Terbit online pada laman web jurnal: **http://journal.isas.or.id/index.php/JASENS**

JOURNAL OF APPLIED SMART ELECTRICAL NETWORK AND SYSTEMS (JASENS)

Vol. 3 No. 1 (2022) 16 - 24 ISSN Media Elektronik: xxxx-xxxx

Implementasi Algoritma *Potential Field Obstacle Avoidance* Pada Robot Beroda 4 *Omni Wheels* Simetris

Kubad Nurhalim[1], Indrazno Siradjuddin[2], Ratna Ika Putri[3]
[1,2,3] Teknik Elektronika, Teknik Elektro, Politeknik Negeri Malang
[1]kubadnurhalim01@gmail.com* , [2]indrazno@polinema.ac.id, [3]ratna.ika@polinema.ac.id

Abstract

In recent years, one of the growing research topics is the field of robotics. Mobile robot technology is used to walk at the intended point and develop again to avoid obstacles when getting to that point there are obstacles. The main problem of avoiding obstacles on the intended route is the limited movement of the robot in avoiding these obstacles. With various considerations, such as problem boundaries, goals to avoid obstacles, reach the intended point and the robot does not stop in areas where there are obstacles. Therefore, a mobile robot was developed using the Potential Field algorithm method where the movement of the robot in avoiding obstacles applies the principle of attraction and repulsion like a magnet. In this study, using an attractive constant of 0.2 and a repulsive constant of 0.52, where the robot managed to avoid obstacles when going to the intended point. The results of the first test on the robot when tested directly, the first test using 1 obstacle obtained an error value of x = 0.2 m, y = 0.1 m and theta = 0.7°. The second test using 2 obstacles, the data obtained error values of x = 0.058 m, y = 0.15 m and theta = 1.4°.

Keywords: Potential Field, Obstacle Avoidance, Local Minima, Mobile Robot

Abstrak

Dalam beberapa tahun terakhir ini, salah satu topik penelitian yang berkembang adalah bidang robotika. Teknologi *mobile robot* digunakan untuk berjalan ke titik yang di tuju dan berkembang lagi untuk menghindari rintangan ketika menuju titik tersebut terdapat rintangan. Permasalahan utama dari menghindari rintangan yang berada di rute yang dituju adalah terbatasnya pergerakan robot dalam menghindari rintangan tersebut. Dengan berbagai pertimbangan yang ada seperti batasan masalah, tujuan untuk menghindari rintangan, mencapai titik yang dituju serta robot tidak berhenti pada daerah yang terdapat rintangan. Oleh sebab itu, dikembangkan *mobile robot* menggunakan metode algoritma *Potential Field* di mana pergerakan robot dalam menghindari rintangan menerapkan prinsip gaya tarik-menarik dan tolak-menolak layaknya magnet. Pada penelitian ini menggunakan konstanta *attractive* sebesar 0.2 dan konstanta *repulsive* sebesar 0.52, di mana robot berhasil menghindari rintangan ketika menuju titik yang dituju. Hasil pengujian pertama pada robot ketika diuji secara langsung, pengujian pertama menggunakan 1 rintangan didapatkan data nilai galat sebesar *x= 0.2 m, y = 0.1 m* dan *theta = 0.7°*. Pengujian kedua menggunakan 2 buah rintangan didapatkan data nilai galat sebesar *x= 0.058 m, y = 0.15 m* dan *theta* = 1.4°.

Kata kunci: *Potential Field, Obstacle Avoidance, Local Minima, Mobile Robot*

Diterima Redaksi : 25-03-2022 | Selesai Revisi : 15-06-2022 | Diterbitkan Online : 30-06-2022

1. Pendahuluan

Perkembangan teknologi yang terjadi saat ini adalah hadirnya robot lengan pemindah barang, *flying robot* atau *drone*, *mobile robot. Mobile robot* memiliki definisi sebagai robot yang mampu bergerak ke segala arah. Dalam sistem pergerakan robot membutuhkan sistem algoritma yang tepat, saat ini sistem algoritma yang digunakan *mobile robot* adalah sistem pergerakkan *differential drive.* Di mana sistem pergerakkan ini memiliki keterbatasan dalam pergerakkan robot karena hanya bergerak maju dan belok tetapi tidak mampu bergerak ke segala arah atau bisa disebut dengan robot *non-holonomic.* [1] Oleh sebab itu, dikembangkan sebuah robot yang dapat bergerak ke segala arah dalam bidang *kartesian* x-y tanpa perlu haluan saat arah hadap berubah, salah satunya adalah robot *omnidirectional* atau biasa disebut robot *holonomic* yang mempunyai mobilitas tinggi. [2]

Penentuan posisi robot adalah salah masalah utama pada pembuatan robot otomatis. Apabila robot tidak mengetahui di mana posisinya, tindakan robot selanjutnya akan sulit untuk ditentukan. Salah satu cara untuk mengetahui posisi robot adalah dengan menggunakan rotary encoder untuk mengetahui seberapa jauh robot sudah berpindah dari posisi awalnya dengan dilengkapi sensor kompas digital.

Namun cara ini mudah terpengaruh slip sehingga menghasilkan error yang nilainya selalu terintegral sehingga semakin lama semakin besar. [3]

Dalam bergerak menuju titik tujuan, robot bergerak tanpa menabrak objek apapun disekitar jalur yang akan dilalui robot. Oleh karena itu, diperlukan pendekatan untuk perluasan masalah kontrol gerak robot dalam mengenali objek hambatan ketika menuju titik tujuan dan dapat menghindari terjadinya tabrakan [4]. Sejauh ini beberapa sistem kontrol dan strategi navigasi menuju titik tujuan dengan kemampuan penghindaran terhadap tabrakan telah diusulkan dalam berbagai literatur. Metode APF telah diperluas untuk menghindari rintangan dengan menggunakan medan potensial buatan yang berubah-ubah terhadap waktu [5]. Dari penelitian [6] [7] [8] menunjukkan bahwa, metode APF adalah metode yang sangat sederhana, fleksibel, dan ringan secara komputasi. Peran APF adalah memberikan gaya tolak ketika robot mendekati *threshold* objek hambatan secara *real time*. Pendekatan dapat digunakan dalam lingkungan objek hambatan statis, tetapi dapat secara efisien digunakan dalam lingkungan objek hambatan dinamis [9]. Metode APF memiliki kelemahan yaitu terdapatnya local minima ketika robot telah memasuki daerah rintangan, di mana menyebabkan robot terjebak dan tidak berhasil menghindari rintangan [10].

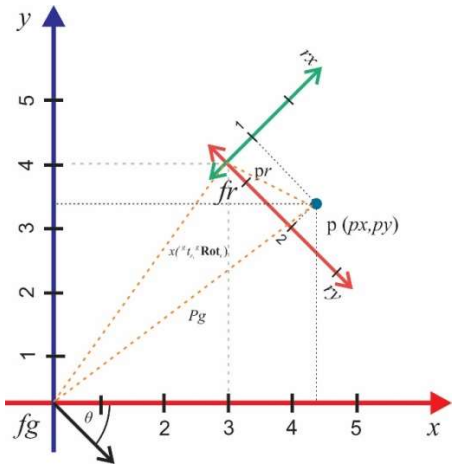
Pada penelitian ini dirancang *mobile robot* dengan dimensi *base* robot berukuran 25cm x 25cm serta ukuran omni wheels yang dipakai menggunakan ukuran diameter 60mm dan robot ini dapat menghindari rintangan ketika bergerak menuju titik tujuan yang telah ditentukan dengan menggunakan roda omni yang dapat bergerak ke segala arah untuk memudahkan robot bermanuver dan tidak memakan waktu banyak. Perhitungan potential field digunakan robot untuk menghindari rintangan ketika menuju titik tujuan dan tidak terjebak pada *local minima*. Penelitian ini juga melakukan uji nyata metode algoritma potensial field secara *real-time.*

2. Metode Penelitian

2.1 Kinematik Robot Beroda 4 Omni wheels

Ketika robot bergerak menuju titik tujuan akan berpindah posisi secara translasi dan rotasi, dimana translasi merupakan perputaran objek terhadap titik pusat [11]. Gerakan ini disebut dengan trasnformasi geometri 2 dimensi.

Jika kita memperhatikan gambar 1 terdapat titik *p*, titk tersebut dapat diukur melalui 2 frame yang berbeda yaitu terhadap *fg* (*global frame*) dengan notasi (${}^{g}x_{p,}$ ${}^{g}y_{p}$) maupun *fr* dengan notasi (${}^{r}x_{p,}$ ${}^{r}y_{p}$). Dengan menjadikan *fg* sebagai *global frame* sehingga dapat diketahui posisi koordinat *P* terhadap *fg* berdasarkan titik koordinat *P* terhadap *fr* atau sebaliknya.



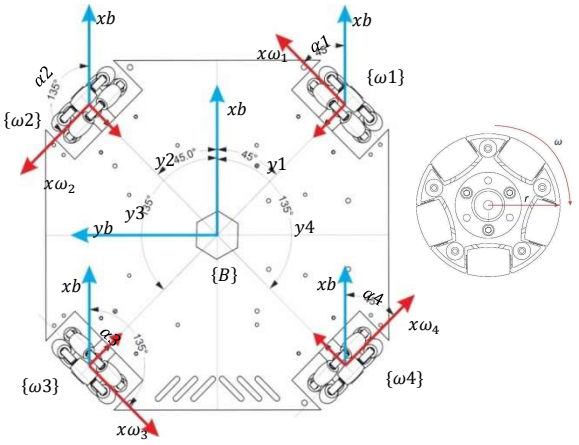
Gambar 1. Transformasi geometri 2 dimensi

Pada gambar 1. Terdapat titik *P*(${}^{r}x_{p,}$ ${}^{r}y_{p}$) terhadap *fr,* posisi titik *P*(${}^{g}x_{p,}$ y_{p}) terhadap *fg*. Selain itu frame *fr* memiliki titik pusat yang bertranslasi terhadap *global frame* (*fg*) dengan notasi (t_r^g). Sehingga nilai rotasi frame *fr* terhadap *global frame* (*fg*) dapat dinotasikan sebagai *fg* (^{g}**ROT**r). Sehingga dari gambar 1 di dapatkan persamaan rotasi dan translasi

$$\boldsymbol{P}_g = {}^{g}\boldsymbol{Rot}_r \,.\, \boldsymbol{P}_r + {}^{g}\boldsymbol{t}_r \qquad (1)$$

$$\begin{bmatrix} {}^{g}x_p \\ {}^{g}y_p \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} {}^{r}x_p \\ {}^{r}y_r \end{bmatrix} + \begin{bmatrix} {}^{g}x_r \\ {}^{g}x_r \end{bmatrix} \quad (2)$$

Pada perhitungan kinematic ini dibutuhkan beberapa variabel seperti roda yang digunakan (W1, W2, W3, W4), kecepatan angular ($\omega 1, \omega 2, \omega 3, \omega 4$) dan linier (v1, v2, v3, dan v4). Selain itu terdapat α1, α2, α3, dan α4 yang merupakan sudut yang terbentuk antara arah depan roda dengan arah hadap robot.



Gambar 2. Permodelan Robot Beroda 4 Omni wheels

Gambar 2 merupakan permodelan robot beroda 4 pada penelitian ini, dimana nilai dari γ1 = 45°, γ2 = 45° , γ3 = 135°, γ4 = 135°, α1 = 45° , α2 = 135°, α3 = -45°, α4 = -135°, l = 0.13 m, r = 0,03 m. sehingga untuk kecepatan anguler tiap roda bisa dirumuskan menjadi

$$v = r.\omega \quad (3)$$

$${}^{R}\dot{x}_R = r\,(\omega_1 \cos\theta_1 + \omega_2 \cos\theta_2 + \omega_3 \cos\theta_3 + \omega_4 \cos\theta_4) \quad (4)$$

$${}^{R}\dot{y}_R = r\,(\omega_1 \sin\theta_1 + \omega_2 \sin\theta_2 + \omega_3 \sin\theta_3 + \omega_4 \sin\theta_4) \quad (5)$$

$${}^{R}\dot{\theta}_R = r\left(\tfrac{1}{k}\,\omega_1 + \tfrac{1}{k}\,\omega_2 + \tfrac{1}{k}\,\omega_3 + \tfrac{1}{k}\,\omega_4\right) \quad (6)$$

Dengan nilai konstanta yang telah didapatkan, sehingga konstanta tersebut dapat dimasukkan kedalam matriks Jacobian. Persamaannya menjadi:

$${}^{R}\dot{\boldsymbol{x}}_R\,(t) = {}^{R}\boldsymbol{J}_R(\mathrm{t})\,.\,\boldsymbol{\omega}(t) \quad (7)$$

$$\begin{bmatrix} {}^{R}\dot{x}_R\,(t) \\ {}^{R}\dot{y}_R(t) \\ {}^{R}\dot{\theta}_R(t) \end{bmatrix} = 0.03 \begin{bmatrix} \cos(45) & \cos(135) & \cos(-45) & \cos(-135) \\ \sin(45) & \sin(135) & \sin(-45) & \sin(-135) \\ \frac{1}{0.013} & \frac{1}{0.013} & \frac{1}{0.013} & \frac{1}{0.013} \end{bmatrix} \begin{bmatrix} \omega1(t) \\ \omega2(t) \\ \omega3(t) \\ \omega4(t) \end{bmatrix} \quad (8)$$

Perlu diketahui persamaan 3.2 tersebut merupakan persamaan *forward velocity kinematic*, namun robot juga memiliki titik acuan terhadap *global frame*. Sehingga persamaan tersebut dikalikan dengan matrik rotasi robot terhadap *global frame*. Persamaan tersebut menjadi:

$${}^{g}\dot{\boldsymbol{x}}_R\,(t) = {}^{R}\boldsymbol{Rot}_R(\mathrm{t})\,.\,{}^{R}\boldsymbol{J}_R(\mathrm{t})\,.\,\boldsymbol{\omega}(t) \quad (9)$$

$$\begin{bmatrix} {}^{R}\dot{x}_R\,(t) \\ {}^{R}\dot{y}_R(t) \\ {}^{R}\dot{\theta}_R(t) \end{bmatrix} = 0.03 \begin{bmatrix} \cos(\theta(t)) & -\sin(\theta(t)) & 0 \\ \sin(\theta(t)) & -\cos(\theta(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \\ \frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \\ \frac{1}{0.013} & \frac{1}{0.013} & \frac{1}{0.013} & \frac{1}{0.013} \end{bmatrix} \begin{bmatrix} \omega1(t) \\ \omega2(t) \\ \omega3(t) \\ \omega4(t) \end{bmatrix} \quad (10)$$

Agar robot dapat mencari nilai kecepatan putar masing masing roda, maka digunakanlah persamaan *invers velocity kinematic* seperti pada persamaan :

$${}^{g}\dot{\boldsymbol{x}}_R.\,{}^{R}\boldsymbol{J^{-1}}_R = {}^{R}\boldsymbol{J}_R.\,{}^{R}\boldsymbol{J^{-1}}_R\,.\,\boldsymbol{\omega}_R \quad (11)$$

$$\boldsymbol{\omega}_R = {}^{g}\dot{\boldsymbol{x}}_R.\,{}^{R}\boldsymbol{J^{-1}}_R \quad (12)$$

$$\begin{bmatrix} \omega1(t) \\ \omega2(t) \\ \omega3(t) \\ \omega4(t) \end{bmatrix} = 0.03 \begin{bmatrix} \cos(\theta(t)) & \sin(\theta(t)) & 0 \\ -\sin(\theta(t)) & \cos(\theta(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \\ \frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \\ \frac{1}{0.013} & \frac{1}{0.013} & \frac{1}{0.013} & \frac{1}{0.013} \end{bmatrix}^{T} \begin{bmatrix} {}^{R}\dot{x}_R\,(t) \\ {}^{R}\dot{y}_R(t) \\ {}^{R}\dot{\theta}_R(t) \end{bmatrix} \quad (13)$$

Agar robot bisa berhenti maka dibutuhkan sistem untuk membuat robot melambat ketika mendekati titik tujuan, maka sistem kendali dapat dirumuskan dengan persamaan pendekataan *error* :

$$e_t = \mathrm{p}_t^* - P_t \quad (14)$$

Persamaan *error* ini diturunkan sebagai berikut :

$$\dot{e}_t = -\lambda e^{-\lambda t} \quad (15)$$

$$\dot{p}_t = \lambda(\mathrm{p}_t^* - p_t) \quad (16)$$

Hasil subtitusi persamaan *invers velocity kinematic*, di dapatkan kecepatan gerak motor sehingga persamaan *error* sebagai berikut :

$$\omega(t) = J^{-1}.\,{}^{g}\dot{\boldsymbol{x}}_R \quad (17)$$

$$\omega(t) = J^{-1}\,\lambda(\mathrm{p}_t^* - p_t) \quad (18)$$

2.1 Algoritma Potential Field

Dalam dasar skenario bidang potensial, suatu robot dalam mendeteksi suatu rintangan sebagai medan potensial tolak-menolak dan titik tujuan sebagai medan tarik-menarik [12]. Dapat di misalkan x = $(x, y)^T$ mempresentasikan dari posisi robot, maka bidang di mana robot bergerak dapat dilambangkan sebagai U(x), di mana :

$$\mathrm{U(x)} = \mathrm{U_{att}(x)} + \mathrm{U_{rep}(x)} \quad (19)$$

Dan $\mathrm{U_a(x)}$ menunjukkan medan Tarik-menarik dan $\mathrm{U_r(x)}$ menyatakan medan tolak-menolak. Dapat ditulis sebagai :

$$\mathrm{U(x)} = \mathrm{U_{att}(x)} + \sum_{n=1}^{N} \mathrm{U_{rep}}^{(n)}\mathrm{(x)} \quad (20)$$

Di mana N merupakan jumlah total rintangan yang berada di lintasan robot bergerak. Gaya yang dihasilkan ketika robot bergerak adalah negative gradient dari bidang potensial, maka :

$$\mathrm{F(x)} = -\underbrace{\nabla_{\mathrm{x}}\mathrm{U_a(x)}}_{\mathrm{F}_{att}\mathrm{(x)}} - \underbrace{\nabla_{\mathrm{x}}\mathrm{U_r(x)}}_{\mathrm{F}_{rep}\mathrm{(x)}} \quad (21)$$

Di mana $\mathrm{F_{att}(x)} = (\mathrm{F}_{att,x}, \mathrm{F}_{att,y})^{\mathrm{T}}$ adalah vektor gaya tarik-menarik dan $\mathrm{F_{rep}(x)} = (\mathrm{F}_{rep,x}, \mathrm{F}_{rep,y})^{\mathrm{T}}$ adalah vektor gaya tolak-menolak. Pada bidang potensial Tarik-menarik dapat didefinisikan bahwa proposional dari nilai kuadrat jarak antara posisi robot dengan posisi tujuan d_a^2. Semakin jauh posisi robot dari posisi tujuan, maka semakin kuat bidang potensial tarik-menarik. Fungsi bidang potensial tarik-menarik dapat dituliskan sebagai :

$$\mathrm{U_{att}(x)} = \frac{1}{2} k_{att} {d_a}^2 \quad (22)$$

$$= \frac{1}{2} k_{att} ||(X^* - X)||^2 \quad (23)$$

Di mana $\boldsymbol{k_{att}}$ adalah konstanta dari bidang potential Tarik-menarik dan $\boldsymbol{X}^*$adalah posisi titik tujuan. Asumsikan bahwa posisi tujuan adalah statis atau tidak berpindah tempat, maka gaya Tarik-menarik adalah :

$$\mathrm{F}_{att}\mathrm{(x)} = k_{att}\|\mathrm{X}^* - X\| \quad (24)$$

$$\mathrm{F}_{att}\mathrm{(x)} = k_{att}\sqrt{(x^* - x)^2 + (y^* - y)^2} \quad (25)$$

Di mana d_a= $\|\mathbf{X}^* - \boldsymbol{X}\| = \sqrt{(x^* - x)^2 + (y^* - y)^2}$ adalah jarak *Euclidean* antara posisi titik tujuan $x^* = (x^*, y^*)^T$ dengan posisi robot x = $(x, y)^T$.

Berbeda dengan medan potensial Tarik-menarik, medan potensial tolak-menolak lebih kuat ketika posisi robot lebih dekat dekat dengan rintangan. Setiap rintangan yang lebih jauh kemungkinana tidak akan berkontribusi pada bidang potensial total, sehingga jika jarak robot dari rintangan ke-n $d_{o_{(n)}}^{(n)}$ lebih besar dari nilai ambang (*threshold*) τ maka bidang potensial diabaikan, $U_{rep}{}^{(n)}\mathrm{(x)} = 0$. Jika tidak, medan potensial tolak-menolak sebanding dengan kebalikan dari jarak

antara robot dan rintangan. Fungsi tolak-menolak dapat definisikan sebagai

$$U_{rep}{}^{(n)}(x) = \begin{cases} \frac{1}{2}k_{rep}\left(\frac{1}{d_o{}^{(n)}}\right)^2 & if \quad d_o{}^{(n)} \leq \tau \\ 0 & if \quad d_o{}^{(n)} > \tau \end{cases} \quad (26)$$

Dimana, $d_o{}^{(n)} = \left\|\mathbf{x_o}^{(n)} - \mathbf{X}\right\| = \sqrt{(x_o{}^{(n)} - x)^2 + (y_o{}^{(n)} - y)^2}$ adalah jarak *Euclidean* antara posisi robot $X = (x, y)^T$ dan posisi rintangan ke-n $\mathrm{X}_o{}^{(n)} = \left(x_o{}^{(n)},\ y_o{}^{(n)}\right)^{\mathrm{T}}$. Konstanta bidang tolak-menolak dapat dilambangkan dengan k_r, diasumsikan setiap rintangan memiliki karakteristik bidang potensial tolak-menolak yang sama. Asumsikan bahwa posisi rintangan adalah statis atau tidak berpindah tempat, maka gaya tolak-menolak adalah

$$\mathrm{F}_{rep}(\mathrm{x}) = \begin{cases} \frac{k_{rep}}{(d_o{}^{(n)})^3} & if \quad d_o{}^{(n)} \leq \tau \\ 0 & if \quad d_o{}^{(n)} > \tau \end{cases} \quad (27)$$

2.2 Local Minima

Masalah utama menggunakan metode *Potential Field* adalah munculnya *local minima* yang di mana kondisi ini menghalangi robot dan mencegah untuk mencapai ke titik tujuan karena hambatan konfigurasi tertentu. Konfigurasi yang menghasilkan gaya potensial nol (0) yang memaksa robot berhenti sebelum mencapai posisi tujuan, atau membuat robot bergerak mundur dan maju terus menerus. Ada beberapa solusi untuk keluar dari masalah local minima ini, berikut beberapa faktornya: Perhitungan $l_c = \frac{\|\mathbf{F}_a\|}{\|\mathbf{F}_r\| + 0.0001}$, di mana l_c merupakan kriteria local minima dan 0.0001 adalah angka untuk menghindari pembagian nol, Periksa apabila kriteria local minima $l_c \leq \tau_l$, di mana τ_l adalah nilai konstanta *threshold* atau ambang batas dan Jika $l_c \leq \tau_l$ adalah salah maka robot terus bergerak ke posisi titik tujuan, jika kondisi local minima diamati, maka gaya dalam menghindari rintangan dapat diterapkan untuk mengubah arah robot dalam menghindari rintangan.

2.3 Kontrol Robot Menghindari Rintangan

Sebelum melangkah lebih jauh dalam merancang kontrol untuk pergerakkan robot dalam menghindari rintangan, pertama-tama perlu menguraikan setiap gaya, gaya tarik-menarik dan tolak-menolak untuk memproyeksikan gaya pada sumbu x dan pada sumbu y. Oleh karena itu untuk penguraian gaya tarik-menarik dari persamaan 6 adalah sebagai berikut :

$$\mathrm{F}_{att,x} = k_{att}(x^* - x) \quad (18)$$

$\mathrm{F}_{a,x}$ adalah gaya tarik menarik pada arah sumbu x. Dengan persamaan yang sama, gaya tarik menarik dalam arah sumbu y didapatkan sebagai berikut :

$$\mathrm{F}_{att,y} = k_{att}(y^* - y) \quad (29)$$

Proyeksi dari gaya tolak menolak pada arah sumbu x dan sumbu y adalah sebagai berikut, perlu diingat bahwa pada persamaan *repulsive* hanya memperhitungankan kasus $d_o{}^{(n)} \leq \tau$

$$F_{rep,x}{}^{(n)} = k_{rep}\frac{(x - x_o{}^{(n)})}{(d_o{}^{(n)})^4} \quad (30)$$

Di mana $F_{rep,x}{}^{(n)}$,menunjukkan daya tolak menolak dari rintangan ke-n dalam arah sumbu x. Dari persamaan yang sama berlaku untuk mendapatkan gaya tolak menolak pada arah sumbu y sebagai berikut :

$$F_{rep,y}{}^{(n)} = k_{rep}\frac{(y - y_o{}^{(n)})}{(d_o{}^{(n)})^4} \quad (31)$$

Jadi, untuk total gaya dapat dituliskan :

$$\mathbf{F}\ (\mathbf{x}) = (F_x, F_y)^{\mathbf{T}} \quad (32)$$

$$F_x = \mathrm{F}_{att,x}(\mathrm{X}) + \sum_{n=1}^{N} F_{rep,x}{}^{(n)}\ (\mathrm{X}) \quad (33)$$

$$F_y = \mathrm{F}_{att,y}(\mathrm{X}) + \sum_{n=1}^{N} F_{rep,y}{}^{(n)}\ (\mathrm{X}) \quad (34)$$

Dari nilai Fx dan Fy yang telah didapatkan, dari 2 buah gaya yang berbeda tersebut dapat dihitung sudut agar robot bisa menghindari rintangan dari nilai Fx dan Fy yang telah diketahui.

Berikut arah gaya dapat di rumuskan sebagai berikut :

$$\theta = arctan\ \frac{F_y}{F_x} \quad (35)$$

Algoritma kontrol pada robot yang paling sederhana dirancang sedemikian rupa sehingga sinyal untuk kontrol kecepatan $\mathbf{V} = (v_x,\ v_y)^{\mathbf{T}}$ diberikan oleh gaya medan potensial total, maka

$$v_x = F_x \quad (36)$$

$$v_y = F_y \quad (37)$$

Maka *update* posisi pada robot dapat dituliskan persamaan sebagai berikut :

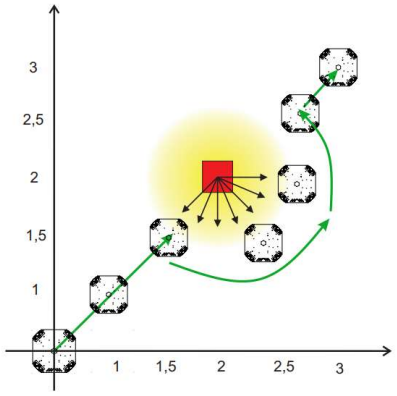
$$x(k) = x\ (k-1) + v_x \Delta t \quad (38)$$

$$y(k) = y\ (k-1) + v_y \Delta t \quad (39)$$

Di mana Δt adalah sampling time, $x(k)$ dan $y(k)$ adalah koordinat robot x dan y pada k-th waktu instan.

2.4 Prinsip Kerja Sistem

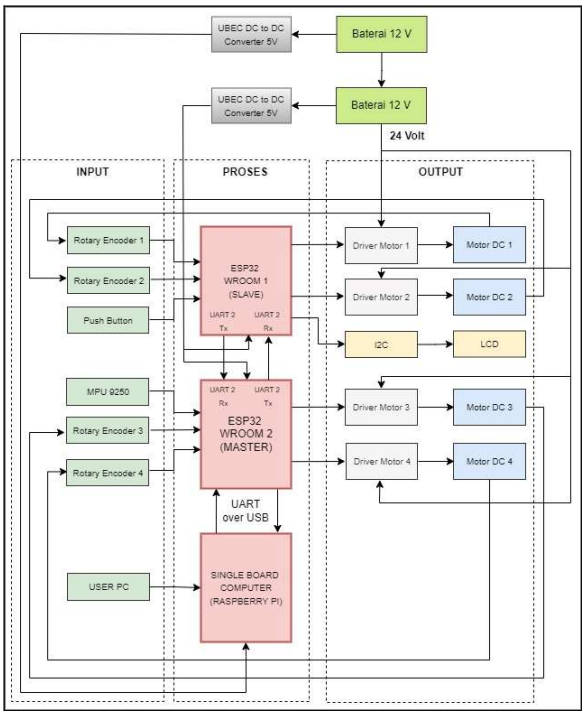
Algoritma robot di desain sedemikian rupa agar dapat bergerak menuju titik tujuan dan menghindari rintangan yang berada ditengah lintasan menuju titik tujuan, di mana posisi awal, titik tujuan dan rintangan telah diketahui posisinya terhadap *global frame* sebelumnya. Pada gambar 1 menunjukkan bahwa posisi awal robot dengan koordinat (0,0), posisi tujuan robot pada koordinat (3,3) dan posisi rintangan dengan koordinat (2,2) ditunjukkan dengan adanya kotak berwarna merah.



Gambar 1. Simulasi Prinsip Kerja Potential Field pada Robot Beroda 4 *Omnidirectional*

2.6 Diagram Blok Sistem

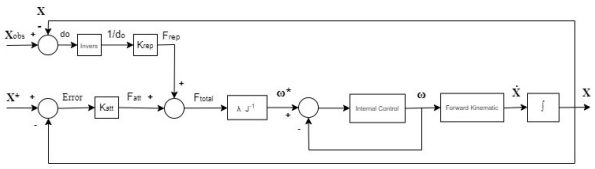
Pada diagram blok sistem yang terdapat pada gambar 3. Dari penelitian ini, diagram blok sistem dibagi menjadi 3 blok, yaitu input, proses dan output. Pada blok input terdapat Push Button, *Rotary Encoder* dan Sensor Kompas MPU9250 Sensor *Rotary Encoder* 1 dan 2 terhubungan dengan mikrokontroller ESP32 (*Slave*), pada *Rotary Encoder* 1 channel A terhubung pada GPIO 36 dan channel B terhubung pada GPIO 39. Untuk *Rotary Encoder* 2 channel A terhubung pada GPIO 34 dan channel B terhubung pada GPIO 35.



Gambar 5. Diagram Blok Sistem

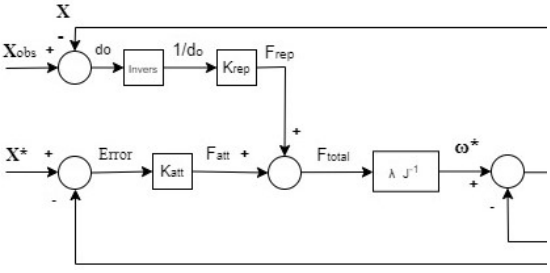
Sensor *Rotary Encoder* 3 dan 4 terhubungan dengan mikrokontroller ESP32 (*Master*). pada *Rotary Encoder* 1 channel A terhubung pada GPIO 36 dan channel B terhubung pada GPIO 39. Untuk *Rotary Encoder* 2 channel A terhubung pada GPIO 34 dan channel B terhubung pada GPIO 35. Perangkat proses yang digunakan pada penelitian ini yaitu sebuah *Single Board Computer* dan 2 buah mikrokontroller, yaitu Raspberry Pi 4 Model B dan ESP32 WROOM. Pemrosesan data dibagi kedalam dua bagian yaitu *low level control* dan *high level control. Low level control* dilakukan oleh mikrokontroller ESP32 (*Master*) yang mana data yang diproses digunakan untuk memerintahkan aktuator bergerak, mengirimkan data kecepatan putar motor (RPM) ke *Single Board Computer* melalui komunikasi serial (UART over USB). Sedangkan *high level control* dibatasi oleh *Single Board Computer* Raspberry Pi 4 Model B yang mana bertugas untuk menerima data kecepatan data putar motor serta melalukan *Interfacing* pada sensor kompas MPU9250, di mana data ini nantinya digunakan untuk perhitungan kinematik dan *potential field*.

2.5 Blok Diagram Kontrol



Gambar 2. Diagram Blok Kontrol

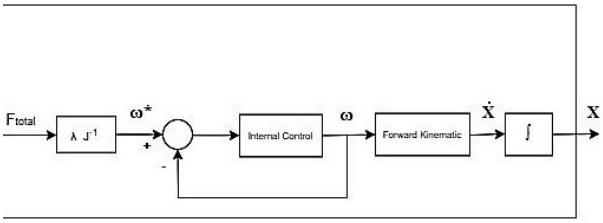
Pada gambar 2 diagram blok control diatas adalah di mana set point berupa **X*** merupakan posisi yang ingin dituju robot (x^*,y^*,θ^*), di mana x^*,y^* adalah koordinat tujuan robot dan θ^* merupakan orientasi (arah hadap) robot. **X** atau (x, y, θ) adalah posisi dan orientasi aktual robot saat ini, yang nilainya di estimasi menggunakan persamaan integral terhadap kecepatan translasi robot $\dot{x}$ atau ($\dot{x}$, $\dot{y}$, $\dot{\theta}$). Adapun kecepatan translasi robot aktual ($\dot{x}$) diukur secara tidak langsung menggunakan persamaan *Forward Velocity Kinematic* untuk robot 4 roda omni. Persamaan ini mengonversi kecepatan sudut 4 buah roda robot (ω1, ω2, ω3 dan ω4) dan menghasilkan kecepatan translasi robot ($\dot{x}$).



Gambar 3. Diagram Blok Kontrol bagian a

Error posisi E merupakan hasil dari setpoint posisi **X*** dikurangi posisi aktual **X**. Nilai Error posisi ini akan dikalikan dengan nilai k_a dan menghasilkan keluaran berupa gaya *attractive* atau F_{att}. Input kontrol ini mendapatkan input dari posisi *obstacle* yang di notasikan dengan Xobs, Xobs dan X akan dijumlahkan dan hasil penjumlahan ini akan di inverskan agar mendapatkan nilai 1/Do.

Hasil dari X_{obs} dan X yang telah di inverskan ini akan dikalikan juga dengan konstanta *repulsive* atau K_{rep} agar mendapatkan nilai F_{rep}. Nilai F_{total} **(F)** merupakan hasil penjumlahan dari nilai $Fatt$ dan $Frep$, di mana nilai F_{total} ini merupakan nilai untuk pergerakkan robot dalam menjalankan algoritma potential field.



Gambar 4. Diagram Blok Kontrol bagian b

Nilai F_{total} dikalikan dengan dikalikan λJ-1 yang merupakan persamaan *Invers Kinematic* akan menghasilkan setpoint kecepatan roda robot (***ω*** ∗), selanjutnya Internal Kontroller digunakan untuk menstabilkan kecepatan roda robot dalam kontrol loop tertutup sehingga robot dapat bergerak ke posisi yang ditentukan. Pembacaan kecepatan roda secara aktual dilakukan oleh encoder, sensor encoder inilah yang memberikan *feedback* berupa nilai kecepatan putar roda.

2.7 Rancangan Mekanik

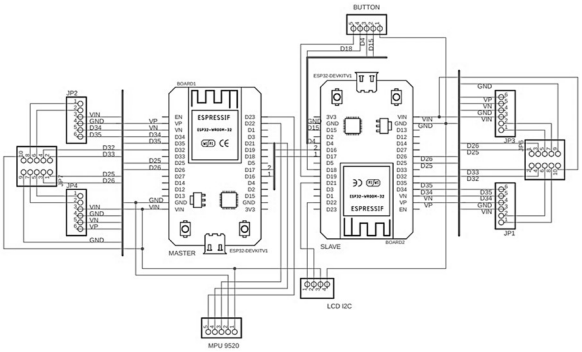
Gambar 6 merupakan desain mekanik robot dari samping kiri belakang yang, base *mobile robot* dirancang menggunakan bahan plat aluminium 3mm. Pada perancangan mekanik menggunakan 4 roda Omni wheels dengan dimensi 60 mm & menggunakan motor DC PG28.

Gambar 6. Gambar Mekanik

Pada bagian atas robot terdapat LCD untuk menampilkan menu, push button berfungsi sebagai alat pemilihan menu pada LCD dan saklar sebagai pemutus tegangan motor dan catu daya rangkaian elektronik. Pada bagian tengah dalam robot terdapat rangkaian *main board*, *power management* dan *single board computer* Raspberry Pi.

2.8 Rancangan Elektronik *Main Board*

Pada gambar 7 dibawah merupakan desain rangkaian elektronik *main board* yang berfungsi sebagai *low control* pada robot yang terdiri dari 2 ESP, yaitu ESP *Slave* dan ESP *Master*.

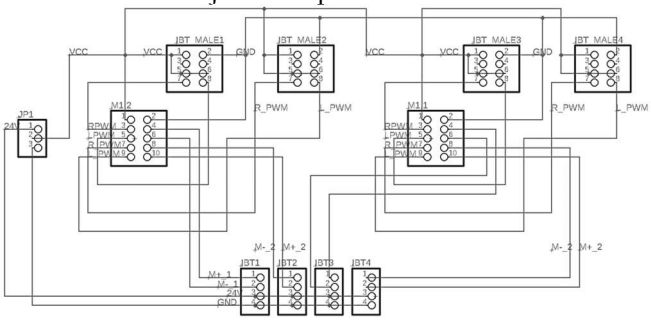


Gambar 7. Desain Rangkaian Elektronik *Main Board*

Pada ESP tersebut menggunakan komunikasi serial (UART) dalam pertukaran data antar ESP. Dimana ESP *Slave* sebagai pengontrol dari keluaran motor 1 dan motor 2, LCD 16x2 sebagai interface pada robot dan push button sebagai pemilihan beberapa fitur yang tersedia pada robot. ESP *Master* berfungsi sebagai pengontrolan data keluaran motor 3 dan motor 4 serta keluaran data kompas MPU9250.

2.9 Rancangan Elektronik *Driver* Motor DC *H-Bridge* IBT_2

Driver motor IBT_2 digunakan untuk mengendalikan arah dan kecepatan putar motor DC berdasarkan perintah dari mikrokontroler yang didapatkan dari perhitungan kinematic sebagai dasar pergerakkan robot dalam menentukan jalur tercepat.

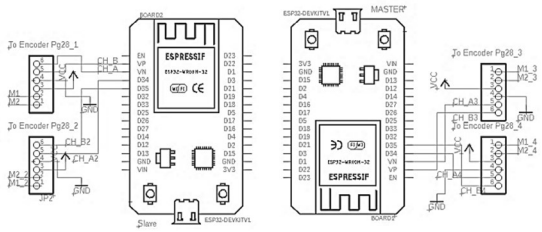


Gambar 8. Desain Rangkaian Skematik Driver Motor DC IBT_2

Driver motor IBT-2 merupakan *H-Bridge driver* yang menggunakan IC (*Integrated Circuit)* BTS7960, IBT-2 menggunakan masukkan sinyal PWM dengan rentang frekuensi 16 kHz. Pada driver ini dapat mengoperasikan arah putar motor searah jarum jam (*Clock Wise)* dan berlawanan dengan arah jarum jam *(Counter Clock Wise)*.

2.10 Rancangan Elektronik *Rotary Encoder*

Rotary encoder untuk robot ini berfungsi sebagai pembacaan nilai putaran kecepatan roda atau disebut dengan RPM (*Rotation per Minute*). Sehingga penggunaan *encoder* ini dapat digunakan untuk pembacaan kecepatan perubahan posisi robot.



Gambar 9. Desain Rangkaian Skematik Rotary Encoder

Jenis *encoder* yang digunakan adalah *incremental encoder*, dalam jenis ini terdapat *channel A* dan *channel B* yang akan menghasilkan pulsa dengan masing masing memiliki beda fase 90°. Perbedaan yang dimiliki tersebut digunakan untuk mengetahui arah putar roda CW atau CCW. Karena menggunakan 4 buah motor, sehingga memerlukan 8 buah pin untuk encoder. *Encoder* motor 1 dan 2 berada pada ESP32 yang bekerja sebagai *slave*, motor 3 dan 4 terhubung dengan ESP yang berfungsi sebagai *master.*

3. Hasil dan Pembahasan

3.1 Pengujian Algoritma *Potential Field Obstacle Avoidance* Secara *Real-time* Menggunakan 1 Buah Rintangan

Pada pengujian ini dilakukan dengan uji nyata secara *real-time* hanya menggunakan 1 buah rintangan, robot berada pada posisi koordinat [0.0, 0.0], titik posisi hambatan pada koordinat [1.0, 0.5] dan titik posisi tujuan pada koordinat [2.0, 1.0, 0°]**.**

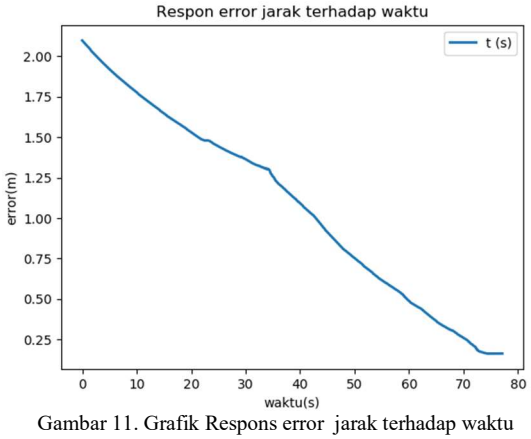
Gambar 10. Posisi awal robot dan posisi rintangan

Berikut merupakan table data pengujian pembacaan posisi robot dengan menggunakan algoritma potential field.

Tabel 1. Hasil Pengujian Algoritma Potential Field Obstacle Avoidance pada Robot Beroda 4 Omni wheels dengan 1 buah rintangan secara *Real-time*

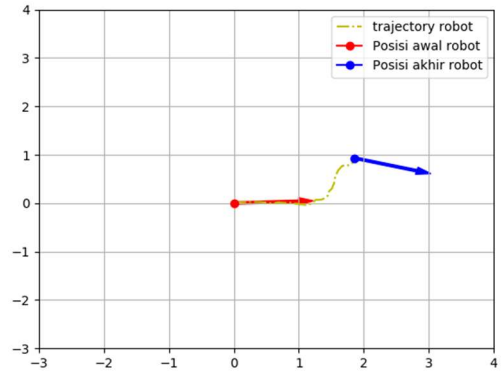
Pengujian Ke-*n*	Posisi Tujuan [*x**, *y**, *theta**]	Posisi Aktual [*x*, *y*, *theta*]	*Error* Posisi [*x*, *y*, *theta*]
1	[2.0, 1.0, 0°]	[1.8, 0.93, -0,2°]	[0.2, 0.07, 0.2°]
2	[2.0, 1.0, 0°]	[1.8, 0.85, -2,3°]	[0.2, 0.15, 2.3°]
3	[2.0, 1.0, 0°]	[1.8, 0.93, 0,2°]	[0.2, 0.07, 0.2°]
4	[2.0, 1.0, 0°]	[1.8, 0.87, -0,3°]	[0.2, 0.13, 0.3°]
5	[2.0, 1.0, 0°]	[1.8, 0.86, -0,5°]	[0.2, 0.14, 0.5°]
	Error rata-rata		[0.2, 0.1, 0.7°]

Tabel 1 merupakan table data pengujian robot menuju titik tujuan sekaligus menghindari rintangan, *error* rata-rata yang didapatkan dari pengujian yang dilakukan sebanyak 5 kali menunjukkan bahwa nilai *error* rata-rata dari pengujian sebanyak 5 kali ini sebesar *x= 0.058 m, y = 0.15 m* dan *theta =* 1.4°.

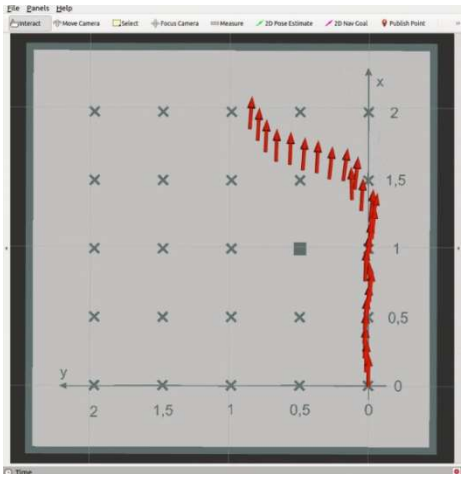


Gambar 11. Grafik Respons error jarak terhadap waktu

Pada gambar 11 menunjukkan bahwa waktu robot dalam menuju titik tujuan terhadap jarak, semakin mendekati titik tujuan robot mengalami *error* jarak yang semakin kecil atau dapat dikatakan berhenti di titik yang diperintahkan. Pada waktu ke 30*s*-40*s error* jarak mengalami sedikit kenaikan karena robot sedang menghindari rintangan. Grafik respons *error* jarak terhadap waktu ini membuktikan bahwa sistem berjalan sesaui yang diaharapkan atau stabil.



Gambar 12. Respons Trajectory robot dari Potential Field Obstacle Avoidance secara realtime

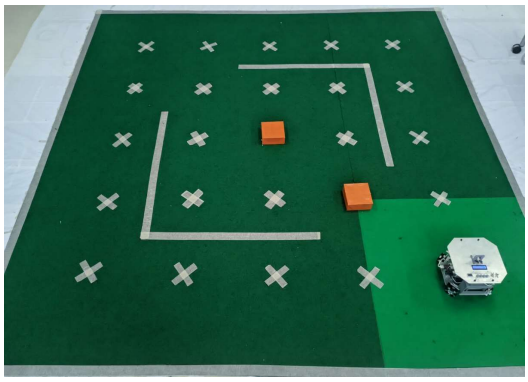


Gambar 13. Visualisasi pergerakkan robot pada Rviz

Pada gambar 13 dapat dilihat bahwa robot dapat menghindari rintangan menggunakan algoritma potential field. Ditunjukkan dengan adanya rintangan kotak pada koordinat (1.0, 0.5) yang tidak tertabrak oleh tanda panah atau robot itu sendiri.

3.2 Pengujian algoritma *potential field obstacle avoidance* secara *real-time* menggunakan 2 buah rintangan

Pada pengujian ini dilakukan dengan uji nyata secara *real-time* hanya menggunakan 2 buah rintangan, robot berada pada posisi koordinat [0.0, 0.0], titik posisi hambatan pada koordinat [0.5, 0.5] [1.0, 1.0] dan titik posisi tujuan pada koordinat [1.5, 1.5, 0°]**.**



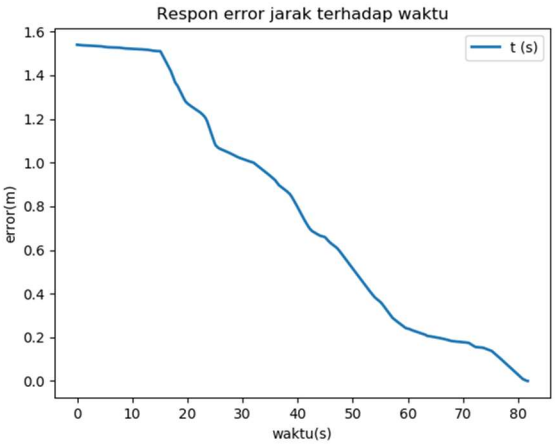
Gambar 14. Posisi awal robot dan posisi rintangan

Berikut merupakan table data pengujian pembacaan posisi robot dengan menggunakan algoritma potential field.

Tabel 2. Hasil Pengujian Algoritma Potential Field Obstacle Avoidance pada Robot Beroda 4 Omni wheels dengan 2 buah rintangan secara *Real-time*

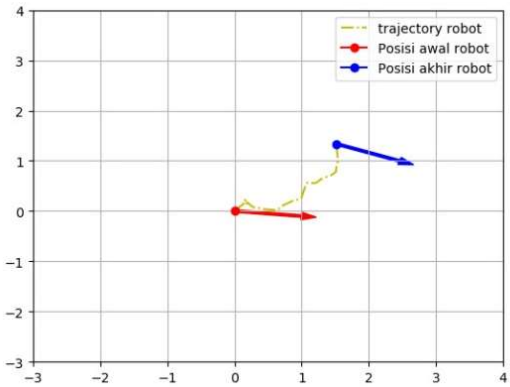
Pengujian Ke-*n*	Posisi Tujuan [*x**, *y**, *theta**]	Posisi Aktual [*x*, *y*, *theta*]	*Error* Posisi [*x*, *y*, *theta*]
1	[1.5, 1.5, 0°]	[1.51, 1.33, -0,4°]	[0.01, 0.17, 0.4°]
2	[1.5, 1.5, 0°]	[1.49, 1.32, -2,3°]	[0.01, 0.18, 2.3°]
3	[1.5, 1.5, 0°]	[1.44, 1.36, -3.2°]	[0.06, 0.14, 3.2°]
4	[1.5, 1.5, 0°]	[1.49, 1.41, 0,06°]	[0.2, 0.09, 0.06°]
5	[1.5, 1.5, 0°]	[1.49, 1.32, -1,04°]	[0.01, 0.18, 1.0°]
	Error rata-rata		[0.058, 0.15, 1.4°]

Tabel 1 merupakan table data pengujian robot menuju titik tujuan sekaligus menghindari rintangan, *error* rata-rata yang didapatkan dari pengujian yang dilakukan sebanyak 5 kali menunjukkan bahwa nilai *error* rata-rata dari pengujian sebanyak 5 kali ini sebesar *x= 0.058 m, y = 0.15 m* dan *theta =* 1.4°.



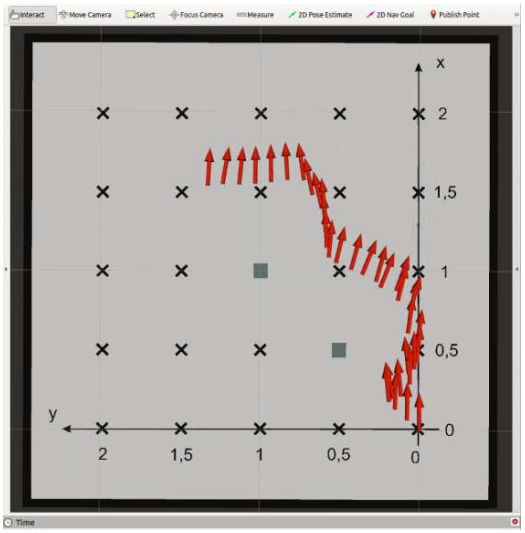
Gambar 15. Grafik Respons error jarak terhadap waktu

Pada gambar 15 menunjukkan bahwa waktu robot dalam menuju titik tujuan terhadap jarak, semakin mendekati titik tujuan robot mengalami *error* jarak yang semakin kecil atau dapat dikatakan berhenti di titik yang diperintahkan. Pada waktu ke 15*s*-50*s error* jarak banyak mengalami perubahan karena robot sedang menghindari 2 buah rintangan yang berdekatan. Grafik respons *error* jarak terhadap waktu ini membuktikan bahwa sistem berjalan sesaui yang diaharapkan atau stabil.



Gambar 16. Respons Trajectory robot dari Potential Field Obstacle Avoidance secara realtime

Dari gambar 16 didapatkan respons *trajectory* robot menghindari rintangan ketika menuju titik tujuan. Hasilnya robot dapat bergerak ke titik tujuan dengan baik, namun robot tidak dapat langsung menuju titik tujuan karena terdapat 2 buah rintangan ditengah jalur. Terdapat pergerakan robot menghindari rintangan.



Gambar 17. Visualisasi pergerakkan robot pada Rviz

Pada gambar 17 dapat dilihat bahwa pembacaan posisi akhir robot mengalami perubahan. Hal ini dapat disebabkan oleh beberapa hal, diantarnya selipnya roda pembacaan posisi, selip roda penggerak, tidak idealnya aktuator yang digunakan, serta efisiensi pemrosesan data. Hal ini masih masuk dalam titik toleransi karena nilai *error trajectory* tidak terlalu signifikan dan robot dapat dengan cepat memperbaiki posisinya ke titik tujuan sesuai perintah.

4. Kesimpulan

Pada penelitian ini parameter yang diperhatikan dalam menggunakan algoritma *potential field* adalah nilai k_{att} dan k_{rep}, di mana dalam penelitian ini menggunakan nilai k_{att} sebesar 0.2 dan k_{rep} sebesar 0.52. Nilai ini yang mempengaruhi gaya robot dalam menghindari rintangan dan menuju titik yang tuju.

Penerapan Algoritma Potential Field dalam menghindari rintangan telah berjalan sesuai dengan yang diinginkan. Di mana nilai galat rata-rata posisi dan arah hadap pada pengujian pertama sebesar *x= 0.2 m, y = 0.1 m* dan *theta = 0.7°*.pada pengujian kedua didapatkan nilai galat rata-rata posisi dan arah hadap sebesar *x= 0.058 m, y = 0.15 m* dan *theta =* 1.4°.

Ucapan Terimakasih

Terima kasih kepada ibu Ratna Ika Putri yang telah meluangkan waktu untuk membantu menyusun artikel ini hingga artikel ini dapat di publikasikan.

Daftar Rujukan

[1] E. M. Veri Hendrayawan, Nanang Sulistiyanto, 2014. Implementasi Invers Kinematics pada Sistem Pergerakan Mobile Robot Roda Mekanum. *Jurnal Mahasiswa Teknik Elektro Universitas Brawijaya*, 2 (6), pp.1-10.

[2] Hassani, I., Maalej, I., & Rekik, C. 2018. Robot path planning with avoiding obstacles in known environment using free segments and turning points algorithm. *Jurnal Hindawi Mathematical Problems in Engineering*, vol. 2018, pp. 1-13

[3] Rachmawan, A. 2017. *Penentuan Posisi Robot Sepak Bola Beroda Menggunakan Rotary Encoder dan Kamera*. Doctoral, Institut Teknologi Sepuluh Nopember.

[4] Kozlowski, K., & Kowalczyk, W. 2020. Control of Two-wheeled Mobile Robots Moving in Formation. *IFAC-PapersOnLine*, *53*(2), 9608-9613.

[5] Khatib, O. 1986. *Autonomous robot vehicles : Real-time obstacle avoidance for manipulators and mobile robots*. Springer, New York, NY.

[6] Hart, P. E., Nilsson, N. J., & Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4 (2), pp. 100-107.

[7] Stentz, A. 1997. In *Intelligent unmanned ground vehicles : Optimal and efficient path planning for partially known environments.* Springer, Boston, MA.

[8] LaValle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning.

[9] Sasi Kumar, G., Shravan, B., Gole, H., Barve, P., & Ravikumar, L. 2011. Path Planning Algorithms: A Comparative Study. *Space Transportation Systems Division: Houston, TX, USA*.

[10] Rostami, S. M. H., Sangaiah, A. K., Wang, J., & Liu, X. 2019. Obstacle avoidance of mobile robots using modified artificial potential field algorithm. *EURASIP Journal on Wireless Communications and Networking*, 2019 (1), pp. 1-19.

[11] M. F. Al Fattika, T. Winarno, I. Siradjuddin, Collision Avoidance pada Mobile Robot Menggunakan Kontrol Kinematik, Jurnal Elkolind, Vol. 8, No. 1, 2019

[12] Sirajuddin, Indrazno. 2022. Obstacle Avoidance and Robot Platooning Using Potential Field. Politeknik Negeri Malang, Malang.