

Strategi *Self-Assembly* Paralel pada *Swarm Robot*

NIKEN SYAFITRI

Teknik Elektro Institut Teknologi Nasional Bandung
Email: nikensyafitri@itenas.ac.id

Received 21 November 2018 | *Revised* 10 Desember 2018 | *Accepted* 7 Januari 2019

ABSTRAK

Dari banyaknya strategi yang diusulkan untuk proses self-assembly pada swarm robotics, hanya beberapa grup riset berkonsentrasi di bidang ini yang mengusulkan proses paralel pada penggabungan antar robot. Tetapi, strategi ini hanya digunakan ketika sebuah robot memerlukan tumpuan dari dua robot atau lebih pada satu waktu. Berdasar pada kebutuhan untuk menyebarkan ratusan hingga ribuan robot pada satu swarm, strategi penggabungan antar robot satu-demi-satu memerlukan waktu yang sangat lama untuk diselesaikan. Di artikel ini, strategi self-assembly antar robot pada suatu swarm secara paralel diusulkan untuk mengurangi waktu proses self-assembly dengan menempatkan sejumlah robot di posisi tertentu. Saat penggabungan, robot-robot ini akan bergerak menempatkan dirinya sesuai dengan posisi akhir yang ditargetkan. Hasil menunjukkan bahwa strategi ini dapat mereduksi waktu proses self-assembly hingga setengah dari waktu yang diperlukan dengan proses penggabungan satu-demi-satu.

Kata kunci: *swarm robot, self-assembly, proses paralel*

ABSTRACT

Despite the number of strategies proposed for self-assembly process in swarm robotics, only few research groups working in this area have proposed the parallel process of robots assembled each other. However, this strategy only works when a robot needs to be supported by two or more robots in a time. When deploying hundred to thousand robots in a swarm is required, the strategy of robots connecting to the structure of assembled robots in a one-by-one manner requires an extremely long time to accomplish. In this paper, a strategy of parallel self-assembly for robots in a swarm is proposed for reducing the self-assembly process time by placing a number of robots at particular positions. While connecting, they will move to position themselves appropriately to the targeted final structure. Result shows that this strategy can reduce the process of self-assembly time up to half of the time required for one-by-one process.

Keywords: *swarm robots, self-assembly, parallel process*

1. PENDAHULUAN

Serangga sosial dapat menyelesaikan pekerjaan yang kompleks ketika berkolaborasi dalam bekerja tetapi mustahil dilakukan ketika mereka bekerja secara individual. Sebagai contoh, lebah membangun sarangnya; rayap membangun sarangnya dengan ketinggian yang ekstrem, lengkap dengan susunan ruangan, koridor dan ventilasi yang kompleks; dan semut dapat menyeberangi celah. Terinspirasi dari perilaku ini, riset di *swarm robotics* meniru kinerja serangga sosial (**Bonabeau, 1999**). *Swarm robotics* didefinisikan sebagai studi perancangan robot dalam jumlah yang relatif banyak, yang dapat bekerja secara kolektif untuk menyelesaikan tugas tertentu (**Şahin, 2005**). Dalam pengaplikasiannya, karakteristik *swarm robotics* harus mencakup beberapa keunggulan seperti *robustness*, *flexibility*, dan *scalability* (**Martinolli, 2001**). *Swarm robotics* memiliki beberapa fitur spesifik yang membedakannya dengan sistem multirobot yang lain, seperti (i) *autonomous* dan *mobile*; (ii) kontrol atau koordinasinya terdesentralisasi (**Winfield, 2012**); (iii) jumlah robot dalam *swarm* harus banyak atau *scalable*; (iv) robot-robot dalam *swarm* memiliki kemampuan penginderaan dan komunikasi yang terbatas; (v) robot-robot dalam *swarm* homogen, atau jika heterogen, setiap tipe robot harus dalam jumlah yang banyak; dan (vi) robot-robot dalam *swarm* dapat berperilaku secara kolektif melalui interaksi lokal antar robot atau antara robot dengan lingkungannya (**Beni, 2005; Şahin, 2005**).

Self-assembly adalah proses penggabungan *autonomous* dari komponen-komponen yang sebelumnya terpisah menjadi sebuah atau beberapa pola atau struktur tanpa adanya intervensi manusia (**Groß, 2008**); perilaku ini berfungsi untuk formasi, replikasi, pertumbuhan, rekonfigurasi, perbaikan sub-modul, dan pertumbuhan berorientasi tugas. Perilaku ini juga mendukung kinerja *swarm robotics* untuk penyederhanaan kontrol, kemudahan tugas, peningkatan kemampuan evolusi, pendukung munculnya perilaku baru, pendukung eksplorasi mandiri dan *scalability* (**Bongard, 2014**). Dalam taksonomi *swarm robotics*, *self-assembly* diklasifikasikan ke dalam masalah yang harus diselesaikan oleh *swarm robot* (**Bayindir, 2007**), dan bagian dari perilaku kolektif-perilaku terorganisasi secara spasial (**Brambilla, 2013**). Secara umum struktur yang terbentuk dari hasil *self-assembly* adalah struktur (i) rantai (satu dimensi), struktur yang terbentuk berupa garis panjang; (ii) *mesh* (dua dimensi), struktur yang terbentuk berupa poligon tak beraturan; dan (iii) *cluster* (tiga dimensi), struktur yang terbentuk berupa prisma tak beraturan (**Anderson, 2002**). Namun untuk robotika, struktur dikelompokkan berdasarkan geometri yang terbentuk, yaitu rantai, kisi, ikatan (*truss*), dan variasi bentuk yang lain (**Gilpin, 2010**).

Self-assembly pada *swarm robotics* diaplikasikan untuk menyelesaikan masalah seperti membantu robot-robot memanjat rintangan yang tidak dapat dilakukan jika robot bekerja secara individual (**Kuyucu, 2013; Li, 2016**). Salah satu konsorsium di bidang *swarm robotics* mengaplikasikan *self-assembly* untuk menghindari lubang besar, menyeberangi celah, memanjat bukit, navigasi untuk segala medan, dan operasi penyelamatan (**Mondada, 2004; O'Grady, 2012; Trianni, 2014; Bruni, 2015; Matthews, 2015**). Sementara konsorsium yang lain mengaplikasikan *self-assembly* untuk pencarian sumber energi listrik jika *swarm robot* dilepas di lingkungan tanpa ada bantuan manusia (**Kornienko, 2007; Meister, 2013; Levi, 2014; Vonásek, 2015**).

Sayangnya, mayoritas proses *self-assembly* dilakukan dengan menggabungkan robot satu-per-satu ke struktur. Beberapa strategi mengimplementasikan penggabungan tiga robot sekaligus, seperti pada proyek Swarmanoid (**Dorigo, 2013**). Terdapat tiga tipe robot pada proyek ini, yaitu *eye-bot* (berupa *drone*, terbang), *foot-bot* (memiliki roda dan penjepit, sehingga bisa berpindah lokasi dan menjepit robot lain untuk *self-assembly*), dan *hand-bot* (hanya memiliki

penghisap untuk menempel ke langit-langit ruangan). Agar *hand-bot* dapat bergerak, robot ini harus ditopang oleh dua buah *foot-bot* yang menjepit *hand-bot* secara bersamaan. Strategi lain diimplementasikan oleh proyek Replicator and Symbion (**Kernbach, 2011**), yang menerapkan heterogenitas robot pada *swarm*. Strategi penggabungan tiga robot dilakukan pada operasi penyelamatan robot yang rusak.

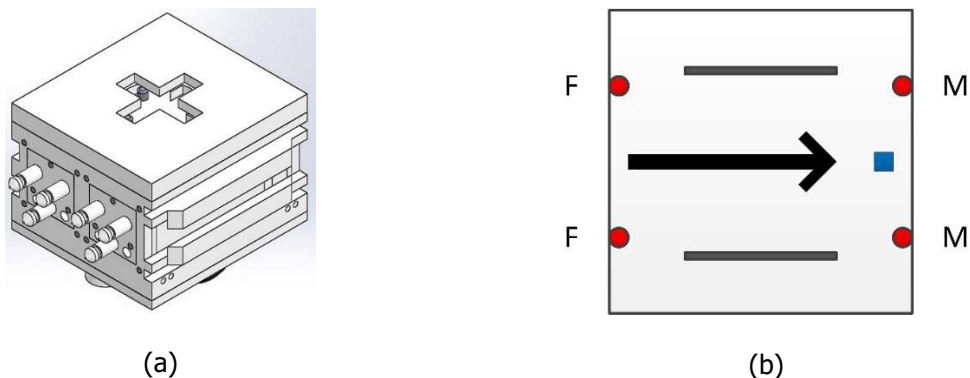
Implementasi penggabungan beberapa robot pada satu waktu baru ditujukan untuk kemudahan gerak atau operasi penyelamatan. Berdasar pada studi kasus *swarm robot* dengan kemampuan adaptif yang melibatkan ratusan, dan menuju ke ribuan robot pada suatu *swarm*, penggabungan robot satu-per-satu ke struktur memerlukan waktu yang sangat lama (**Syafitri, 2018**). Strategi ini akan berdampak ke terhentinya operasi *swarm* untuk menyelesaikan tugas tertentu karena keterbatasan energi.

Dalam tulisan ini, strategi *self-assembly* secara paralel diusulkan untuk mengurangi waktu yang diperlukan agar robot dapat menyelesaikan tugas yang diberikan dengan baik. Tulisan ini terdiri dari empat bagian. Metodologi penelitian akan dipaparkan di Bagian 2, hasil simulasi dan analisis didiskusikan di Bagian 3, dan ditutup dengan kesimpulan di Bagian 4.

2. METODOLOGI

2.1 Model *Swarm Robot*

Model *swarm robot* yang digunakan adalah CuBot, robot yang khusus didesain untuk keperluan *self-assembly* untuk membangun struktur kompleks (**Syafitri, 2018**). Robot berbentuk balok dengan panjang dan lebar yang sama, masing-masing 10cm, dan tinggi balok 7cm. Sistem penggerak robot menggunakan *differential-drive steering* dengan penopang dua buah *adjustable caster* dengan sistem suspensi bawaan. Robot kemudian dimodelkan ke dalam bentuk dua dimensi (2D) tampak atas seperti terlihat pada Gambar 1. Arah panah menunjukkan arah gerak maju robot. Kotak biru pada model 2D menunjukkan bagian depan robot. Sedangkan bulatan merah bertanda M adalah konektor bagian depan (*male*) berupa sepasang susunan pin, dan bulatan merah bertanda F adalah sepasang konektor bagian belakang (*female*) berupa piringan pengunci. Kedua tipe konektor ini berfungsi untuk melakukan penggabungan robot dengan bentuk barisan memanjang ke belakang. Proses *self-assembly* dipandu oleh sensor inframerah yang terpasang di masing-masing konektor. Sementara dua persegi panjang abu-abu merepresentasikan kedua roda dari robot.



Gambar 1. (a) CuBot dan (b) Model *Swarm Robot*

Untuk *self-assembly* bagian samping, konektor yang digunakan adalah sistem *sliding* dengan desain khusus sehingga beban robot akan tertopang jika robot tidak memijak permukaan tanah. Penguncian koneksi dilakukan dengan penghentian gerakan maju robot yang bergabung dengan struktur saat pemasangan sensor inframerah di bagian depan dan belakang sistem *sliding* menginformasikan bahwa mereka sudah menerima sinyal dari sensor inframerah robot acuan pada struktur.

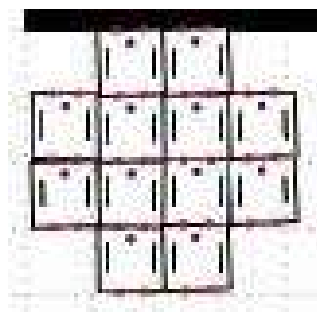
Ukuran model robot seperti panjang dan lebar, ukuran roda baik jarak antar roda, lebar dan radius roda, diinformasikan pada Tabel 1, beserta kecepatan maksimum robot. Ukuran roda diperlukan untuk perhitungan gerakan robot dan pembatasan kecepatan maksimum robot, kecuali parameter lebar roda. Lebar roda diminimumkan di pemodelan agar proses *self-assembly* pada simulasi terlihat dengan jelas.

Tabel 1. Parameter Model *Swarm Robot*

Parameter	Besaran
Panjang robot	10cm
Lebar robot	10cm
Jarak antar roda	7cm
Lebar roda	0.1cm
Radius roda	2.1cm
Kecepatan maksimum	14m/s

2.2 Algoritma

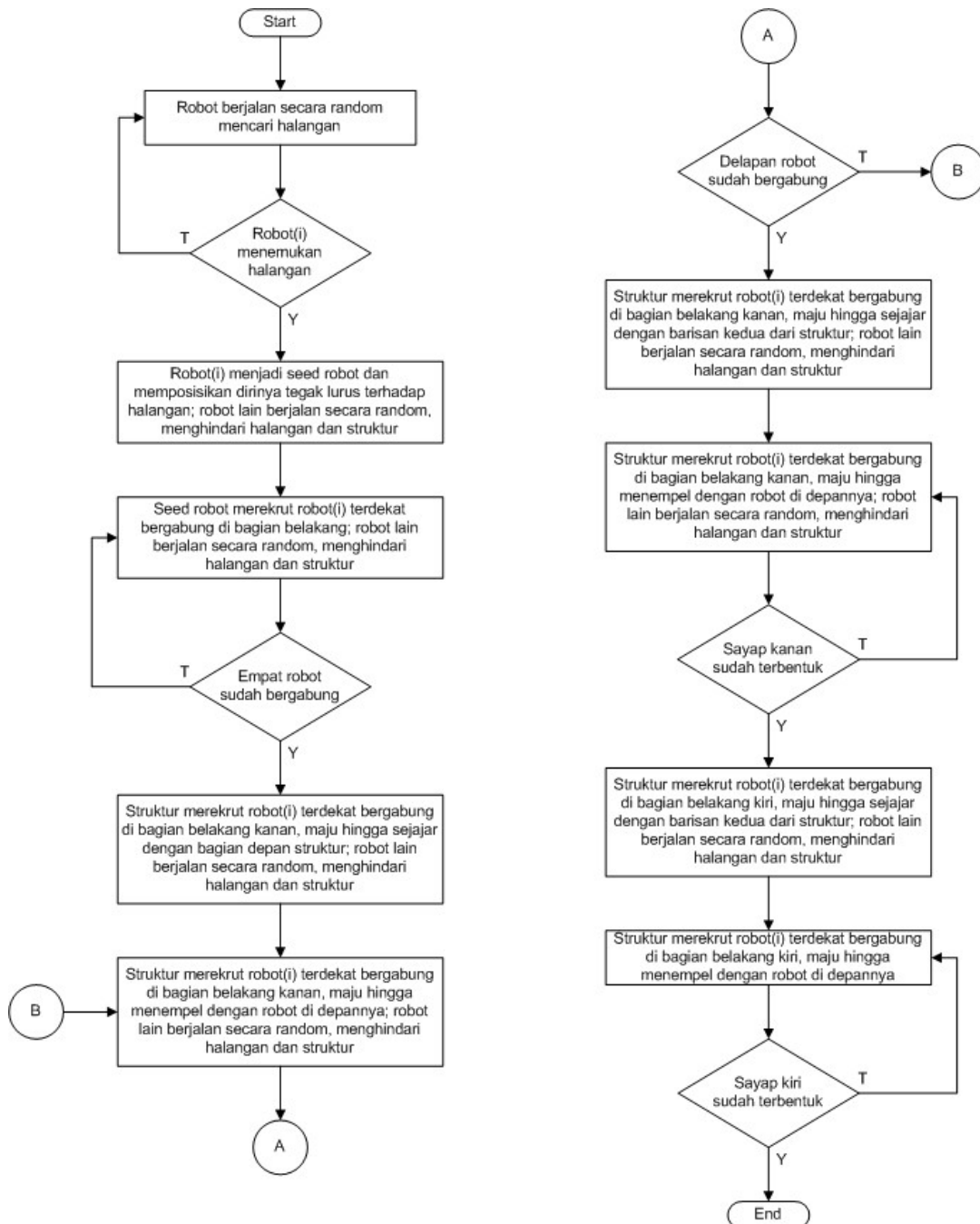
Algoritma simulasi terbagi menjadi dua, yaitu algoritma *self-assembly* satu-per-satu dan algoritma *self-assembly* paralel. Di kedua sistem *self-assembly*, robot disusun secara teratur di zona awal, lalu bergerak menggunakan *random walk* – pergerakan robot secara acak dengan memberi probabilitas arah kiri dan kanan yang sama sebesar 0.5, lalu mencari halangan berupa lubang besar memanjang. Jika salah satu robot mendeteksi lubang tersebut, proses pencarian lubang dihentikan dan proses *self-assembly* dimulai. Sementara robot yang sedang direkrut untuk bergabung dengan struktur bergerak mendekati struktur, robot lain yang belum direkrut masih berjalan secara *random*, namun menghindari struktur dan lubang. Proses *self-assembly* berhenti jika struktur target terbentuk berupa dua barisan memanjang yang masing-masing terdiri dari empat robot, dengan tambahan sayap di samping-sampingnya yang masing-masing terdiri dari dua robot, seperti terlihat pada Gambar 2.



Gambar 2. Struktur Target

2.2.1 Algoritma *Self-Assembly* Satu-Per-Satu

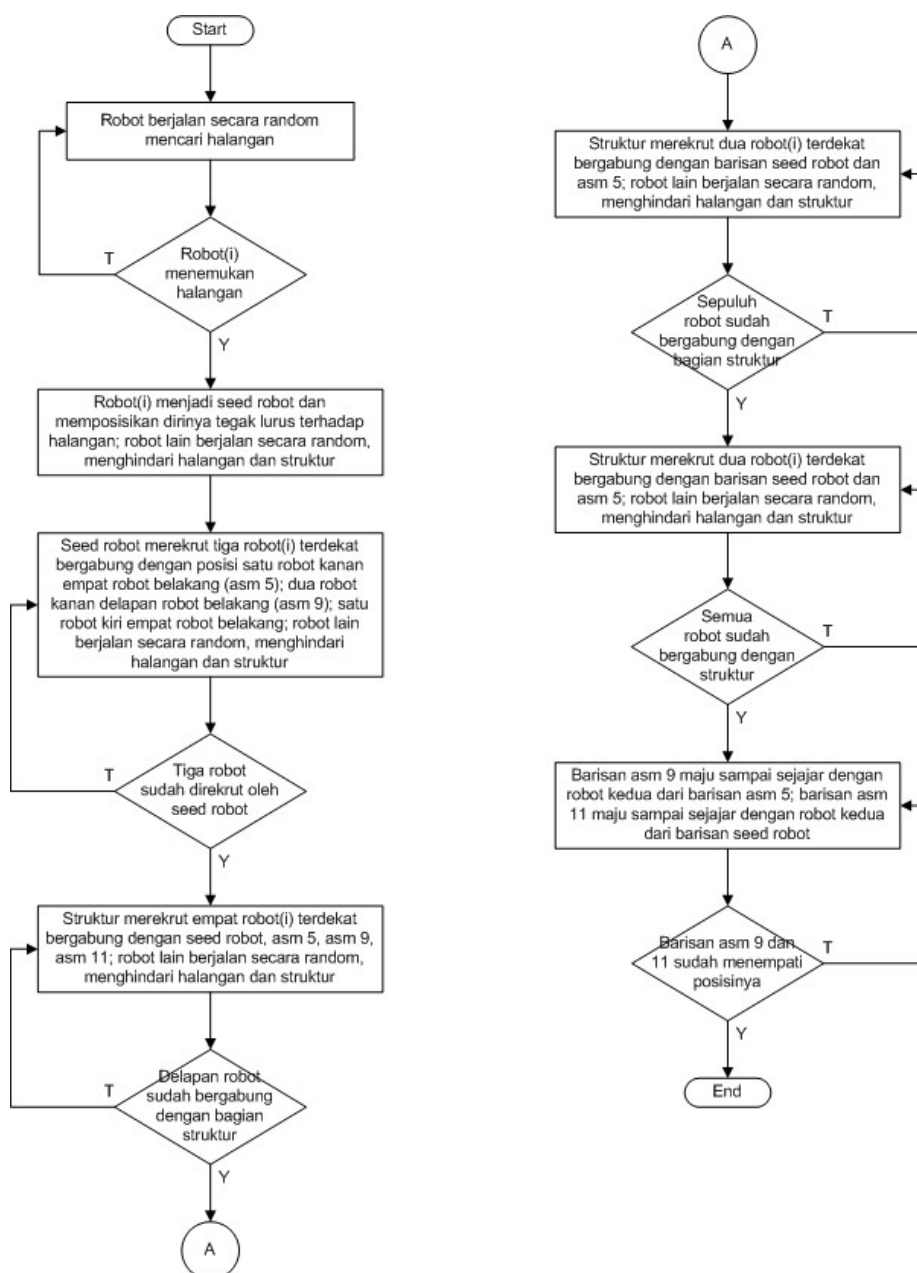
Untuk strategi *self-assembly* satu-per-satu, algoritma yang diterapkan ditampilkan pada Gambar 3. Saat simulasi dimulai, semua robot berjalan dengan arah *random* mencari halangan. Jika salah satu robot menemukan halangan, robot tersebut menjadi *seed robot* dan memposisikan dirinya tegak lurus terhadap halangan, lalu merekrut robot lain untuk bergabung membentuk sebuah struktur garis lurus. Robot yang belum direkrut tetap berjalan secara *random* sekaligus menghindari struktur dan halangan. Setelah empat robot bergabung, struktur merekrut empat robot berikutnya dengan posisi garis lurus di sebelah garis lurus pertama. Setelah dua baris terbentuk, struktur lalu merekrut robot membentuk sayap kanan, kemudian sayap kiri, dan proses *self-assembly* berhenti.



Gambar 3. Flowchart *Self-Assembly* Satu-Per-Satu

2.2.2 Algoritma *Self-Assembly* Paralel

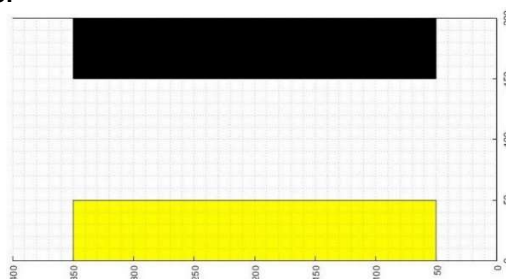
Sedangkan untuk strategi *self-assembly* paralel, algoritma yang diterapkan diilustrasikan pada Gambar 4. Ketika simulasi dimulai semua robot berjalan dengan arah acak untuk mencari halangan. Saat salah satu robot menemukan halangan, robot tersebut memposisikan dirinya tegak lurus terhadap halangan dan menjadi *seed robot*; sementara robot lain berjalan dengan arah *random*, menghindari *seed robot* dan halangan. *Seed robot* lalu merekrut tiga robot sekaligus untuk menempati posisi tertentu dengan dirinya sebagai acuan posisi. Lalu bagian-bagian struktur ini masing-masing merekrut satu robot untuk bergabung. Setelah itu dua bagian struktur yang menjadi baris utama dan baris kedua merekrut masing-masing satu robot, diikuti lagi perekrutan yang terakhir. Setelah baris utama, baris kedua dan dua baris samping terbentuk, baris samping akan maju bergabung ke baris utama dan baris kedua sebagai sayap. Lalu gabungan baris kedua dan sayapnya maju bergabung ke baris utama, dan proses *self-assembly* selesai.



Gambar 4. *Flowchart Self-Assembly* Paralel

3. SIMULASI

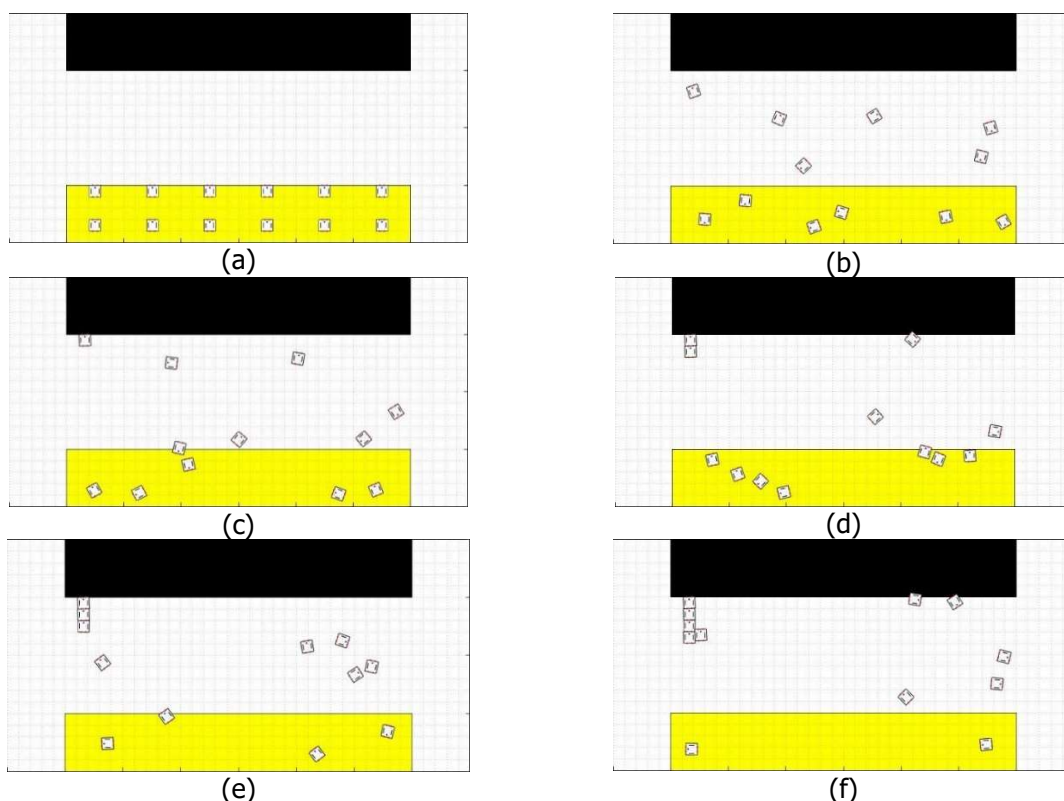
Kedua strategi *self-assembly* disimulasikan menggunakan MATLAB 2018a. Sebanyak dua belas robot diposisikan di zona awal pada arena berukuran 4m x 2m, seperti terlihat pada Gambar 5. Zona awal berukuran 3m x 0.5m dan halangan berupa lubang hitam persegi panjang berukuran sama dengan zona awal. Kedua strategi masing-masing disimulasikan sebanyak tiga puluh kali, dan perekaman video diambil pada simulasi ketiga puluh. Robot dengan ID 1 sampai dengan 6 ditempatkan di bagian bawah, sedangkan robot dengan ID 7 sampai dengan 12 ditempatkan di bagian atas.



Gambar 5. Arena Simulasi

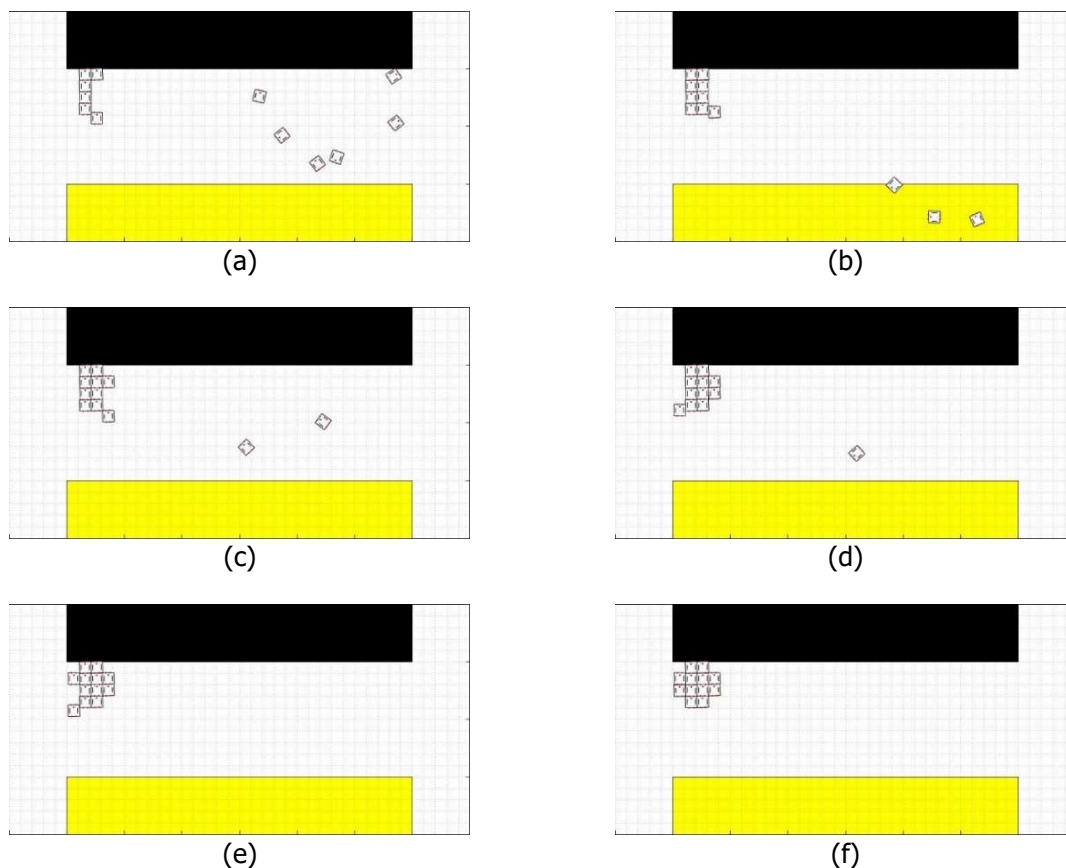
3.1 Simulasi *Self-Assembly* Satu-Per-Satu

Hasil simulasi untuk algoritma *self-assembly* satu-per-satu diperlihatkan di Gambar 6 dan Gambar 7. Pada Gambar 6.a, terlihat dua belas robot menempati posisinya di zona awal. Lalu robot berjalan secara random mencari halangan, Gambar 6.b, dan sebuah robot menemukan lubang hitam persegi panjang, menjadi *seed robot* yang menginisiasi proses *self-assembly*, dan memposisikan dirinya tegak lurus terhadap lubang, Gambar 6.c. *Seed robot* lalu merekrut robot terdekat untuk bergabung di belakangnya, Gambar 6.d. Lalu struktur berturut-turut merekrut dua buah robot untuk bergabung di belakangnya, Gambar 6.e dan Gambar 6.f.



Gambar 6. Simulasi *Self-Assembly* Satu-Per-Satu

Kemudian struktur merekrut robot terdekat untuk bergabung dengan menempel di bagian kanan belakang untuk bergerak maju dan robot yang direkrut terkoneksi dengan sistem sliding. Robot lalu maju hingga sejajar dengan *seed robot*, lalu struktur merekrut robot lain yang terdekat, Gambar 7.a. Setelah empat buah robot membentuk barisan baru, struktur merekrut robot terdekat untuk membentuk sayap kanan, Gambar 7.b. Perekrutan berlanjut hingga dua robot bergabung di barisan sayap kanan, dengan robot terdepan sejajar dengan robot ke dua dari barisan utama, Gambar 7.c. Kemudian struktur merekrut robot terdekat untuk bergabung membentuk sayap kiri, Gambar 7.d. Perekrutan berlanjut hingga dua robot dengan robot terdepan di sayap kiri sejajar dengan robot ke dua di barisan utama, Gambar 7.e. Proses *self-assembly* berhenti ketika robot terakhir di sayap kiri sudah terkoneksi dengan robot di depannya, Gambar 7.f.

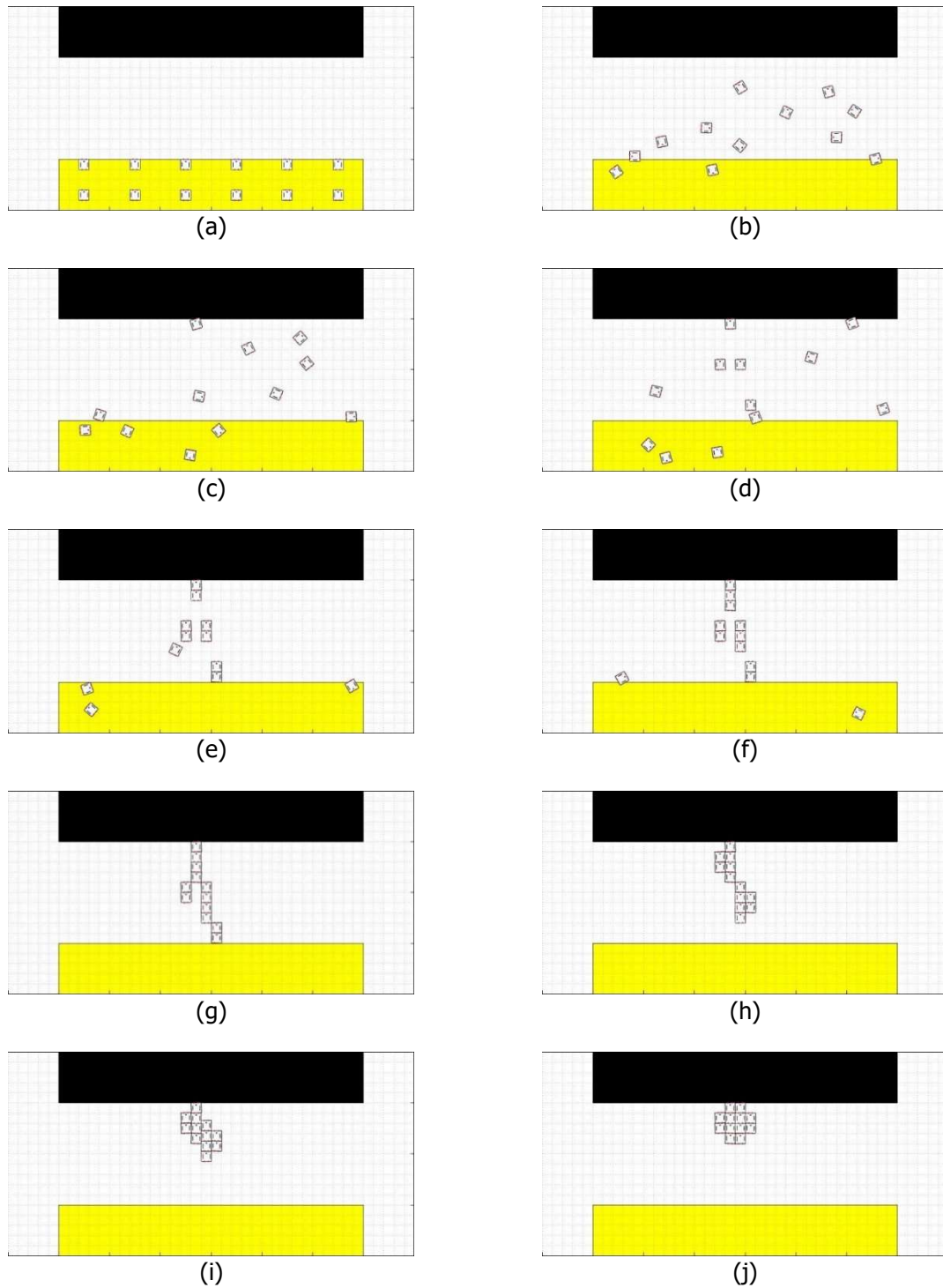


Gambar 7. Simulasi *Self-Assembly* Satu-Per-Satu

3.2 Simulasi *Self-Assembly* Paralel

Untuk hasil simulasi *self-assembly* paralel, diperlihatkan di Gambar 8. Di awal, dua belas robot ditempatkan di posisi yang sama dengan simulasi *self-assembly* satu-per-satu, Gambar 8.a. Kemudian robot berjalan secara random mencari lubang, Gambar 8.b. Robot yang menemukan lubang pertama kali menjadi *seed robot* dan memosisikan dirinya tegak lurus dengan lubang, Gambar 8.c. Setelah posisi *seed robot* tegak lurus terhadap lubang, tiga robot terdekat direkrut untuk menempati posisi tertentu di bagian kanan dan kiri belakang terhadap *seed robot*, Gambar 8.d. Setelah ketiga robot menempati lokasinya dan memosisikan dirinya tegak lurus terhadap lubang, masing-masing robot merekrut robot terdekat untuk bergabung, Gambar 8.e. Barisan *seed robot* dan barisan tepat di sebelah kanan belakangnya masing-masing merekrut robot terdekat, Gambar 8.f dan Gambar 8.g. Kemudian barisan sayap kanan dan kiri, yang masing-masing terdiri dari hanya dua robot, bergerak sampai robot terdempannya sejajar dengan robot ke dua dari barisan tengah yang masing-masing terdiri dari empat robot, Gambar

8.h. Setelah masing-masing sayap menempati posisinya, bagian struktur yang memiliki *seed robot* tetap diam di posisinya, sementara bagian yang lain, yaitu barisan tengah kanan dan sayap kanan bergerak maju, Gambar 8.i, sampai posisi robot terdepan dari barisan tengah kanan sejajar dengan posisi *seed robot*, Gambar 8.j, dan proses *self-assembly* berhenti.

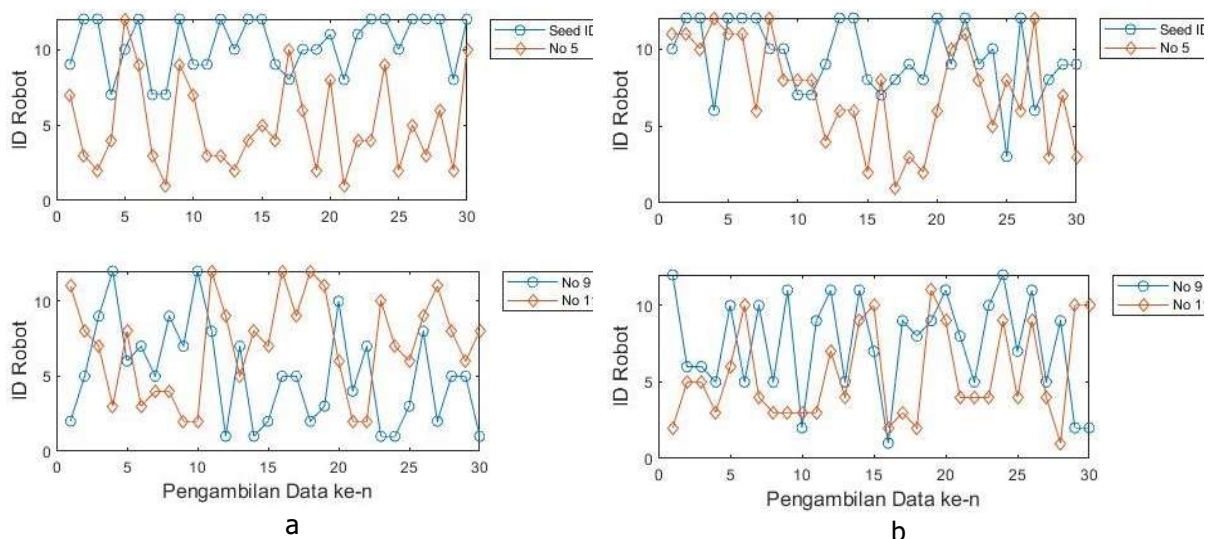


Gambar 8. Simulasi *Self-Assembly* Paralel

3.3 Perbandingan Kedua Strategi *Self-Assembly*

Beberapa data penting direkam selama simulasi selain perekaman video pada simulasi ke tiga puluh, yaitu ID robot yang menjadi *seed robot*, robot asm 5, 9, dan 11; lokasi inisiasi proses *self-assembly*; dan waktu simulasi dan waktu *self-assembly* yang diperlukan.

Untuk kedua strategi, robot yang mendeteksi lubang pertama kali bervariasi, namun mayoritas dilakukan oleh robot dengan ID 7 sampai dengan 12, Gambar 9. Hal ini disebabkan posisi robot-robot dengan ID tersebut lebih dekat dengan lubang yang menjadi target posisi *self-assembly*. Sedangkan untuk posisi robot asm 5, 9, dan 11, ID robot-robot dengan posisi ini lebih bervariasi. Hanya saja untuk posisi robot asm 5 pada proses *self-assembly* satu-per-satu, mayoritas posisi ini didapat oleh robot dengan ID 1 sampai dengan 6, yang merupakan komplement dari *seed robot*. Bervariasinya ID robot untuk *seed robot*, asm 5, 9, dan 11, adalah pembuktian bahwa tidak ada pengesetan awal robot mana yang menjadi pemimpin pada *swarm*, sehingga semua robot memiliki kesempatan untuk menjadi *seed robot* yang menginisiasi terbentuknya struktur.

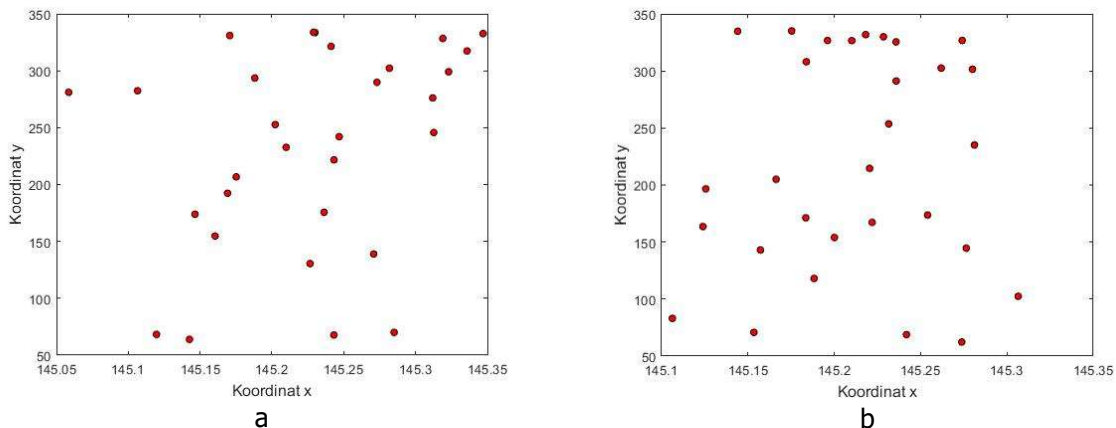


Gambar 9. ID Robot yang Menjadi *Seed Robot*, (a) Proses *Self-Assembly* Satu-Per-Satu, (b) Proses *Self-Assembly* Paralel

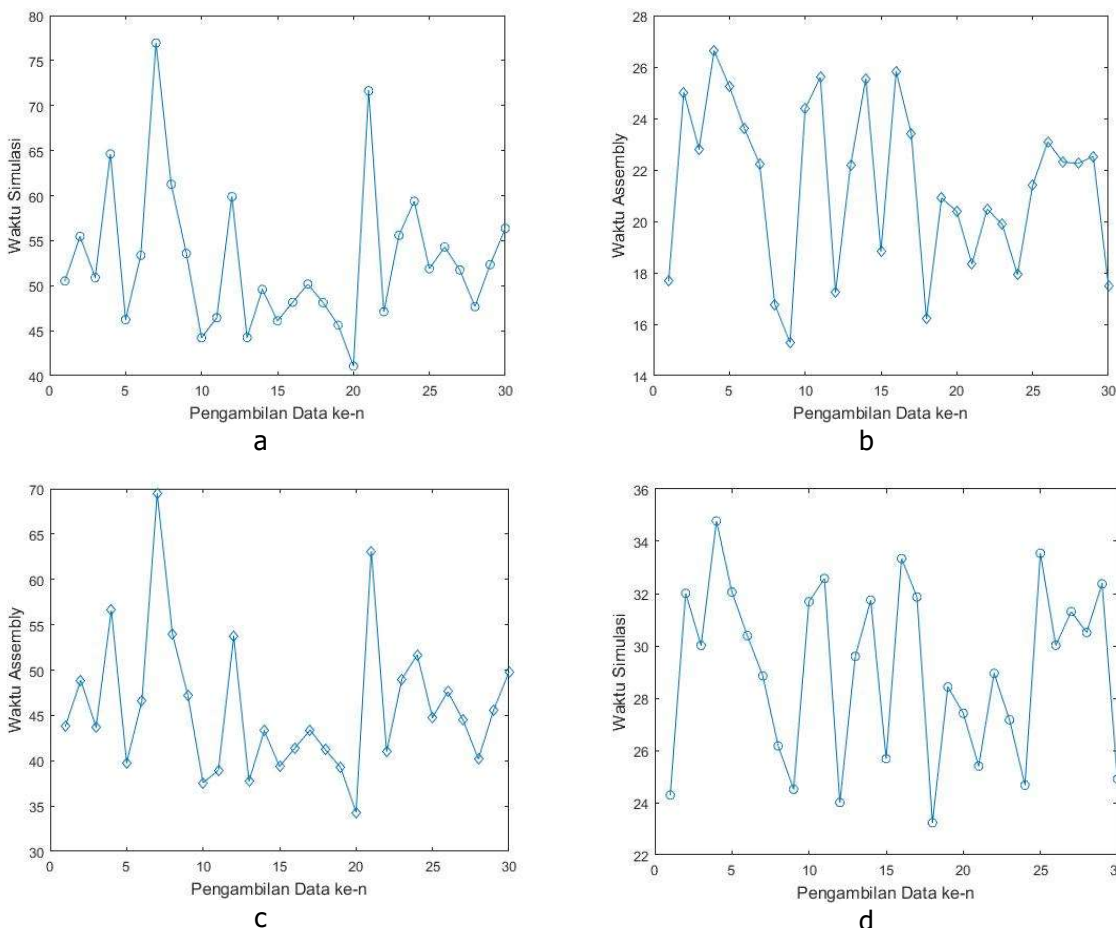
Sedangkan untuk lokasi inisiasi pembentukan struktur melalui proses *self-assembly*, dengan koordinat-x yang relatif sama (145.xx) karena posisi lubang sama secara melebar, didapat koordinat-y yang bervariasi, Gambar 10. Hal ini membuktikan bahwa robot-robot dalam *swarm* tidak mengetahui posisi halangan. Sebagai catatan, untuk tiga puluh kali pengambilan data, inisiator pembentukan struktur dilakukan oleh *seed robot* yang bervariasi, bukan hanya satu robot yang diset sebagai *seed* untuk beberapa kali pengulangan simulasi.

Untuk waktu simulasi, Gambar 11, yaitu waktu dari inisiasi, proses mencari halangan hingga selesai proses *self-assembly*, dari tiga puluh kali simulasi, untuk proses *self-assembly* satu-per-satu, Gambar 11.a, didapat waktu minimum 41.05detik, waktu maksimum 76.95detik, dan waktu rata-rata 52.81detik. Sedangkan dari proses *self-assembly* paralel, Gambar 11.b, didapat waktu minimum 23.23detik, waktu maksimum 34.78detik, dan waktu rata-rata 29.05detik. Sedangkan untuk waktu proses *self-assembly*, dari proses *self-assembly* satu-per-satu, Gambar 11.c, didapat waktu minimum 34.3detik, waktu maksimum 69.46detik, dan waktu rata-rata 45.91detik. Untuk proses *self-assembly* paralel, Gambar 11.d, didapat waktu minimum 15.28detik, waktu maksimum 26.63detik, dan waktu rata-rata 21.38detik. Dari hasil-hasil tersebut, strategi *self-assembly* paralel dapat mereduksi waktu proses *self-assembly* dan waktu simulasi hingga lebih dari setengah waktu yang diperlukan oleh strategi *self-assembly*

satu-per-satu. Dan bervariasinya waktu yang diperoleh dari keseluruhan simulasi disebabkan oleh posisi struktur yang acak dan posisi yang terus bergerak secara random dari robot-robot yang belum direkrut oleh struktur.



Gambar 10. Lokasi Inisiasi Proses *Self-Assembly*, (a) Proses *Self-Assembly* Satu-Per-Satu, (b) Proses *Self-Assembly* Paralel



Gambar 11. Waktu yang Diperlukan untuk Keseluruhan Simulasi dan Proses *Self-Assembly*

4. KESIMPULAN

Dengan membandingkan dua strategi *self-assembly*, yang satu adalah dengan proses bergabungnya robot ke struktur satu-per-satu, dan yang diusulkan adalah strategi penggabungan beberapa robot secara paralel pada suatu struktur, hasil yang diinginkan didapat, yaitu tereduksinya waktu *self-assembly* dengan menerapkan proses penggabungan paralel. Waktu yang tereduksi lebih dari setengah waktu yang diperlukan oleh strategi satu-per-satu dapat memperpanjang masa suplai energi untuk robot, sehingga mereka dapat menyelesaikan tugas yang diberikan dan mengurangi kekuatiran terhentinya operasi *swarm* karena daya yang tidak mencukupi.

Namun, strategi *self-assembly* paralel lebih lanjut perlu dikerjakan untuk mendapat hasil yang minimum, misalnya dengan menggabungkan per-dua robot. Strategi general juga perlu dikembangkan untuk menunjang masalah *scalability* pada *swarm*, dengan kasus jika jumlah robot pada *swarm* dikurangi atau ditambah dengan jumlah berapapun. Metoda pembelajaran evolusi juga akan ditinjau sehingga *swarm* dapat mencari sendiri strategi yang optimal bagi dirinya untuk mereduksi waktu *self-assembly* dan beroperasi sesuai tugas yang diberikan.

DAFTAR RUJUKAN

- Anderson, C., Theraulaz, G. & Deneuborg, J. (2002). Self-assemblages in insect societies. *Insectes Socioux*, 49, 99 - 110.
- Bayindir, L. & Şahin, E. (2007). A Review of Studies in Swarm Robotics. *Turkish Journal of Electrical Engineering*, 15, 115 - 147.
- Beni, G. (2005). From Swarm Intelligence to Swarm Robotics. *Swarm robotics workshop: state-of-the-art survey*, (pp. 1 - 9).
- Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press.
- Bongard, J. (2014). *Why Morphology Matters*. The MIT Press.
- Brambilla, M., Ferrante, E., Birattari, M., & Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7, 1 - 41.
- Bruni, R., Corradini, A., Gadducci, F., Lafuente, A., & Vandin, A. (2015). Modelling and analyzing adaptive self-assembly strategies with Maude. *Science of Computer Programming*, 99, 75 - 94.
- Dorigo, M. et. al. (2013). Swarmanoid: A Novel Concept for the Study of Heterogeneous Robotic Swarms. *IEEE Robotics & Automation Magazine*, (pp. 60 - 71).
- Gilpin, K. & Rus, D. (2010). Modular Robot Systems: From Self-Assembly to Self-Disassembly. *IEEE Robotics & Automation Magazine*, (pp. 38 - 55).
- Groß, R. & Dorigo, M. (2008) Self-Assembly at the Macroscopic Scale. *Proceedings of the IEEE*, (pp. 1490 - 1508).

- Kernbach, S. et. al. (2011). Heterogeneity for Increasing Performance and Reliability of Self-Reconfigurable Multi-Robot Organisms. *IROS11 workshop on Reconfigurable Modular Robotics*.
- Kornienko, S., Kornienko, O., Nagarathinam, A., & Levi, P. (2007). From real robot swarm to evolutionary multi-robot organism. *2007 IEEE Congress on Evolutionary Computation*, (pp. 1483 - 1490).
- Kuyucu, T., Tanev, I., & Shinohara, K. (2013). Hormone-Inspired Behavior Switching for the Control of Collective Robotics Organisms. *Robotics, 2*, 165 - 184.
- Levi, P., Meister, E., & Schlachter, F. (2014). Reconfigurable swarm robots produce self-assembling and self-repairing organisms. *Robotics and Autonomous Systems, 62*, 1371 - 1376.
- Li, H., Wang, T., & Wei, H. (2016). Response Strategy to Environmental Cues for Modular Robots with Self-Assembly from Swarm to Articulated Robots. *Journal of Intelligent Robot Systems, 81*, 359 - 376.
- Martinolli, A. (2001). Book Review: Collective Complexity out of Individual Simplicity. *Artificial Life, 7*, 315 - 319.
- Matthews, N., Valentini, G., Christensen, A., O'Grady, R., Brutschy, A., & Dorigo, M. (2015). Spatially targeted communication in decentralized multirobot systems. *Autonomous Robots, 38*, 439 - 457.
- Meister, E., Nosov, E., & Levi, P. (2013). Automatic Onboard and Online Modeling of Modular and Self-Reconfigurable Robots. *6th IEEE Conference on Robotics, Automation and Mechatronics*, (pp. 91 - 96).
- Mondada, F., Pettinaro, G., Guignard, A., Kwee, I., Floreano, D., Deneuborg, J., Nolfi, S., Gambardella, L., & Dorigo, M. (2004). Swarm-bot: A New Distributed Robotic Concept. *Autonomous Robot, 17*, 193 - 221.
- O'Grady, R., Christensen, A., & Dorigo, M. (2012). *SWARMORPH: Morphogenesis with Self-Assembling Robots*. Springer Berlin Heidelberg.
- Şahin, E. (2005). Swarm Robotics: From Sources of Inspiration to Domains of Application. *Swarm robotics workshop: state-of-the-art survey*, (pp. 10 - 20).
- Syafitri, N. (2018). *Self-Organising Assembly using Swarm Robots*. University of Southampton.
- Vonásek, V., Saska, M., Winkler, L. & Přebušil, L. (2015). High-level motion planning for CPG-driven modular robots. *Robotics and Autonomous Systems, 68*, 745 - 757.
- Winfield, A. (2012). *Robotics: A Very Short Introduction*. Oxford University Press.