dimensi robot dan juga mempertimbangkan agar robot tersebut dapat beroperasi dengan lancar.

B. Perancangan Alur sistem Indoor positioning system



Gambar 15. Tahapan Indoor Positioning System.

Proses alur yang dibangun pada sistem ini meliputi proses pengumpulan data sampel, proses evaluasi sinyal lemah dan penambahan access point, serta proses penentuan lokasi pengguna. Alur sistem dapat dilihat pada Gambar 15. Pada gambar tersebut, alur sistem dimulai dengan pengambilan data sampel.

Data sampel diambil dengan cara mengambil data-data Received Signal Strength (RSS) sesuai koordinat lokasi dan mengirimkannya ke basis data. Kemudian dari data sampel yang sudah diambil, dilakukan evaluasi lokasi yang memiliki cakupan sinyal access point dengan kekuatan lemah, setelah itu dilakukan penambahan access point di beberapa lokasi tersebut dan dilakukan pengambilan ulang data sampel.

Setelah proses penambahan access point dilakukan, maka proses penentuan lokasi pengguna bisa dilakukan dengan mengirim data-data RSS di lokasi pengguna ke server untuk diproses dengan clustering Filtered KNN.

IV. SOFTWARE SIMULASI

Ada dua program yang digunakan untuk bisa mensimulasikan robot nya ini, yaitu CoppeliaSim dan Python.

CoppeliaSim berguna untuk menampilkan simulasi dari robot nya ini beserta seluruh objek (rintangan) dan pergerakan dari robotnya, sementara Python digunakan untuk memproses algoritma yang digunakan pada robotnya yang selanjutnya akan mengatur pergerakan robot yang ditampilkan pada CoppeliaSim.

A. Prosedur Simulasi CoppeliaSim

Untuk bisa menjalankan sebuah simulasi robot pengantar obat ini, pada bagian prosesing algoritma nya kami menggunakan Python3 (Backend). Sedangkan untuk menampilkan visualisasi gerakan robot nya kami menggunakan software CoppeliaSim (Frontend). Agar dapat menghubungkan keduanya, kami memanfaatkan komunikasi Frontend - Backend menggunakan API (Application Programming Interface).

API berfungsi sebagai perantara yang menghubungkan antara program Python yang telah dibuat dengan rancangan simulasi pada CoppeliaSim. Secara spesifik API pada CoppeliaSim terdapat sebuah middleware bernama BlueZero (b0) yang bersifat opensource sehingga bisa diakses oleh banyak Bahasa pemrograman yaitu C++, Java, Matlab, Lua, dan juga Python.

Pada simulasinya, dibuatkan sebuah target yang sudah diambil koordinat nya, dan juga sebuah robot yang koordinatnya juga sudah disiapkan pada program, yang selanjutnya robot akan berusaha untuk mencapai koordinat target dengan menelusuri ruangan sesuai dengan algoritma DWA yang sudah dijelaskan diatas.

B. API berbasis B0

Agar Python bisa berkomunikasi dengan CoppeliaSim, dibutuhkan sebuah API yang bisa menghubungkan keduanya, yang akan mempermudah pengendalian robot pada CoppeliaSim melalui Python.

Seluruh sensor dan data yang didapatkan oleh robot berdasarkan simulasi pada CoppeliaSim juga bisa dikirimkan pada program Python menggunakan API ini yang membuat kami lebih leluasa untuk memproses seluruh datanya pada Python.

Untuk pengaplikasian API nya pada Python, kami menggunakannya dengan cara *synchronous* agar tahapan dari pemrosesan data, pemrosesan algoritma, dan pengendalian robot terjadi secara berurutan sehingga tidak akan terjadi *race condition* pada program yang dibuat.

C. CoppeliaSim Script

Kami menggunakan beberapa *embedded script* yang kami masukan langsung kedalam robotnya pada simulasi CoppeliaSim untuk mempermudah pengambilan data, dikarenakan ada beberapa hal yang tidak bisa dilakukan pada Python dan/atau lebih efisien dan cepat untuk dilakukan langsung pada CoppeliaSim nya.

Contohnya seperti pengambilan koordinat objek (rintangan) yang berhasil di deteksi oleh sensor ultrasonik, kami menggunakan fungsi berikut untuk mengolah datanya secara langsung pada simulasi CoppeliaSim nya (menggunakan bahasa Lua):

```
1 function getObstacleCoord(dP)
2     local matrix=sim.getObjectHandle('sensor'),-1)
3     local absoluteDP=sim.multiplyVector(matrix,dP)
4
5     return absoluteDP
6 end
```

Listing 1. Fungsi Pengambilan Koordinat Collision Object

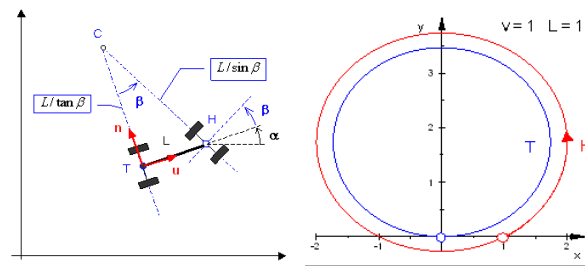
V. ANALISA DAN KESIMPULAN

Dalam penelitian ini, kami menggunakan algoritma Dynamic Window Approach (DWA) sebagai *path planning* serta digunakan untuk menghindari rintangan dalam *local planning*.

A. Pengujian Dynamic Window Approach untuk pathfinding Robot

B. Pengujian model steering robot

Berdasarkan perhitungan matematis dan pengujian dari simulasi, model *steering* roda hanya bisa berbelok maksimal 20 derajat sampai -20 derajat. Sedangkan roda belakang tidak melakukan perputaran. Jika diambil garis lurus berdasarkan roda depan dan roda belakang, maka titik pertemuan dari garis lurus itu akan menjadi pusat perputaran dari robot. Robot akan mencapai titik semula dan menghasilkan lintasan seperti pada gambar 16.



Gambar 16. *Steering model.*

Pada desain robot ini kami menggunakan algoritma *Dynamic Window Approach* agar robot dapat menemukan jalur paling efisien untuk mencapai tujuan yang diinginkan. kemudian untuk *obstacle avoidance* atau untuk menghindari rintangan, robot yang kami desain menggunakan sensor ultrasonik sehingga pada jarak tertentu robot akan mengenali adanya rintangan yang terdapat didepan maupun disamping robot.

Kemudian dari hasil yang kami dapat pada perancangan robot ini didapatkan kelemahan dari robot yang kami desain adalah apabila adanya objek yang bergerak seperti manusia, maka sistem akan menyimpulkan bahwa objek tersebut adalah rintangan, kemudian ketika sistem mencari jalur baru tetapi objek bergerak tersebut juga ikut bergerak kearah sensor, maka sistem tidak dapat mengganti jalur ke jalur sebelumnya. padahal pada jalur sebelumnya sudah tidak ada penghalang atau rintangan karena objek tersebut sudah bergerak.

REFERENSI