

# Kendali Formasi *Mobile Robot* berdasarkan Jarak menggunakan Algoritma *Cosinus*

ANGGORO DWI NUR ROHMAN, MUHAMMAD AZIZ MUSLIM, BAMBANG SISWOJO

Program Studi Magister, Teknik Elektro, Universitas Brawijaya, Indonesia  
Email : [anggoro\\_dwi@student.ub.ac.id](mailto:anggoro_dwi@student.ub.ac.id)

*Received* 30 November 201x | *Revised* 30 Desember 201x | *Accepted* 30 Januari 201x

## ABSTRAK

*Kendali formasi adalah topik penelitian kendali multi-robot, dimana sekelompok robot dapat mencapai formasi tertentu dan mempertahankannya ketika berpindah ke arah yang diinginkan. Salah satu pengembangan kendali formasi adalah kendali formasi berdasarkan jarak dimana setiap individu robot menggunakan informasi jarak antara sesamanya untuk mencapai tujuan formasi. Banyak pengembangan yang dilakukan pada kendali formasi berdasarkan jarak menggunakan model yang sederhana dan membutuhkan pengembangan lebih lanjut untuk penerapan kendali ke model yang lebih nyata. Ketika penerapan kendali formasi berdasarkan jarak, terdapat permasalahan kondisi awal yaitu robot tidak dapat menentukan koordinat tetangganya. Penelitian ini akan mengembangkan algoritma cosinus sebagai solusi untuk kondisi awal kendali formasi berdasarkan jarak. Algoritma cosinus terinspirasi dari rumus segitiga sederhana dan mengharuskan robot melakukan dua langkah saja untuk dapat menemukan koordinat tetangganya. Hasil percobaan simulasi, kendali formasi berdasarkan jarak menggunakan tiga model robot holonomic dan penerapan algoritma cosinus membutuhkan waktu rata-rata 6.5 detik untuk menemukan koordinat tetangganya.*

**Kata kunci:** *Kendali Formasi, Multi-Robot, Algoritma Cosinus, Mobile Robot.*

## ABSTRACT

*Formation control is a research topic of multi-robot control, where a group of robots can reach a certain formation and defend it when moving in the desired direction. One of the developments is distance-based where formation goals achieved using the distance between each other only. Many developments are using a simple model and need further development into a realistic model. When applying distance-based, there is a problem in the initial condition, namely that the robot cannot find the coordinates of its neighbors when using only distance. In this work, the cosine algorithm was developed as a solution to the initial conditions which are inspired by a simple triangle formula and need only two steps to find the coordinates. From simulation experiment results, distance-based formation control using three holonomic robot models and the application of the cosine algorithm takes an average of 6.5 seconds to find the coordinates of its neighbors.*

**Keywords:** *Formation Control, Multi-robot, Cosine Algorithm, Mobile Robot.*

## 1. PENDAHULUAN

Kendali formasi adalah topik penelitian kendali multi-robot untuk memecahkan permasalahan koordinasi pergerakan. Kendali formasi bertujuan untuk mengendalikan sekelompok robot dalam mencapai formasi tertentu dan dapat mempertahankan formasi tersebut ketika bermanuver menuju arah yang diinginkan. Penjabaran oleh Guanghua (**Guanghua, dkk, 2013**), pengembangan kendali formasi dilakukan dari beberapa algoritma strategi. Seperti yang dikembangkan oleh Wang (**Wang, dkk, 2014**) menggunakan strategi *leader-follower*, menggunakan *Fuzzy-Logic* sebagai tingkah laku robot (**Ferik, dkk, 2016**) dan menggunakan *struktur virtual* dimana sekelompok robot memiliki titik referensi sebagai satu robot oleh Xuenren (**Li, dkk, 2015**).

Penjabaran oleh Kwang-Kyo (**Oh, dkk, 2015**) bahwa dari berbagai pengembangan kendali formasi dapat ambil garis besar menjadi tiga bagian, yaitu berdasarkan posisi, perpindahan, dan jarak. Ketiga bagian tersebut tertuju pada jawaban dari pertanyaan, "variabel apa yang digunakan sebagai sensor" dan "variabel apa yang aktif dikendalikan oleh sistem multi-robot untuk mencapai formasi yang diinginkan". Kendali formasi berdasarkan posisi adalah metode kendali formasi dimana robot diharuskan memiliki kemampuan untuk mengetahui koordinatnya sendiri berdasarkan koordinat global. Karena itu kendali berdasarkan posisi membutuhkan sensor posisi seperti GPS. Kendali formasi berdasarkan perpindahan adalah kendali formasi dimana setiap robot tidak mengetahui koordinatnya berdasarkan koordinat global dikarenakan variabel yang digunakan sebagai sensor adalah kecepatan terhadap tetangganya. Metode berdasarkan posisi, robot membutuhkan kemampuan penyesuaian orientasi koordinat global sehingga setiap robot dibutuhkan sensor kompas untuk menyarahkannya. Formasi berdasarkan jarak adalah kendali formasi dimana variabel yang dikendalikan adalah variabel jarak antar robot yang terhubung sehingga koordinat yang digunakan tidak mengacu pada koordinat global. Penerapan formasi berdasarkan jarak menggunakan sensor yang lebih sedikit dibanding dengan posisi dan perpindahan. Namun pembahasan model yang lebih nyata untuk diterapkan kendali formasi berdasarkan jarak masih sedikit. Pengembangan formasi berdasarkan jarak telah dikembangkan menggunakan teori *graph* dengan model *single* dan *double integrator* sederhana oleh Kwang-Kyo (**Oh & Ahn, 2014**), menggunakan informasi jarak untuk mengendalikan model sederhana *double integrator* oleh Xiaoyu (**Cai & de Queiroz, 2014**), menggunakan konsensus antara robot untuk mencapai bentuk formasi dengan model *single integrator* sederhana oleh Deghat (**Deghat, dkk, 2016**), mengendalikan bentuk formasi menggunakan *adaptive control* untuk mengestimasi kecepatan tetangga dari model robot yang sederhana oleh Sung-Mo (**Kang, dkk, 2014**), menggabungkan kendali formasi berdasarkan jarak dan perpindahan untuk mengendalikan model robot yang sederhana oleh Myoung-Chul (**Park & Ahn, 2015**), dan menggunakan kendali *Proportional-Integral* (PI) untuk mengendalikan jarak setiap model robot yang sederhana oleh Rozenheck (**Rozenheck, dkk, 2015**).

Kendali PI pada penelitian oleh Rozenheck (**Rozenheck, dkk, 2015**) tidak dapat langsung diterapkan menggunakan sensor jarak karena kendali tersebut mengambil informasi jarak menggunakan selisih dari koordinat kartesian global setiap robot. Sedangkan dalam praktiknya robot hanya bisa mengukur jarak dan tidak mengetahui koordinat dari robot tetangganya. Sudah umum penelitian dibidang lokalisasi menggunakan sensor jara seperti yang dibahas oleh Kexing (**Guo, dkk, 2020**) yang menggantikan sensor berbasis *vision* untuk mendapatkan koordinat dari beberapa robot, penelitian oleh Qiang (**Qiang, dkk, 2017**) membutuhkan beberapa sensor jarak yang terpasang statis digunakan untuk mengetahui koordinat sekelompok robot dan mengendalikannya secara terpusat, dan pengembangan oleh Qiang (**Qiang, dkk, 2018**) dengan memasang dua sensor jarak di salah satu dari sekelompok robot,

lalu mendistribusikan koordinat ke robot tetangganya. Lokalisasi menggunakan dua sensor jarak tersebut memanfaatkan rumus segitiga untuk mendapatkan koordinat robot tetangganya. Penelitian ini akan mengadopsi rumus segitiga dan menggantikan salah satu dari dua sensor dengan algoritma *cosinus* sehingga tidak mengharuskan salah satu robot memiliki dua sensor. Penelitian oleh Rozenheck (**Rozenheck, dkk, 2015**) mengembangkan kendali formasi menggunakan model *holonomic* sederhana dimana robot bergerak ke suatu arah tidak bergantung dari kondisi awal arah robot dan juga model sederhana tidak mempertimbangkan parameter fisik pada modelnya. Percobaan akan menggunakan tiga model robot *holonomic* dengan harapan menjadi langkah awal untuk mengembangkan kendali formasi berdasarkan jarak menggunakan model robot yang lebih nyata.

## 2. METODE

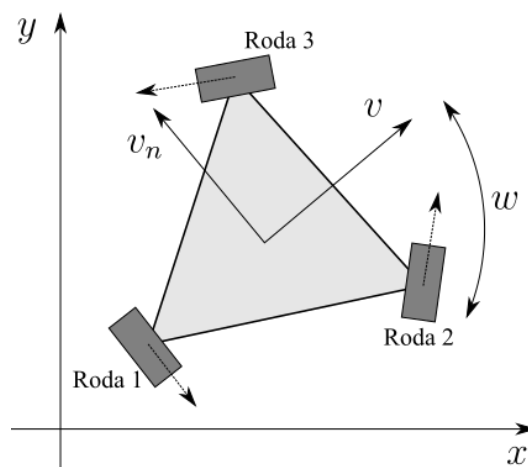
### 2.1 Model Robot

Robot menggunakan tiga roda *omniwheel* yang dipasang sesuai dengan geometri rangka robot di Gambar 1. Dengan mengatur kecepatan ketiga roda tersebut robot dapat bergerak ke segala arah tanpa dipengaruhi oleh keadaan awal arah robot sehingga robot bersifat *holonomic*. Berikut adalah model dari robot *omniwheel holonomic* dalam bentuk *state-space*.

$$\dot{x}(t) = A_r x(t) + B_r u(t) + K_r \text{sgn}(x(t)) \quad (1)$$

$$y(t) = Cx(t) \quad (2)$$

Dimana vektor  $u(t) = [u_1(t) \ u_2(t) \ u_3(t)]^T$  adalah vektor tegangan motor untuk memutar roda  $\vartheta = \{1,2,3\}$  yang memiliki batasan tegangan  $-6 \leq u_\vartheta \leq 6$  volt. Vektor  $y(t) = x(t) = [v(t) \ v_n(t) \ w(t)]^T$  adalah kecepatan robot yang akan diperoleh dari sensor percepatan, dimana  $v$  adalah kecepatan robot bergerak maju/mundur atau koordinat  $x$  berdasarkan kerangka robot,  $v_n$  adalah kecepatan robot untuk menyamping kiri/kanan atau koordinat sumbu  $y$  berdasarkan kerangka robot, dan  $w(t)$  adalah kecepatan rotasi dari kerangka robot. Matriks  $A_r \in R^{3 \times 3}$  dan  $B_r \in R^{3 \times 3}$  adalah parameter fisik robot seperti dimensi robot, dimensi roda, masa jenis dan lain lain. Matriks  $K_r \in R^{3 \times 3}$  adalah parameter *friction* dari robot yang diestimasi dari hasil identifikasi sistem dengan cara percobaan. Penelitian ini menggunakan parameter fisik dan hasil estimasi *friction* yang dibahas secara jelas didalam paper penelitian oleh Mariane (**Correia, dkk, 2012**).



**Gambar 1.** Geometri Rangka Robot

## 2.2 Kendali Robot Holonomic

Kendali robot holonomic akan dibagi menjadi dua mode. Perbedaan kedua mode tersebut adalah *setpoint* kendalinya, dimana mode satu akan memiliki *setpoint* kecepatan robot sedangkan mode dua memiliki *setpoint* koordinat kerangka robot. Penjelasan lebih lengkap kegunaan dari kedua mode tersebut akan dijelaskan di pembahasan metode algoritma *cosinus* (subbab 2.4)

**Mode Satu.** Kendali robot mode satu bertujuan untuk robot mencapai kecepatan yang diinginkan. Untuk mencapai tujuan tersebut akan menggunakan metode *state-feedback* sebagai kendalinya.

$$u_{c1}(t) = -K_1^c x(t) + N_1^c r_1^c \quad (3)$$

$$N_1^c = -[C(A_r - B_r K_1^c)^{-1} B_r]^{-1} \quad (4)$$

Dimana vektor  $r_1^c = [v^* \ v_n^* \ w^*]$  adalah *setpoint* kendali mode satu,  $K_1^c \in R^{3 \times 3}$  adalah konstanta yang diperoleh dari optimasi persamaan *Riccati* terhadap maktriks  $A_r$  dan  $B_r$  di Persamaan (1), dan  $N_1^c \in R^{3 \times 3}$  adalah konstanta yang diperoleh dari *inverse state-space* ketika keadaan *steady state* sehingga menghasilkan Persamaan (4). Dengan menggabungkan Persamaan (1) dan Persamaan (3) akan mendapatkan persamaan *state-space* robot yang baru.

$$\dot{x}(t) = A_c x(t) + B_c r_1^c + K_r \text{sgn}(x(t)) \quad (5)$$

$$A_c = (A_r - B_r K_1^c) \quad (6)$$

$$B_c = B_r N_1^c \quad (7)$$

**Mode Dua.** Kendali robot mode dua bertujuan untuk robot mencapai titik koordinat tertentu berdasarkan koordinat kerangkanya. Untuk mencapai tujuan tersebut akan memodifikasi Persamaan (1) dengan menambah koordinat *frame* robot pada *state*-nya. *State* robot menjadi  $x_{c2}(t) = [x_r(t) \ y_r(t) \ \theta_r(t) \ v(t) \ v_n(t) \ w(t)]^T$ . Berikut adalah *state-space* robot mode dua.

$$\dot{x}_{c2}(t) = A_{c2} x_{c2}(t) + B_{c2} u_{c2}(t) + K_{c2}(x_{c2}(t)) \quad (8)$$

$$y_{c2}(t) = C_{c2} x_{c2}(t) \quad (9)$$

$$A_{c2} = \begin{bmatrix} 0 & I \\ 0 & A_r \end{bmatrix} \in R^{6 \times 6} \quad (10)$$

$$B_{c2} = \begin{bmatrix} 0 & B_r \end{bmatrix} \in R^{6 \times 3} \quad (11)$$

$$K_{c2}(x) = \begin{bmatrix} 0 & 0 \\ 0 & K_r \text{sgn}(x) \end{bmatrix} \in R^{6 \times 3} \quad (12)$$

Untuk menentukan  $u_{c2}(t)$  sebagai persamaan kendali di Persamaan (8), akan dilakukan metode yang sama seperti mode satu.

$$u_{c2}(t) = -K_2^c x_{c2}(t) + N_2^c r_2^c \quad (13)$$

Dimana  $r_2^c = [x_r^* \ y_r^* \ \theta_r^* \ v^* \ v_n^* \ w^*]^T$  adalah *setpoint* kendali mode dua,  $K_2^c \in R^{3 \times 6}$  adalah konstanta yang diperoleh dari optimasi persamaan *Riccati* terhadap maktriks  $A_{c2}$  dan  $B_{c2}$

menggunakan Persamaan (8) dan  $N_2^c \in R^{3 \times 6}$  adalah kostanta yang diperoleh dari *inverse state-space* ketika keadaan *steady state*.

### 2.3 Kendali Formasi Berdasarkan Jarak

Kendali formasi berdasarkan jarak memanfaatkan variabel jarak untuk mengendalikan sekelompok robot mencapai formasi jarak yang diinginkan. Penelitian oleh Oh (**Oh & Ahn, 2014**) menggunakan teori *graph* untuk menjelaskan kendali formasi berdasarkan jarak. Pada teori *graph* terdapat notasi himpunan simpul ( $\mathcal{V}$ ), dan himpunan sisi ( $\mathcal{E}$ ). Himpunan simpul, ( $\mathcal{V}$ ) juga dapat disebut sebagai *node* atau *points* atau dalam kontes penelitian ini simpul adalah robot. Himpunan sisi ( $\mathcal{E}$ ) mempresentasikan hubungan antara simpul dan juga dapat disebut sebagai *edge* atau *link* atau dalam kontes penelitian ini adalah variabel jarak antara dua robot yang bertetangga. Didalam teori *graph* hubungan antara simpul memiliki arah hubungan atau dapat diilustrasikan sebagai komunikasi, *undirect graph* adalah kerangka *graph* yang memiliki komunikasi dua arah, dalam konteks penelitian ini setiap robot saling mengukur jarak antara tetangganya. Penelitian ini akan menggunakan definisi *undirect graph*  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  dimana terdiri dari himpunan simpul  $\mathcal{V}$  dan himpunan sisi  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ . Didefinisi  $n \triangleq |\mathcal{V}|$  sebagai jumlah dari robot dan  $m \triangleq |\mathcal{E}|$  sebagai jumlah dari variabel jarak antar robot yang bertetangga. Didefinisi  $p = [x_1^T \dots x_n^T]^T \in R^{3n}$  adalah vektor posisi dari semua robot dimana  $x_i \in R^3$  adalah koordinat robot  $i$  dan kadang menggunakan notasi  $j$  sebagai notasi bahwa robot  $i$  bertetangga dengan  $j$ , maka himpunan sisi  $\mathcal{E}$  berlaku  $x_i \neq x_j$  untuk semua  $i \neq j$ . Dinotasikan vektor posisi relatif antara robot sebagai  $e_k \triangleq x_j - x_i$  dan semua vektor sisi  $e = [e_1^T \dots e_m^T] \in R^{3m}$ . Kendali formasi yang dikembangkan oleh Oh (**Oh & Ahn, 2014**) menggunakan persamaan potensial sebagai berikut

$$\Phi(e) = \frac{1}{2} \sum_{i \in \mathcal{V}} \|v_i\|^2 + \frac{1}{2} \sum_{k=1}^m \left( \|e_k\|^2 - d_k^2 \right)^2 \quad (14)$$

Dimana  $v_i$  adalah kecepatan robot  $i$  dan  $d_k$  adalah jarak yang diinginkan. Mengadopsi metode dari penelitian oleh Oh (**Oh & Ahn, 2014**) menggunakan model *double integrator* sederhana dan menggantikan variabel masukannya dengan *gradient negative* akan menghasilkan persamaan *state-space* yang akan dijalankan secara simulasi untuk mengamati pergerakan semua robot.

$$\dot{p} = v(t) \quad (15)$$

$$= A_f(p(t))p(t) + B_f \frac{\partial \Phi(e)}{\partial v} \quad (16)$$

$$= A_f(p(t))p(t) + B_f v(t) \quad (17)$$

$$\dot{v} = u(t) \quad (17)$$

$$= -C \left( \frac{\partial \Phi(e)}{\partial v} + \frac{\partial \Phi(e)}{\partial p} \right) \quad (18)$$

$$= -k_{p1} v(t) + R^T(p(t)) k_{p2} (R(p(t))p(t) - d)$$

Model sederhana di Persamaan (15) memiliki masukan kecepatan, maka dapat diganti dengan dengan Persamaan (5) menjadi Persamaan (16) dimana variabel masukan  $r_1^c$  adalah kecepatan. Lalu Persamaan (17) menggunakan *gradient negative* dari Persamaan (14) akan menghasilkan Persamaan (18) sebagai kendali formasinya. Merujuk dari penelitian oleh Rozenheck (**Rozenheck, dkk, 2015**),  $R(p)$  adalah fungsi *rigidity matrix*

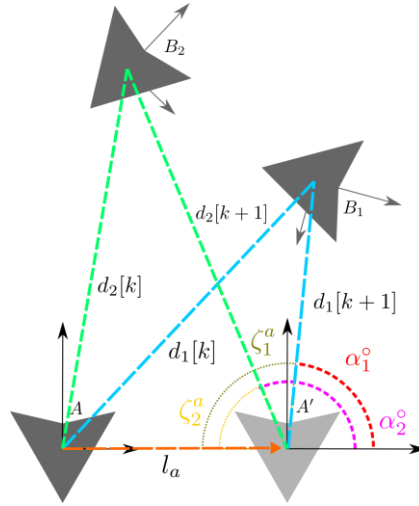
$$R(p) \triangleq \text{diag}(e_i^T) (E^T \otimes I_2) \in R^{m \times 3n} \quad (19)$$

Menggunakan ilustrasi maktriks  $A_i \in R^{p \times q}$  dapat dijelaskan fungsi  $diag(A_i)$ , yaitu operasi, *block diagonal matrix* yang dapat didefinisikan sebagai  $diag(A_i) \triangleq blkdiag\{A_1, \dots, A_n\} \in R^{np \times nq}$ ,  $I_2$  adalah maktriks identitas  $2 \times 2$  dan  $E \in R^{n \times m}$  adalah maktriks *incidence*  $\{0, \pm 1\}$  yang baris dan kolom maktriksnya mengindikasikan simpul dan sisinya.

#### 2.4 Algoritma *Cosinus*

Telah didefinisi di pembahasan pendahuluan (bab 1) bahwa menggunakan informasi jarak saja robot tidak mengetahui koordinat tetangganya, sedangkan di Persamaan (16) dan (18), sebagai kendali formasi, membutuhkan *state* awal koordinat. Penelitian ini mengusulkan algoritma *cosinus* untuk menemukan *state* awal tersebut. Algoritma *cosinus* membutuhkan robot untuk mengerjakan dua Langkah yang akan dijelaskan sebagai berikut.

Dinotasikan  $\mathcal{N}_A$  adalah himpunan tetangga dari robot A. Dinotasikan  $B_i \in \mathcal{N}_A$  adalah robot  $i$  sebagai tetangga dari robot A. Dinotasikan  $d_i[k]$  adalah informasi jarak antara robot A ke robot  $B_i$ .



**Gambar 2. Strategi Algoritma *Cosinus* Pencarian Koordinat Tetangga Langkah Pertama**

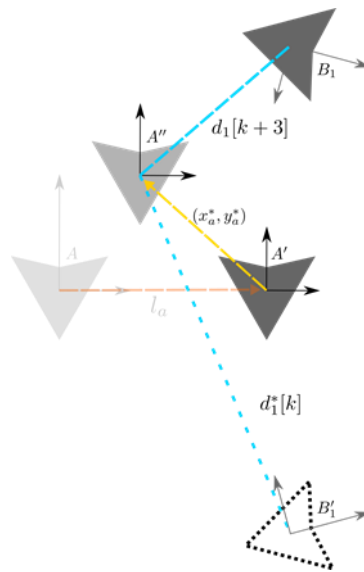
**Langkah pertama.** Gambar 2 adalah Ilustrasi langkah pertama algoritma *cosinus* dimana robot A diharuskan berpindah sepanjang  $l_a$  atau ke koordinat  $A' = (0, l_a)$  menggunakan kendali mode dua di Persamaan (8) akan tetapi robot harus menyimpan jarak  $d_i[k]$  terlebih dahulu. Setelah berpindah robot A mendapatkan jarak  $d_i[k + 1]$  digunakan untuk menentukan sudut  $\alpha_i^o$  menggunakan rumus segitiga *cosinus*. Berikut adalah Persamaan  $\alpha_i^o$ .

$$\zeta_i^a = \cos^{-1} \left( \frac{l_a^2 + d_i[k + 1]^2 - d_i[k]^2}{2d_i[k + 1]l_a} \right) \quad (20)$$

$$\alpha_i^o = 180^\circ \pm \zeta_i^a \quad (21)$$

Variable  $\alpha_i^o$  dan  $d_i[k + 1]$  adalah nilai dari koordinat polar dari setiap robot tetangga A. Diubah menjadi koordinat kartesian untuk dapat dimasukkan dalam *state* kendali formasi.

$$x_{B_i}^A = \begin{bmatrix} x_{B_i} = d_i[k + 1] \cos \alpha_i^o \\ y_{B_i} = d_i[k + 1] \sin \alpha_i^o \end{bmatrix} \quad (22)$$



**Gambar 3. Strategi Algoritma *Cosinus* Pencarian Koordinat Tetangga Langkah Kedua**

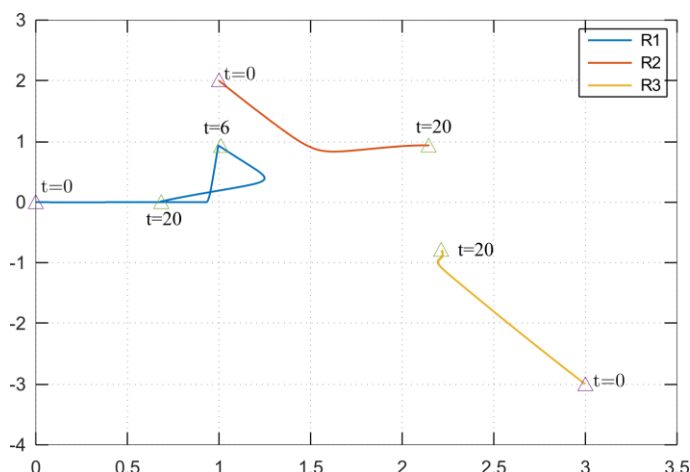
**Langkah kedua.** Koordinat di Persamaan (22) akan menghasilkan bias dikarenakan Persamaan (21) tidak mengetahui letak kuadran sudutnya. Menggunakan ilustrasi di Gambar 3, langkah pertama menghasilkan dua kemungkinan koordinat robot  $B_1$  dan  $B'_1$ . Apabila di perhatikan di Gambar 2, Sudut  $\zeta_i^a$  adalah sudut segitiga  $\angle AA'B_1$  atau  $\angle AA'B_2$  sehingga dimungkinkan koordinat yang dihasilkan Persamaan (22) bisa berada pada kuadran 1 atau kuadran 4. Oleh karena itu di Persamaan (21) terdapat operasi  $\pm$  dimana operasi tersebut akan dilakukan berdasarkan letak kuadran  $B_i$ .

$$\alpha_i^\circ = \begin{cases} 180^\circ - \zeta_i^a, & \text{Robot berada pada kuadran 1 dan 2} \\ 180^\circ + \zeta_i^a, & \text{Robot berada pada kuadran 3 dan 4} \end{cases} \quad (23)$$

Langkah kedua ini bertujuan untuk menentukan kejadian di Persamaan (23) dimana robot harus berpindah ke koordinat  $A'' = (x_a^*, y_a^*)$  (Gambar 3). Sebelum robot berpindah, kondisi robot telah mendapatkan koordinat dari langkah pertama. Koordinat tersebut akan diubah menjadi jarak dan akan dibandingkan jarak tersebut dengan informasi jarak dari sensor setelah berpindah ke  $A''$ . Apabila terdapat perbedaan maka kejadian di Persamaan (22) diubah ke kejadian selanjutnya dan mengkoreksi koordinat sebelumnya.

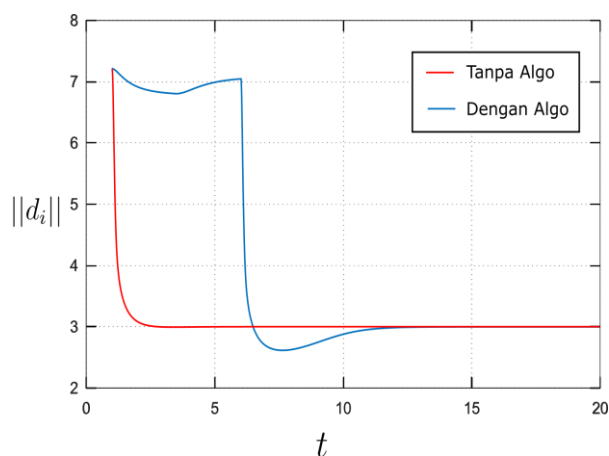
### 3. HASIL DAN PEMBAHASAN

Kendali formasi berdasarkan jarak akan dijalankan secara simulasi menggunakan MATLAB/GNU Octave dan algoritma *cosinus* akan dijalankan pertama kali untuk mendapatkan *state* yang akan digunakan di Persamaan (16) dan (18). Simulasi akan menggunakan 3 robot dengan himpunan simpul  $\mathcal{V} = \{R_1, R_2, R_3\}$  dan himpunan sisi  $\mathcal{E} = \{(R_1, R_2), (R_3, R_2), (R_3, R_1)\}$  sehingga variable jarak  $d_1 = \|x_1 - x_2\|$ ,  $d_2 = \|x_3 - x_2\|$  dan  $d_3 = \|x_3 - x_1\|$ . Gambar 4 adalah pergerakan robot terhadap koordinat global robot  $R_1$  dengan kendali formasi algoritma *cosinus*. dapat diperhatikan robot  $R_1$  menjalankan algoritma *cosinus* langkah pertama dimana robot berpindah sepanjang  $l_a = 1$  sehingga *setpoint* kendali Persamaan (13) adalah  $r_2^c = [1 \ 0 \ 0 \ 0 \ 0 \ 0]^T$ . Setelah *setpoint* tercapai, robot  $R_1$  mendapatkan jarak yang dibutuhkan



**Gambar 4. Grafik Pergerakan Robot Menggunakan Algoritma *Cosinus* dan Kendali Formasi**

untuk menyelesaikan Persamaan (21) dan (22) lalu dilanjutkan ke langkah dua dengan berpindah ke koordinat (1,1) sehingga *setpoint* kendali Persamaan (13) adalah  $r_2^c = [1 \ 1 \ 0 \ 0 \ 0 \ 0]^T$ . Setelah *setpoint* tercapai maka dilakukan pengecekan kejadian Persamaan (23) dengan membandingkan jarak dari sensor dengan jarak dari koordinat yang dihasilkan dari Persamaan (22). Robot  $R_3$  berada di kuadran 4 maka kejadian yang digunakan adalah  $\alpha_i^o = 180^\circ + \zeta_i^a$  dan robot  $R_2$  berada di kuadran 1 maka kejadian yang digunakan adalah  $\alpha_i^o = 180^\circ - \zeta_i^a$ . Setelah koordinat ditemukan, maka kendali robot mulai berpindah menggunakan kendali formasi Persamaan (16) dan (18) dengan kondisi koordinat awal *state* berada di  $t = 6$  dan akhir dari formasi di  $t = 20$  pada Gambar 4.



**Gambar 5. Grafik *Response* Jarak Terhadap Waktu Pada Robot Dengan Kendali Formasi.**

Seperti yang telah dijelaskan di pembahasan algoritma *cosinus* (subbab 2.4), bahwa kendali formasi di Persamaan (16) dan (18) mengharuskan *state* dalam bentuk koordinat dimana ketika kondisi awal robot tidak mengetahuinya, maka peran utama dari algoritma *cosinus* untuk mendapatkan *state* tersebut. Gambar 5 adalah grafik perbandingan kendali formasi Persamaan (16) dan (18) dengan algoritma *cosinus* dan tanpa algoritma *cosinus* dengan asumsi ketika dilakukan percobaan tanpa algoritma, kondisi awal *state* telah diketahui. Hasil jarak  $\|d_i\|$  pada kendali formasi tanpa algoritma *cosinus* menunjukkan bahwa setiap robot mencapai jarak yang sama dalam waktu kurang lebih 4 detik. Sedangkan pada kendali formasi dengan algoritma *cosinus* terdapat tambahan waktu 6 detik. Tambahan waktu tersebut digunakan untuk menjalankan algoritmanya.



**Tabel 1. Settling Time Dengan Konstanta Kp Yang Berbeda**

Koordinat			Settling Time Tanpa Algoritma <i>cosinus</i> (detik)			Settling Time Dengan Algoritma <i>cosinus</i> (detik)			Selisih (detik)		
R1	R2	R3	A	B	C	A	B	C	A	B	C
(0, 0)	(1, 2)	(-2, 3)	6	5	3	14	11	9	8	6	6
(0, 0)	(-2, -4)	(3, -2)	4	2	2	9	8	7	5	6	5
(0, 0)	(1, 2)	(3, -3)	7	4	2	16	13	11	9	9	9
(0, 0)	(-2, -3)	(3, 2)	10	7	4	16	11	9	6	4	5
<b>Konstanta:</b> (A) $K_{p1} = 50$ ; $K_{p2} = 3$ ; (B) $K_{p1} = 80$ ; $K_{p2} = 7$ ; (C) $K_{p1} = 100$ ; $K_{p2} = 15$											

Percobaan di Gambar 5 adalah grafik *settling time* saat koordinat robot seperti di Gambar 4 yaitu R2 berada di kuadran 1 dan R3 berada di kuadran 4. Hasil percobaan Tabel 1 bertujuan untuk mengetahui kecepatan algoritma *cosinus* mencari koordinat tetangga dengan kombinasi kemungkinan dua robot dengan kejadian di Persamaan (23). Serta diberikan nilai  $K_{p1}$  dan  $K_{p2}$  di Persamaan (18) yang berbeda untuk mengetahui pengaruh parameter kendali formasi terhadap algoritma *cosinus*. Setelah dilakukan percobaan, selisih waktu ketika algoritma *cosinus* berjalan dengan perbedaan parameter menghasilkan waktu yang tidak jauh beda antara 0-2 detik. Maka, algoritma *cosinus* mendapatkan koordinat tetangga yang digunakan untuk nilai kondisi awal kendali hampir tidak berpengaruh terhadap kendali formasi Persamaan (18) dengan rata-rata waktu 6.5 detik, minimal waktu 4 detik dan maksimal 9 detik.

#### 4. KESIMPULAN

Penelitian ini mengusulkan penggunaan algoritma *cosinus* untuk kendali formasi berdasarkan jarak menggunakan tiga model robot *holonomic*. Dari hasil percobaan dicapai rata-rata waktu sebesar 6.5 detik ketika algoritma *cosinus* menemukan keadaan awal kendali formasi berdasarkan jarak menggunakan model robot *holonomic*.

#### DAFTAR RUJUKAN

- Cai, X., & de Queiroz, M. (2014). Multi-agent formation maneuvering and target interception with double-integrator model. *2014 American Control Conference*, (pp. 287–292).
- Correia, M. D., Gustavo, A., & Conceição, S. (2012). Modeling of a Three Wheeled Omnidirectional Robot Including Friction Models. *IFAC Proceedings Volumes, 45(22)*, 7–12.
- Deghat, M., Anderson, B. D. O., & Lin, Z. (2016). Combined Flocking and Distance-Based Shape Control of Multi-Agent Formations. *IEEE Transactions on Automatic Control, 61(7)*, 1824–1837.
- Ferik, S. El, Nasir, M. T., & Baroudi, U. (2016). A Behavioral Adaptive Fuzzy controller of multi robots in a cluster space. *Applied Soft Computing, 44*, 117–127.
- Guanghua, W., Deyi, L., Wenyan, G., & Peng, J. (2013). Study on Formation Control of Multi-Robot Systems. *2013 Third International Conference on Intelligent System Design and*

*Engineering Applications*, (pp. 1335–1339).

- Guo, K., Li, X., & Xie, L. (2020). Ultra-Wideband and Odometry-Based Cooperative Relative Localization With Application to Multi-UAV Formation Control. *IEEE Transactions on Cybernetics*, *50*(6), 2590–2603.
- Kang, S.-M., Park, M.-C., Lee, B.-H., & Ahn, H.-S. (2014). Distance-based formation control with a single moving leader. *2014 American Control Conference*, (pp. 305–310).
- Li, X., Li, B., Zhang, P., Zhang, J., & Zhang, S. (2015). Large-Scale Unmanned Aerial Vehicle Formation Maintenance Algorithm Based on Virtual Structure. *2015 8th International Symposium on Computational Intelligence and Design (ISCID)*, *1*, (pp. 417–421).
- Oh, K.-K., & Ahn, H.-S. (2014). Distance-based undirected formations of single-integrator and double-integrator modeled agents in n-dimensional space. *International Journal of Robust and Nonlinear Control*, *24*(12), 1809–1820.
- Oh, K.-K., Park, M.-C., & Ahn, H.-S. (2015). A survey of multi-agent formation control. *Automatica*, *53*, 424–440.
- Park, M.-C., & Ahn, H.-S. (2015). Distance-based control of formations with orientation control. *2015 54th IEEE Conference on Decision and Control (CDC)*, (pp. 2199–2204).
- Qiang, L., Heng, W., Huican, L., Shuqi, Q., Nanxun, D., & Bing, L. (2018). Formation control of multi robot based on UWB distance measurement. *2018 Chinese Control And Decision Conference (CCDC)*, (pp. 2404–2408).
- Qiang, L., Heng, W., Huican, L., & Ying, Z. (2017). Design and implementation of multi robot research platform based on UWB. *2017 29th Chinese Control And Decision Conference (CCDC)*, (pp. 7246–7251).
- Rozenheck, O., Zhao, S., & Zelazo, D. (2015). A proportional-integral controller for distance-based formation tracking. *2015 European Control Conference (ECC)*, (pp. 1693–1698).
- Wang, X., Yan, Z., & Wang, J. (2014). Model predictive control of multi-robot formation based on the simplified dual neural network. *2014 International Joint Conference on Neural Networks (IJCNN)*, (pp. 3161–3166).