

PEMBAGIAN TUGAS DALAM KOLONI ROBOT OTONOM BERGERAK PADA STUDI KASUS PEMBENTUKAN POLA FORMASI SECARA GRADUAL

W. Jatmiko¹, A. A. Krisnadhi², dan N. W. Pambudi³

^{1,2,3}Fakultas Ilmu Komputer, Universitas Indonesia, Depok, Indonesia
wisnuj@cs.ui.ac.id, adila@cs.ui.ac.id, nwp50@ui.ac.id

Abstrak

Pada penelitian ini dilakukan suatu eksperimen pembentukan formasi pola dengan menggunakan Robot LEGO Mindstorms NXT. Pembentukan pola ini dilakukan secara bertahap dimulai dari pembentukan formasi pola yang lebih sederhana menuju kepada formasi pola yang lebih kompleks. Dalam penelitian ini, diterapkan konsep pembedaan peranan dalam pembentukan formasi pola poligon. Konsep pembedaan peranan ini diterapkan dengan menggunakan persamaan periodik. Selain itu, persamaan pembedaan peranan dibuat dengan fitur terbebas dari kondisi awal dari nilai variabel pembeda. Pembentukan formasi pola pada penelitian ini dilakukan dalam 3 tahapan, yaitu tahap penerapan persamaan, tahap penerapan robot simulasi, dan tahap penerapan perangkat robot. Pembentukan formasi pola yang dilakukan pada penelitian ini ada tiga macam, yaitu pembentukan formasi pola segitiga, pembentukan formasi pola segiempat / jajaran genjang, dan pembentukan formasi pola segilima.

Kata kunci: Pola formasi gradual, robot bergerak terdistribusi, robot lego mindstorm nxt.

1. Pendahuluan

Pada banyak keadaan, penggunaan robot otonom dalam kelompok terbukti dapat memberikan solusi yang baik bagi penyelesaian suatu permasalahan yang sulit [1-5]. Salah satu kunci agar penggunaan robot otonom dalam kelompok dapat memberikan hasil yang baik adalah dengan adanya faktor saling berbagi antarrobot yang terlibat di dalam kelompok tersebut. Agar dapat mendukung proses *sharing* informasi ini, komunikasi antarrobot menjadi permasalahan yang harus ditangani dengan baik.

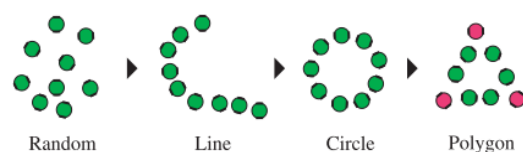
Penelitian kali ini bertujuan untuk melakukan simulasi komunikasi antarrobot otonom dalam melaksanakan suatu tugas tertentu. Lebih jauh lagi, dalam simulasi yang dilakukan akan diperlihatkan kemungkinan bahwa robot dapat diberikan fungsi dan peranan yang berbeda dalam kelompok sehingga dapat mencapai tujuan yang diinginkan secara lebih optimal. Dengan adanya pembedaan fungsi dan peranan ini kemudian robot dapat menentukan posisinya di dalam kelompok. Selanjutnya, jika diperlukan, akan dapat dibedakan robot yang akan berfungsi sebagai pemimpin dalam kelompok tersebut, ataupun robot-robot lainnya yang akan berfungsi untuk melakukan suatu fungsi spesifik secara mandiri [6].

Untuk keperluan inilah, penelitian ini menggunakan studi kasus pembentukan beragam pola formasi secara gradual oleh beberapa robot otonom yang tergabung dalam suatu kelompok. Gradual yang dimaksud di sini adalah bahwa untuk dapat mencapai suatu formasi tertentu harus dilakukan melalui bentuk

lain yang sederhana. Adapun pada contoh kasus ini, proses pembagian fungsi dan peran pada masing-masing robot diperlihatkan pada pembedaan robot yang akan menjadi sudut dalam pola formasi, dibandingkan dengan robot lain yang akan menjadi garis dalam pola formasi tersebut.

Selanjutnya, karena tujuan dari penelitian ini untuk mensimulasikan masalah komunikasi dan koordinasi antarrobot dalam kelompok, hanya akan dipilih beberapa pola formasi yang cukup dapat merepresentasikan masalah tersebut. Pola formasi yang akan dibentuk antara lain pola poligon segitiga, segiempat, dan segilima. Pola formasi dibentuk secara gradual. Sebelum mencapai masing-masing pola poligon yang dimaksud, robot akan membentuk pola lingkaran. Akan tetapi karena robot memulai pergerakannya tanpa suatu pola formasi tertentu, untuk dapat mencapai pola formasi lingkaran tersebut, robot terlebih dahulu membentuk pola formasi garis lurus dari posisi acak tersebut [7].

Gambar 1 berikut ini merupakan contoh tahapan pembentukan pola formasi poligon melalui pola garis lurus dan lingkaran.



Gambar 1. Berbagai tahap yang dilalui robot pada saat membentuk pola formasi secara gradual

Pada penelitian ini, jenis koloni robot otonom yang digunakan bersifat homogen, artinya semua unit robot memiliki spesifikasi yang sama. Hal ini bertujuan agar proses pembagian fungsi dan peran pada setiap robot tidak bergantung pada spesifikasi khusus yang dimilikinya. Dengan demikian, metode yang digunakan pada penelitian ini diharapkan akan dapat diterapkan pada berbagai spesifikasi robot yang berbeda, serta pada berbagai permasalahan yang berbeda pula, selama membutuhkan adanya proses komunikasi dan koordinasi dengan menggunakan koloni robot otonom [8].

2. Metodologi

Salah satu hal yang kemudian harus dipikirkan dalam rangka mengembangkan simulasi pada penelitian ini adalah adanya keterbatasan infrastruktur yang menyebabkan robot juga memiliki keterbatasan dalam mendapatkan informasi di sekitarnya. Walaupun mungkin untuk digunakan robot yang memiliki kemampuan *sensing* dengan jarak yang luas, tentu saja hal ini akan membutuhkan biaya yang besar [9]. Oleh karena itu, ide pembentukan formasi secara bertahap ini kemudian dimaksudkan untuk menangani keterbatasan yang dimiliki setiap robot dalam mendapatkan informasi dari lingkungannya, selain agar masalah yang akan diselesaikan menjadi lebih sederhana.

Kemudian, untuk menyelesaikan permasalahan pembentukan pola formasi ini akan dimanfaatkan informasi-informasi sederhana yang diperoleh melalui proses interaksi secara lokal antara masing-masing robot, sehingga kebutuhan akan informasi lingkungan yang bersifat global dapat dihindari. Dalam hal ini, kita akan menggunakan model persamaan periodik untuk menentukan pembagian fungsi dan peran pada masing-masing robot.

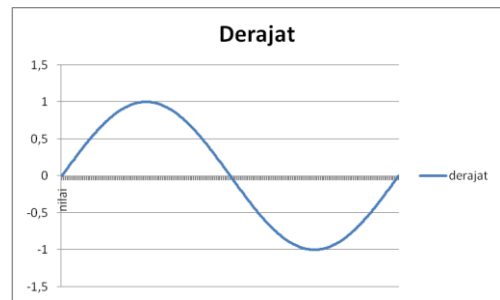
2.1. Fungsi Periodik

Fungsi periodik adalah suatu fungsi matematika yang nilainya akan berulang pada interval atau periode tertentu. Suatu fungsi bisa disebut sebagai fungsi periodik bila fungsi tersebut memenuhi persamaan 1.

$$f(x + P) = f(x) \tag{1}$$

Di mana P adalah sebuah nilai bukan nol yang menandakan periode atau interval dari fungsi tersebut. Contoh fungsi periodik adalah fungsi trigonometris yang nilainya berulang pada interval 2π . Gambar 2 menunjukkan sebuah fungsi

trigonometris, yaitu fungsi $f(x) = \sin x$ [10].



Gambar 2. Fungsi periodik sinus

2.2. Fungsi Flow on the Line

Asumsikan terdapat sebuah persamaan diferensial seperti persamaan 2.

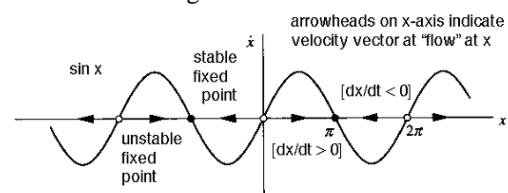
$$\dot{x} = f(x) \tag{2}$$

\dot{x} adalah turunan pertama dari fungsi $x(t)$, $x(t)$ adalah fungsi terhadap waktu t dan $f(x)$ adalah fungsi terhadap x . Persamaan ini merupakan persamaan diferensial orde pertama atau satu dimensi.

Sedangkan *Flows on the Line* adalah cara memandang sebuah persamaan diferensial orde pertama sebagai sebuah medan vektor. Sebagai contoh berikut ini adalah sebuah persamaan diferensial orde pertama.

$$\dot{x} = \sin x \tag{3}$$

Jika direpresentasikan dalam bentuk grafik, persamaan 3 dapat dilihat pada Gambar 3. Selanjutnya, untuk melakukan analisis terhadap grafik tersebut, dapat dianalogikan dalam permasalahan sebagai berikut.



Gambar 3. Representasi dinamika dari fungsi sinus x

Bayangkan sebuah cairan yang bergerak sepanjang garis x dengan kecepatan yang berubah-ubah yang mengikuti aturan. Dengan mengikuti aturan tadi maka solusi kualitatif dari persamaan tersebut pun akan dapat diketahui.

Bayangkan sebuah titik maya (*phase point*) yang berada pada sebuah posisi di cairan tersebut. Untuk mengetahui solusi, kemudian bayangkan titik maya tersebut bergerak pada cairan. Pergerakan titik maya inilah yang kemudian merepresentasikan solusi dari persamaan diferensial tersebut. Hasil analisis yang diperoleh adalah sebagai berikut.

1. Pada saat $\dot{x} > 0$, aliran akan bergerak ke kanan.
2. Pada saat $\dot{x} < 0$, aliran akan bergerak ke kiri.
3. Pada saat $\dot{x} = 0$, aliran akan berhenti.

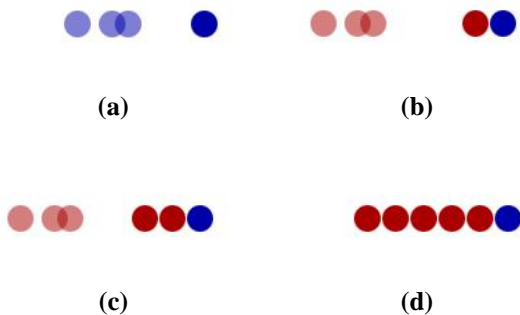
Hal ini berarti perubahan nilai x terhadap waktu sama dengan nol atau nilai x akan konstan. Titik di mana $\dot{x} = 0$ disebut sebagai *fixed points*. Terdapat

dua macam *fixed points*, yaitu *stable fixed point* (disebut juga *sinks* atau *attractors*) di mana aliran cairan maya akan bergerak menuju titik tersebut, dan *unstable fixed point* (disebut juga sebagai *repellers* atau *sources*) di mana aliran cairan maya akan bergerak menjauhi titik tersebut. Dengan melihat jumlah dan posisi dari *stable fixed points* dan *unstable fixed points* pada sebuah analisa grafik dari persamaan diferensial orde pertama, dapat dilihat sifat dari persamaan tersebut saat $t \rightarrow \infty$ [10].

2.3. Pembentukan Pola Garis

Seperti yang disebutkan sebelumnya, bentuk pola formasi yang pertama kali harus dibentuk oleh robot otonom bergerak adalah pola yang paling sederhana, yaitu bentuk garis lurus. Pada saat simulasi dimulai, diasumsikan masing-masing robot akan bergerak memasuki lingkungan simulasi secara berurutan dari arah yang sama. Untuk memudahkan penamaan, selanjutnya robot yang pertama kali memasuki lingkungan, disebut sebagai robot kepala.

Adapun dalam proses memasuki ruang simulasi, robot kepala pertama kali akan berhenti pada suatu jarak yang sudah ditentukan. Selanjutnya, melalui notifikasi yang diberikannya, robot berikutnya akan memasuki ruang simulasi mengikuti robot kepala tersebut. Dengan menggunakan sensor jarak dan sensor rotasi putaran roda, robot ini akan berhenti saat sudah mencapai posisi robot kepala. Selanjutnya robot ini memberikan notifikasi bagi robot selanjutnya, seperti halnya robot kepala juga memberikan notifikasi sebelumnya. Begitu seterusnya hingga semua robot memasuki ruang simulasi dan terbentuk suatu pola berbentuk garis lurus. Pada Gambar 4 terdapat beberapa tahapan dalam membentuk pola formasi garis lurus ini. Pada gambar tersebut, robot kepala digambarkan berwarna biru, sementara robot lainnya digambarkan berwarna merah.



Gambar 4. Semua robot anggota mengikuti posisi robot kepala untuk membentuk formasi garis

2.4. Pembentukan Pola Lingkaran

Setelah berhasil membentuk pola formasi garis lurus, selanjutnya robot diharapkan membentuk pola formasi lingkaran, sebelum akhirnya disimulasikan proses pembentukan pola formasi poligon segitiga,

segiempat serta segilima. Berikut adalah langkah yang harus dilakukan oleh robot untuk mendapatkan pola formasi lingkaran.

1. Robot kepala melakukan perhitungan untuk menentukan lintasan pola formasi lingkaran. Dari hasil tersebut kemudian akan diperoleh nilai kecepatan yang diperlukan pada masing-masing roda kiri dan kanan, yaitu dengan menggunakan persamaan 4..

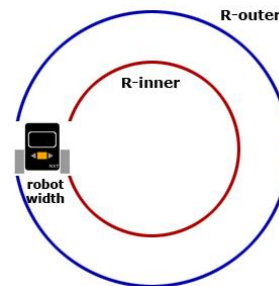
$$V_{\text{left}} = R_{\text{inner}} / t * W * T \tag{4}$$

$$V_{\text{right}} = R_{\text{outer}} / t * W * T$$

Di mana,

- V_{left} : kecepatan roda kiri.
- V_{right} : kecepatan roda kanan.
- R_{inner} : keliling lintasan dalam.
- R_{outer} : keliling lintasan luar.
- t : waktu yang dibutuhkan.
- W : keliling roda.
- T : besar torsi motor.

2. Selanjutnya robot kepala bergerak mengikuti lintasan tersebut hingga melewati satu lingkaran penuh. Saat mencapai tujuan, robot kepala memberikan notifikasi untuk berhenti, dan kemudian berputar menghadapi titik pusat lingkaran sampai kembali ke posisi semula. Gambar 5 menjelaskan perbedaan antara keliling lintasan dalam dengan keliling lintasan luar.



Gambar 5. Perbedaan keliling luar dan dalam formasi lingkaran yang dibentuk

3. Sementara itu, bagi robot-robot lainnya hanya bergerak mengikuti pergerakan robot kepala, serta berhenti saat mendapat notifikasi untuk berhenti. Kemudian masing-masing robot ini berputar menghadapi titik pusat lingkaran.

2.5. Pembentukan Pola Poligon

Selanjutnya untuk dapat membentuk pola formasi poligon dari pola formasi lingkaran akan dilakukan proses pembagian fungsi dan peran bagi masing-masing robot di dalam kelompok. Adapun perbedaan ini akan didasarkan oleh suatu variabel pembeda yang dihitung dengan persamaan periodik yang bersesuaian dengan pola formasi poligon yang

diharapkan. Kemudian dengan menggunakan parameter yang berbeda-beda, pembagian fungsi dan peran pun dapat diterapkan pada setiap robot di dalam kelompok. Hal ini dapat berjalan karena sifat persamaan periodik yang selalu berulang dalam suatu periode tertentu, sehingga menimbulkan grafik yang bergelombang puncak dan lembah, yang dimanfaatkan sebagai konsep dasar pembedaan fungsi dan peran. Pada penelitian ini kemudian digunakan persamaan 5.

$$u_i = \sin(w \theta_i) \quad (5)$$

Di mana, u_i adalah nilai variabel pembeda yang dipetakan pada gelombang sinusoidal, dan w adalah nilai penentu jumlah gelombang yang terbentuk. Selanjutnya juga akan digunakan persamaan diferensial yang diubah agar nilai variabel pembeda tersebut di atas dapat bergerak suatu nilai tertentu dengan $t \rightarrow \infty$. Dengan demikian nilai variabel kemudian dapat dipetakan pada gelombang sinusoidal. Adapun persamaan diferensial yang dimaksud adalah persamaan 6.

$$\Delta\theta_i = \varphi_i - \theta_i \quad (6)$$

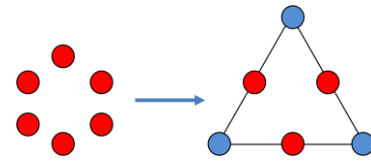
Di mana nilai variabel tersebut masing-masingnya berarti sebagai berikut.

- $\Delta\theta$ Perubahan variabel pembeda.
- φ_i Nilai tujuan yang diinginkan.
- θ_i Variabel pembeda.

Untuk menentukan persamaan untuk masing-masing poligon, sebelumnya dilakukan analisis terhadap bentuk dan karakteristik poligon yang bersangkutan. Pada penelitian ini akan dibahas tiga contoh pola formasi poligon yang dapat dibentuk, yaitu segitiga, segiempat dan segilima. Walaupun demikian, sesungguhnya dengan menerapkan cara penurunan persamaan yang serupa, bentuk poligon yang lain juga dapat dibentuk, tergantung kepada persamaan yang digunakan. Pada paper ini akan diberikan contoh menurunkan persamaan periodik pada ketiga bentuk pola formasi poligon yang akan diterapkan.

2.6. Pola Formasi Segitiga

Untuk dapat menurunkan persamaan periodik dan nilai tujuan variabel pembeda yang akan digunakan pada pembentukan pola formasi segitiga, terlebih dahulu dilakukan analisis terhadap bentuk pola yang dimaksud. Bentuk pola formasi segitiga yang akan dibentuk ini dapat dilihat pada Gambar 6.



Gambar 6. Analisis transformasi pola formasi dari bentuk lingkaran hingga bentuk segitiga

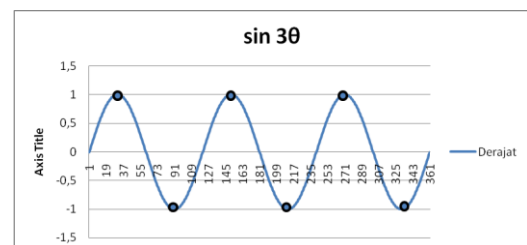
Untuk dapat membentuk pola formasi segitiga seperti pada gambar, diperlukan tiga robot yang bertindak menjadi sudut. Dengan demikian, diperlukan tiga puncak pada persamaan periodik yang akan dirancang, atau dengan kata lain, persamaan periodik yang dibentuk akan mempunyai tiga gelombang untuk mendapatkan tiga robot yang memiliki nilai variabel pembeda yang lebih tinggi dibanding nilai variabel pembeda pada robot lainnya. Jika diberi penomoran mengikuti arah jarum jam, robot yang menjadi sudut adalah robot nomor 1, robot nomor 3 dan robot nomor 5. Untuk itu, nilai variabel pembeda yang dimiliki robot-robot tersebut haruslah lebih tinggi dari robot-robot tetangganya.

Berdasarkan analisis seperti ini, persamaan periodik dan nilai tujuan variabel pembeda yang digunakan pada pembentukan pola formasi segitiga ini mengikuti persamaan 7.

$$u_i = \sin 3\theta_i \quad (7)$$

$$\theta_i = \frac{360 * i}{N} - 30$$

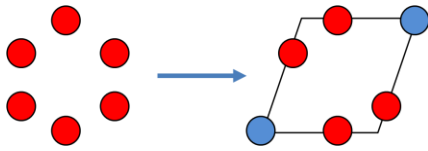
Di mana i adalah penomoran dari masing-masing robot dan N adalah total jumlah robot yang digunakan pada simulasi. Selanjutnya, dari persamaan tersebut akan diperoleh bentuk grafik persamaan periodik seperti Gambar 7.



Gambar 7. Grafik fungsi periodik $\sin 3\theta$ untuk pembentukan pola formasi segitiga

2.7. Pola Formasi Segiempat

Sama seperti pada pembentukan pola formasi segitiga, untuk dapat menentukan persamaan periodik dan nilai tujuan variabel pembeda yang digunakan untuk membentuk pola formasi segiempat, terlebih dahulu dilakukan analisis terhadap bentuk pola formasi yang dimaksud. Bentuk pola formasi segiempat yang akan dibentuk ini dapat dilihat pada Gambar 8.



Gambar 8. Analisis transformasi pola formasi dari bentuk lingkaran hingga bentuk segiempat

Untuk dapat membentuk pola formasi segiempat (atau lebih tepatnya bentuk jajaran genjang) seperti pada gambar, diperlukan dua robot yang bertindak sebagai sudut. Dengan demikian, diperlukan dua puncak pada persamaan periodik yang akan dirancang, atau dengan kata lain, persamaan periodik yang dibentuk akan mempunyai dua gelombang untuk mendapatkan dua robot yang memiliki nilai variabel pembeda yang lebih tinggi dibanding nilai variabel pembeda pada robot lainnya. Jika diberi penomoran mengikuti arah jarum jam, robot yang menjadi sudut adalah robot nomor 2 dan robot nomor 5. Untuk itu, nilai variabel pembeda yang dimiliki robot-robot tersebut haruslah lebih tinggi dari robot-robot tetangganya.

Berdasarkan analisis seperti ini, persamaan periodik dan nilai tujuan variabel pembeda yang digunakan pada pembentukan pola formasi segitiga ini mengikuti persamaan 8.

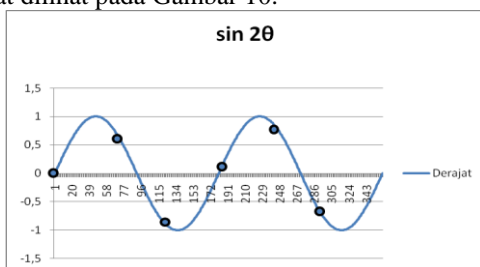
$$u_i = \sin 2\theta_i$$

$$\theta_i = \frac{360 * i}{N} - 60 \tag{8}$$

Di mana i adalah penomoran dari masing-masing robot dan N adalah total jumlah robot yang digunakan pada simulasi. Selanjutnya, dari persamaan tersebut akan diperoleh bentuk grafik persamaan periodik seperti Gambar 9.

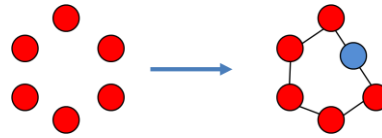
2.8. Pola Formasi Segilima

Sama seperti pada pembentukan pola formasi segitiga dan pola formasi segiempat, untuk dapat menentukan persamaan periodik dan nilai tujuan variabel pembeda yang digunakan untuk membentuk pola formasi segilima, terlebih dahulu dilakukan analisis terhadap bentuk pola formasi yang dimaksud. Bentuk pola formasi segilima yang akan dibentuk ini dapat dilihat pada Gambar 10.



Gambar 9. Grafik fungsi periodik $\sin 2\theta$ untuk pembentukan pola formasi segiempat

Untuk dapat membentuk pola formasi segilima seperti pada gambar, diperlukan satu robot yang bertindak sebagai sudut. Namun berbeda dengan robot yang bertindak sebagai sudut pada pembentukan pola formasi segitiga dan segiempat, robot yang bertindak sebagai sudut pada pembentukan pola formasi segilima ini akan bergerak ke arah titik pusat lingkaran, alih-alih bergerak ke arah luar seperti biasa.



Gambar 10. Analisis transformasi pola formasi dari bentuk lingkaran hingga bentuk segilima

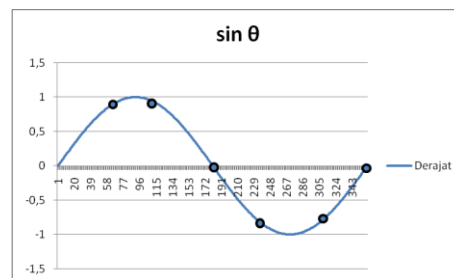
Dengan demikian, diperlukan satu puncak pada persamaan periodik yang akan dirancang, atau dengan kata lain, persamaan periodik yang dibentuk akan mempunyai satu gelombang untuk mendapatkan satu robot yang memiliki nilai variabel pembeda yang lebih tinggi dibanding nilai variabel pembeda pada robot lainnya. Jika diberi penomoran mengikuti arah jarum jam, robot yang menjadi sudut adalah robot nomor 2. Untuk itu, nilai variabel pembeda yang dimiliki robot tersebut haruslah lebih tinggi dari robot-robot tetangganya.

Berdasarkan analisis seperti ini, persamaan periodik dan nilai tujuan variabel pembeda yang digunakan pada pembentukan pola formasi segilima ini mengikuti persamaan 9.

$$u_i = \sin \theta_i$$

$$\theta_i = \frac{360 * i}{N} \tag{9}$$

Di mana i adalah penomoran dari masing-masing robot dan N adalah total jumlah robot yang digunakan pada simulasi. Selanjutnya, dari persamaan tersebut akan diperoleh bentuk grafik persamaan periodik seperti Gambar 11 di bawah.

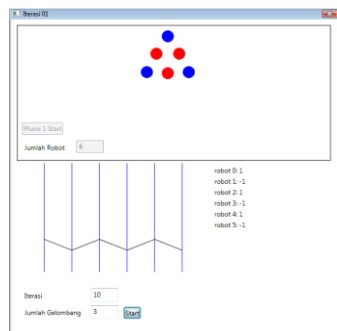


Gambar 11. Grafik fungsi periodik $\sin \theta$ untuk pembentukan pola formasi segilima

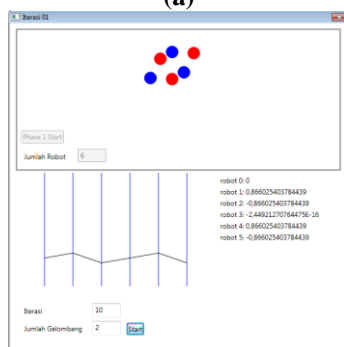
3. Eksperimen dan Hasil

3.1. Implementasi Berbasis Perangkat Lunak

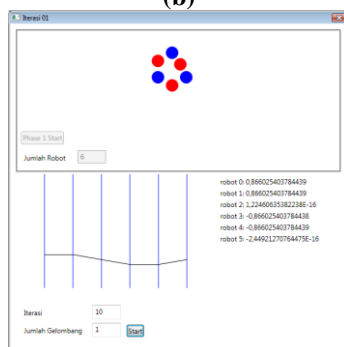
Tahapan penelitian kemudian akan dilanjutkan pada implementasi simulasi. Untuk keperluan ini maka akan dikembangkan bentuk simulasi yang dilakukan pada perangkat lunak dan perangkat keras. Untuk tahapan pada perangkat lunak simulasi dikembangkan dengan menggunakan aplikasi WPF (*Windows Programming Foundation*) dan *Microsoft Robotic Developer Studio* dari Microsoft. Tujuan dari pengembangan tahap ini adalah melakukan verifikasi persamaan dalam level simulasi yang berbasis animasi. Salah satu keunggulan dari penerapan berbasis perangkat lunak ini adalah bahwa kinerja algoritma dapat dievaluasi tanpa banyak dipengaruhi kondisi lingkungan simulasi. Implementasi berbasis perangkat lunak sendiri dilakukan dalam dua tahapan iterasi.



(a)



(b)



(c)

Gambar 12. Contoh simulasi berbasis perangkat lunak tahap iterasi pertama pada pembentukan pola formasi poligon dan verifikasi persamaan periodik

Hasil yang diperoleh pada tahap ini ditunjukkan pada Gambar 12 sebagai iterasi pertama. Pada layar simulasi, tidak hanya ditampilkan hasil akhir formasi yang diperoleh, tetapi juga dengan tampilan grafik dan nilai angka fungsi periodik yang digunakan untuk mendapatkan pola formasi tersebut. Dengan demikian proses verifikasi persamaan periodik yang digunakan juga dilakukan pada tahap iterasi ini. Pada tahap iterasi kedua dimaksudkan dalam rangka melihat efek pergerakan robot dalam pembentukan formasi. Proses koding pun sudah mendekati model arsitektur implementasi berbasis perangkat keras nantinya.

Selanjutnya, pada Gambar 13 terdapat contoh tampilan pada saat penerapan iterasi kedua (langkah-langkah pembentukan pola segitiga).



(a)



(b)



(c)

Gambar 13. Contoh simulasi berbasis perangkat lunak tahap iterasi kedua pada pembentukan pola formasi segitiga

3.2. Implementasi Berbasis Perangkat Keras

Setelah persamaan yang disusun berhasil diverifikasi dengan baik tahapan selanjutnya adalah implementasi pada robot hardware. Pada penelitian ini akan digunakan spesifikasi robot *Lego Mindstorm NXT* dengan detail seperti pada Tabel 1.

Tabel 1. Spesifikasi robot

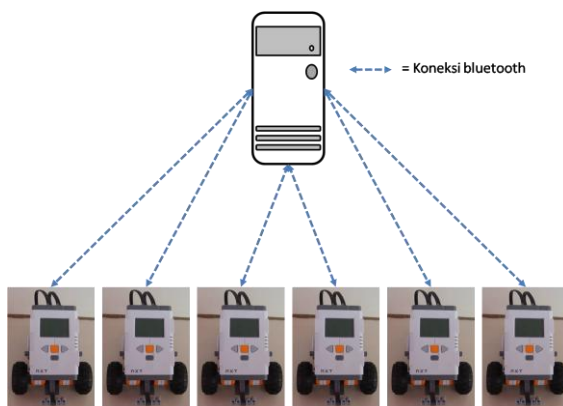
Nama robot	Tribot
Dimensi	13.5 X 18.5 X 18 cm
Berat	250 gram
Konsep pergerakan	Roda dengan formasi 3 roda
Metode pergerakan	<i>Differential drive</i>
Diameter roda depan	5.5 cm
Jarak antarroda depan	11.2 cm
Konektivitas	Bluetooth

Pada Gambar 14 terdapat tampilan dari perangkat keras yang digunakan dalam simulasi dilihat dari berbagai sisi, yaitu robot *Lego Mindstorm NXT*.



Gambar 14. Robot *lego mindstorm nxt* yang digunakan pada penelitian ini

Penerapan kode dilakukan dengan menggunakan aplikasi *Microsoft Robotic Developer Studio* yang membutuhkan *.NET Framework* agar dapat berjalan dengan baik. Karena spesifikasi robot *LEGO Mindstorm NXT* tidak mendukung hal tersebut, diperlukan spesifikasi tambahan berupa PC yang akan berfungsi sebagai modul komputasi. Komunikasi antarrobot dan PC pun dilakukan dengan fitur koneksi *Bluetooth*. Selanjutnya penerapan pada tahap perangkat keras akan mengikuti arsitektur sebagai berikut.



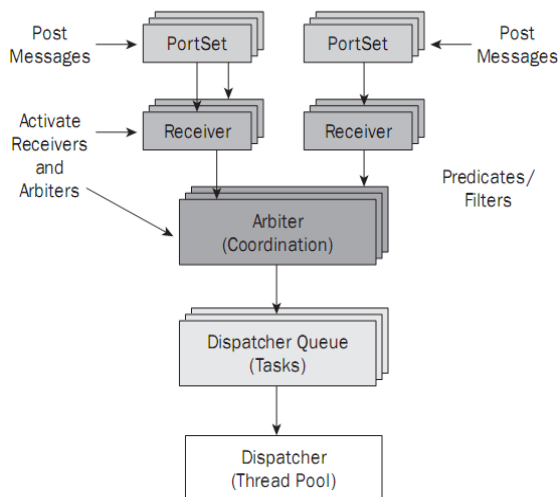
Gambar 15. Rancangan arsitektural implementasi simulasi berbasis perangkat keras

Selanjutnya, dalam penerapan kode perintah terhadap robot dalam simulasi dapat dilakukan dengan menggunakan model arsitektur CCR (*Coordination and Concurrency Runtime*) serta DSS (*Decentralized Software Service*) [11].

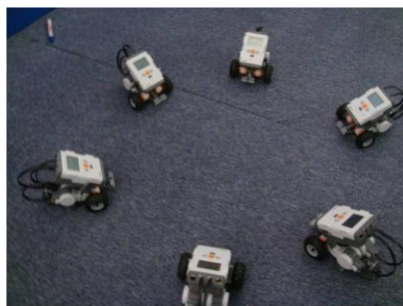
Coordination and Concurrency Runtime (atau CCR) adalah library tambahan untuk .NET

Framework yang dirancang untuk mengatur komunikasi yang terjadi secara *asynchronous* antara dua layanan yang bersifat *loosely coupled* (berkaitan secara renggang). CCR menggunakan model *message-passing* untuk memodelkan suatu proses *asynchronous* yang terjadi pada *service*. Model *message-passing* pada dasarnya adalah sebuah model untuk menangani antrian. Dalam CCR, data atau pesan yang akan dieksekusi datang melalui *PortSet*. Pesan-pesan tersebut akan berdatangan dan akan mengantri pada *PortSet* hingga pesan tersebut keluar dari antrian karena dipanggil oleh *receiver*. Proses pemanggilan ini bersifat *first-in-first-out*, sama dengan sifat antrian secara umum. Selain itu, pemanggilan pesan pada *receiver* bisa ditambahkan ekspresi logika, seperti AND (pesan dari dua *PortSet* harus datang terlebih dahulu) atau OR (pesan yang mencapai satu *PortSet* terlebih dahulu dari sekumpulan *PortSet* yang dipanggil). Saat *receiver* memanggil pesan dari *PortSet*, *arbiter* akan membuat suatu objek *Task* yang akan menggunakan pesan yang dipanggil oleh *receiver* untuk melakukan suatu aksi. *Task* ini akan mengantri pada *dispatcher queue* hingga dipanggil oleh *dispatcher*. *Task* yang dipanggil akan masuk pada *Thread Pool* dan akan dieksekusi pada komponen prosesor.

Sedangkan *Decentralized Software Services* (atau DSS) adalah sebuah model arsitektur perangkat lunak yang menitikberatkan pada konsep *service* dan perangkaian *service*. Dalam DSS, sebuah perangkat lunak adalah sekumpulan *service* yang dirangkai secara *loosely coupled*. Proses komunikasi antar-*service* berlangsung dengan memakai protokol DSSP (*Decentralized Software Services Protocol*), sebuah transfer protokol yang memenuhi kaidah REST. Arsitektur DSS berdiri di atas CCR, di mana CCR mengatur lalu-lintas dan pemrosesan pesan yang terjadi antar-*service*. Setiap *service* memiliki sebuah *Port* tempat menerima pesan dari *service* lain. *Port* pada sebuah *service* bisa terdiri dari dua macam, yaitu *Operations Port* dan *Notifications Port*. *Operations Port* digunakan untuk tempat mengantri pesan-pesan yang masuk pada klasifikasi *Operation*. Begitu juga dengan *Notifications Port*, digunakan sebagai tempat mengantri pesan-pesan yang bertipe *notification*. Pesan yang masuk pada klasifikasi *operation* adalah pesan yang digunakan pada operasi sehari-hari *service* tersebut. Pesan *notification* berguna sebagai pesan untuk berlangganan pada *service* tersebut. *Service* yang berlangganan pada *service* lain akan menerima perubahan status *service* yang dilanggan secara berkala. Pesan-pesan yang diterima pada *Port Operations* maupun *Port Notifications* akan diolah pada *handler-handler* yang tersedia di dalam *service* tersebut. Hasil olahan pesan tersebut akan dikirimkan kepada layanan yang bersangkutan melalui *port service* yang bersangkutan.



Gambar 16. Arsitektur dan alur proses yang terjadi pada CCR (*Coordination and Concurrency Runtime*)



(a)



(b)

Gambar 17. Contoh tampilan simulasi berbasis perangkat keras pada pembentukan pola segitiga

Pada Gambar 16 terdapat representasi visual dari CCR. Bersama dengan DSS, kedua arsitektur ini kemudian ini digunakan sebagai model perintah pada implementasi berbasis perangkat keras. Sementara pada Gambar 17 dapat dilihat contoh hasil simulasi yang didapatkan pada tahap ini (proses pembentukan pola formasi lingkaran, kemudian pola formasi segitiga).

4. Kesimpulan

Dari hasil yang diperoleh pada penelitian ini kemudian dapat disimpulkan bahwa proses pembagian fungsi dan peran dapat diterapkan pada

koloni robot otonom bergerak untuk menyelesaikan suatu permasalahan tertentu. Adanya perbedaan peran ini akan membantu robot menentukan posisinya masing-masing dalam kelompok, sehingga proses pencapaian tujuan yang lebih besar dapat dilakukan melalui pembagian tugas spesifik yang berbeda. Proses pembagian tugas diharapkan hanya menggunakan interaksi lokal yang bersifat sederhana, tanpa adanya kebutuhan bagi robot untuk mengetahui informasi lingkungan secara global. Selain itu, penggunaan robot yang homogen di dalam kelompok perlu agar aplikasi yang dikembangkan dan proses pembagian tugasnya tidak bergantung pada spesifikasi robot yang digunakan. Pada contoh kasus pembentukan pola formasi poligon dengan menggunakan robot *Lego Mindstorm NXT* ini, pembagian fungsi dan peran diterapkan melalui persamaan periodik yang mempunyai sifat berulang setiap periode tertentu. Penerapan metode secara gradual atau bertahap dimaksudkan untuk menyusun metode yang lebih menyederhanakan permasalahan. Selanjutnya dengan mengubah parameter persamaan yang digunakan akan dapat diperoleh bentuk pola formasi poligon yang berbeda-beda.

REFERENSI

- [1] Jatmiko, Wisnu, Adila A Krisnadi, Rizki Mardian, and Nulad W Pambudi. "Aplikasi jaringan komunikasi robot multihop terdistribusi pada lingkungan statis terbatas: implementasi, simulasi dan analisis pada kasus robot lego mindstorm NXT". Faculty of Computer Science – University of Indonesia, 2009.
- [2] Jatmiko, Wisnu et al. "evolutionary computation approach for odor source localization problem: practical review", in *IEEE Transaction on Evolutionary Computation*, 2009.
- [3] Jatmiko, Wisnu, Kosuke Sekiyama and Toshio Fukuda, "A PSO-based mobile robot for odor source localization in extreme dynamic advection-diffusion environment with obstacle: theory, simulation and measurement", in *IEEE Computational Intelligence Magazine: Special Issue on Biometric*. Vol. 2, Issue 2, 2007, pp. 37-51.
- [4] Jatmiko, Wisnu et al, "Traffic signal control modification based on self-organizing in Indonesia", in *IEEE Transactions on Evolutionary Computation*, (to be submitted)
- [5] Takahashi, Junji, Kosuke Sekiyama dan Toshio Fukuda, "Cooperative object tracking with mobile robotic sensor network", Departement of Micro-Nano Systems Engineering Nagoya University, 2007.
- [6] Pambudi, Nulad W, *Pembentukan Formasi Pola Robot Homogen secara Gradual: Studi Kasus Pada Robot Lego Mindstorms NXT*. Faculty of Computer Science – University of Indonesia, 2009.

- [7] Ikemoto, Yusuke, Yasuhisa Hasegawa, Toshio Fukuda and Kazuhiko Matsuda, "Gradual spatial pattern of homogeneous robot group", Departement of Micro-Nano Systems Engineering Nagoya University, 2004.
- [8] Ikemoto, Yusuke, "A research on pattern formation on robot group", Departement of Micro-Nano Systems Engineering Nagoya University, 2005.
- [9] Mardian, Rizki, *Penanganan Jaringan Komunikasi Multihop Terkonfigurasi Sendiri pada Kasus Dua atau Banyak Sumber dengan Menggunakan Koloni Robot Otonom Terdistribusi Berdasarkan Prinsip Kecerdasan Kolektif serta Pengembangan Simulator Naml*, Faculty of Computer Science – University of Indonesia, 2009.
- [10] Strogatz, Steven H, *Nonlinear Dynamics and Chaos with Applications to Physics, Biology, Chemistry and Engineering*, Westview Press, 1994.
- [11] Johns, Kyle, and Trevor Taylor, *Microsoft Robotics Developer Studio Professional*. Indianapolis: Wiley Publishing Inc, 2008.