

## OPTIMASI SISTEM NAVIGASI ROBOT BENCANA DENGAN ALGORITMA *BUG* DAN JARINGAN SYARAF TIRUAN

Son Kuswadi<sup>1</sup>, Ardelia Natasya G.O.<sup>2</sup>, M. Nasir Tamara<sup>3</sup>, Indra Adji S.<sup>4</sup>

<sup>1</sup>Prodi Teknik Mekatronika, Politeknik Elektronika Negeri Surabaya  
<sup>2</sup>P.T. Sutindo, Surabaya

Email: <sup>1</sup>sonk@pens.ac.id, <sup>2</sup>Ardelia.griselda@gmail.com, <sup>3</sup>nasir\_meka@pens.ac.id, <sup>4</sup>indra@pens.ac.id,

(Naskah masuk: 17 Oktober 2018, diterima untuk diterbitkan: 29 Oktober 2018)

### Abstrak

Agar robot bencana bisa melaksanakan tugas tertentu pada medan yang tak beraturan dan tidak diketahui keadaannya secara dinamis, harus memiliki kemampuan pemetaan. Berdasarkan peta yang telah dibuat, maka robot bisa bergerak sesuai dengan peta tersebut. Makalah ini membahas implementasi pemetaan dan navigasi robot, dengan menggunakan algoritma *bug* untuk membuat lintasan yang dapat menghindari halangan. Lintasan tersebut kemudian dioptimalkan dengan menggunakan jaringan syaraf tiruan untuk memilih lintasan terpendek. Metode yang diusulkan ini kemudian diuji baik menggunakan perangkat lunak dan eksperimen di medan laboratorium.

**Kata kunci:** robot bencana, perencanaan lintasan, navigasi, algoritma *bug*, jaringan syaraf tiruan

## OPTIMISATION OF DISASTER ROBOT NAVIGATION SYSTEMS USING *BUG* ALGORITHMS AND ARTIFICIAL NEURAL NETWORKS

### Abstract

*In order to perform certain task in a cluttered and unknown dynamic field, a disaster robot should have mapping capability. Based on the map, then robot will move accordingly. This paper describes the implementation of mapping and navigation of the robot by using bug algorithm to avoid the obstacle. The paths optimized by using artificial neural networks to select the shortest one. The proposed method then implemented both by simulation and experimental in lab scale field.*

**Keywords:** disaster robot, path planning, navigation, bug algorithms, neural networks

### 1. PENDAHULUAN

Penggunaan robot untuk pencarian dan penyelamatan korban, atau dikenal dengan *USAR* (Urban Search and Rescue) merupakan tantangan menarik bagi peneliti bidang robotika dan interaksi manusia robot *HRI* (Human Robot Interaction) (J. Scholtz dkk, 2004). Robot bencana harus memiliki mobilitas yang fleksibel dan handal. Lingkungan yang dihadapi umumnya sulit, labil dan mudah berubah (karenanya berbahaya) dan sifatnya susah diprediksi. Robot ini harus mampu menjadi anggota tim SAR, mengirimkan informasi di medan bencana terutama keadaan korban, derajat kerusakan lingkungan dan integritas struktur bangunan (Burke J., dkk, 2004). Operator robot bencana akan bekerja dalam waktu yang lama dalam kondisi dengan tekanan fisik dan psikis yang berat. Untungnya, bencana alam bila dibandingkan dengan bencana lain termasuk sangat jarang. Artinya, interaksi manusia-robot harus diantisipasi sedemikian rupa sehingga

meski jarang digunakan, interaksi tersebut dapat berlangsung dengan baik.

Ide penggunaan robot untuk operasi pencarian dan penyelamatan dipicu oleh peristiwa pemboman di kota Oklahoma tahun 1995 dan gempa Kobe, Jepang tahun 1995 (Blitch, 2000) (Lopes, 2001). Semua robot ini memiliki bentuk, ukuran, visi, komunikasi, kecepatan, sensor dan daya yang berbeda-beda. Robot Talon memiliki ukuran sebesar bagasi mobil yang mempunyai kemampuan untuk membawa beban 136 kg dan ini cukup kuat untuk menarik beban seberat 91 kg. Jalur yang biasa dilewati Talon adalah jalur bersalju, lumpur, dan dapat juga beroperasi di air dengan kedalaman sampai 30 meter. Sedangkan robot Solem memiliki ukuran yang lebih kecil dari Tolen yaitu sekitar 40 persen ukuran Tolen dan dari segi kemampuan dan bentuk Solem sama dengan Tolen hanya saja Solem lebih cepat dari Tolen.

Pengembangan robotika untuk operasi penyelamatan bertumpu pada dua kegiatan, yaitu pengembangan platform (Voyles, 2000) (Seron dkk, 2014) dan perangkat lunak (Masuda dkk, 1996). Platform robot yang digunakan untuk operasi SAR bermacam-macam, mulai dari robot yang beroda, berantai dan atau kombinasinya. Robot ular, yang diinspirasi dari sistem biologi, digunakan juga untuk keperluan tersebut (Hirose, 1993). Robot penyelamat kebakaran juga banyak dibahas (Casper dkk, 2000). Tetapi belum dilakukan pengujian menyeluruh pada suatu musibah yang sebenarnya. Sepanjang pengalaman penulis, belum pernah ada robot berkaki yang digunakan untuk operasi SAR, mengingat bahwa medan yang sangat sulit dan tidak dapat diprediksi dengan mudah.

Pengembangan perangkat lunak untuk robot penyelamat meliputi pembuatan perangkat lunak untuk kendali robot, kerjasama multi-robot (*multi robot collaboration*), kendali multi sensor dan alat bantu manusia menggunakan robot. Misalnya, ide perilaku yang diotomatiskan (*automated behavior*) untuk robot yang bisa berubah bentuk (*shape-shifting*) dipaparkan oleh (Blitch dkk, 2000), sedangkan kerjasama multi robot bisa dilihat di (Jennings dkk, 1997), (Geoffrey, Singh, 2012) (Son Kuswadi dkk, 2016a, 2016b). Sistem kendali multisensor untuk robot penyelamat bisa dilihat di (Blitch, Mauer, 2000), sedangkan perangkat lunak yang dikembangkan untuk membantu operator manusia melalui sistem pakar cerdas (*intelligent expert systems*) dan sistem inisiatif campuran (*mixed initiatives systems*) dapat dilihat di (Tadokoro dkk, 1999).

Makalah ini memaparkan bagaimana perencanaan lintasan robot di daerah bencana, yang tidak beraturan dan dinamis. Yang dilakukan pertama adalah pemetaan, dengan menggunakan sensor ultrasonic. Dari peta yang diperoleh, dilakukan perencanaan lintasan optimal menuju sasaran yang diperintahkan oleh operator. Dengan menggunakan algoritma *bug* yang sederhana maka peta tersebut diperbaiki untuk menghindari halangan yang dihadapi robot. Hasil tersebut kemudian dioptimalkan menggunakan jaringan syaraf tiruan (neural network) agar bisa dipilih lintasan terpendek dari alternative lintasan yang ada.

## 2. ROBOT BENCANA YANG TELAH DIKEMBANGKAN

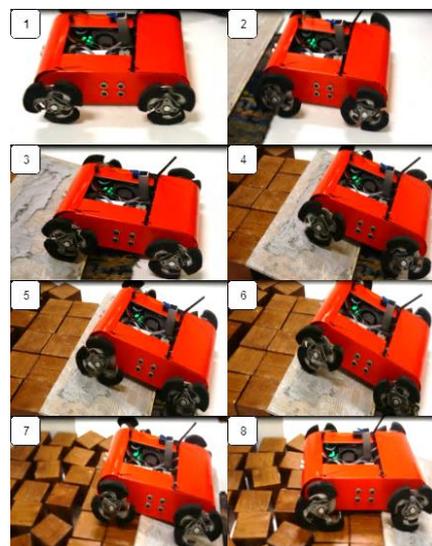
Desain *robot rescue* yang mengadopsi tipe *mobile manipulator*. *Robot rescue* terdiri dari 3 robot yang meliputi 1 *leader* dan 2 *follower*. Ketiga robot ini bertugas untuk mencari dan memberikan pertolongan kepada korban bencana alam. Masing-masing robot dilengkapi dengan mekanisme mobilitas dan mekanisme lengan. Robot *leader* dilengkapi dengan kamera untuk memvisualisasikan kondisi sekitar daerah bencana. Kedua robot *follower*

di lengkapi dengan tabung air dan oksigen untuk memberi pertolongan saat robot menemukan korban.

Kemampuan dasar mekanisme mobilitas robot adalah mampu melakukan mobilitas di daerah bencana. Jika ada halangan (*obstacle*) yang menyebabkan terbalik robot akan tetap mampu berjalan untuk tetap melakukan mobilitas dalam proses pencarian korban bencana alam. Kemampuan dasar mekanisme lengan adalah mampu melipat kedalam bodi dan keluar untuk melaksanakan tugas pertolongan korban (*robot follower*) dan pemantauan kondisi sekitar daerah bencana alam (*robot leader*). Saat robot menemukan korban robot akan memberikan pertolongan pertama dengan mekanisme lengan yang memberikan air dan oksigen. Rancangan salah satu robot dan realisasinya tampak pada gambar 1, yaitu robot pembawa oksigen. Gambar 2 menunjukkan contoh eksperimen pengujian robot di medan bencana yang tidak beraturan.



(a) Rancangan (b) Realisasi  
Gambar 1. Robot follower 2 (pembawa oksigen)

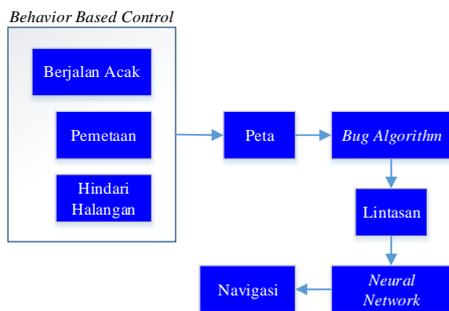


Gambar 2. Contoh pengujian di medan tak beraturan

## 3. SISTEM NAVIGASI ROBOT

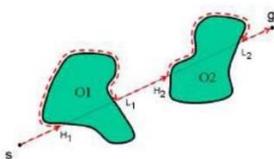
Perancangan sistem secara navigasi ditunjukkan pada gambar 3. Awal robot beroperasi di daerah bencana akan diimplementasikan *behavior based control*. *Behavior based control* yang dirancang memiliki *primitive behavior* berjalan acak dan menghindari halangan. Pemetaan dilakukan ketika *behavior-based control* tersebut diimplementasikan. Dari proses ini dihasilkan peta yang selanjutnya menjadi masukkan perencanaan lintasan.

Perencanaan lintasan ini menggunakan *bug algorithm* yang berfungsi untuk membangkitkan lintasan dari titik awal menuju titik target yang bebas dari tabrakan dengan halangan. Lintasan yang dihasilkan nantinya menggunakan *neural network* sebagai penentu lintasan terpendek ketika menghadapi halangan. Dari proses dan metode yang telah diimplementasikan tersebut akan didapat lintasan yang kemudian menjadi masukan dari navigasi robot. Cara ini relatif lebih praktis, karena hanya dibutuhkan robot di darat saja (ground robot), dibandingkan riset dari (Delmerico, 2017) yang membutuhkan robot terbang (flying robot) untuk observasi medan.



Gambar 3. Proses penentuan lintasan dan navigasi robot

**Algoritma bug** adalah pendekatan navigasi *mobile robot* yang telah dikenal untuk perencanaan jalan lokal dengan lingkungan yang tidak diketahui. Lokasi robot tersedia kapan saja dalam lingkungan di mana robot memiliki kemampuan efisien dan robot dilengkapi dengan sensor untuk mendeteksi dan mengikuti tepi dari halangan (Buniyamin, 2014). Gambar 4 merupakan hasil path planning dengan *Bug Algorithm*. Dimana *s* adalah titik awal, *H1* adalah titik peretemuan dengan obyek *O1*, *L1* adalah titik minimum jarak dimana robot meninggalkan kontur obyek *O1*, *H2* adalah titik pertemuan dengan objek *O2*, *L2* adalah titik jarak minimum dimana robot meninggalkan kontur obyek *O2*, *g* adalah titik target.

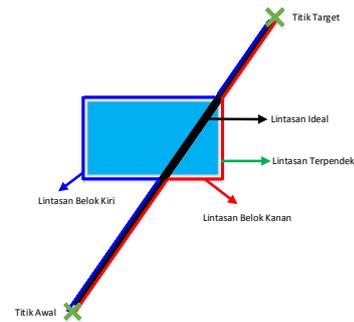


Gambar 4. Path-planning dengan algoritma bug (Vaduva, 2009)

Lintasan yang dihasilkan oleh *Bug Algorithm* masih berupa lintasan yang mencegah terjadinya tabrakan antara robot dengan halangan. *Bug Algorithm* perlu dioptimasi menggunakan Jaringan Syaraf Tiruan (JST) atau *Artificial Neural Network* untuk menghasilkan lintasan yang lebih efisien. R. Glasius (1995), merupakan salah satu pencetus gagasan pemakaian JST dalam optimasi lintasan robot.

Ketika terdeteksi halangan *neural network* akan memutuskan navigasi mana yang menghasilkan

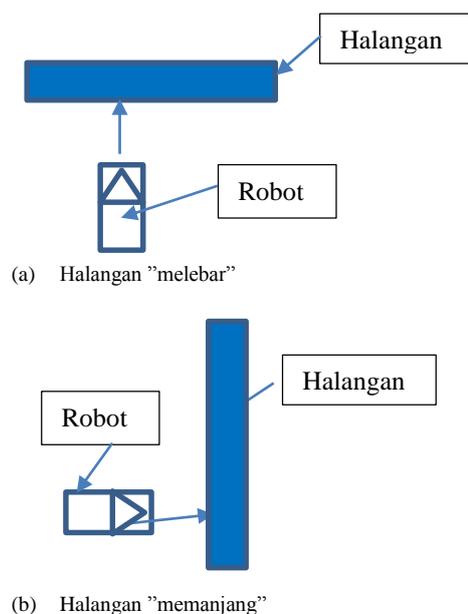
lintasan terdekat. Gambar 5 merupakan ilustrasi optimasi lintasan. Setelah diketahui lintasan ideal, ketika bertemu halangan terdapat 2 kemungkinan pergerakan yaitu ke kanan dan ke kiri. Bila digunakan *Bug Algorithm* saja maka navigasi yang dihasilkan selalu belok kanan. Dengan dikombinasikan dengan *neural network* yang kami usulkan, dapat ditentukan navigasi mana yang menghasilkan lintasan lebih dekat, apakah ke kiri atau ke kanan.



Gambar 5. Ilustrasi optimalisasi lintasan

Strategi untuk meminimalisir lintasan dengan JST adalah menentukan masukan dan keluaran dari JST tersebut. Dirancang masukan sebanyak empat dan keluaran tunggal.

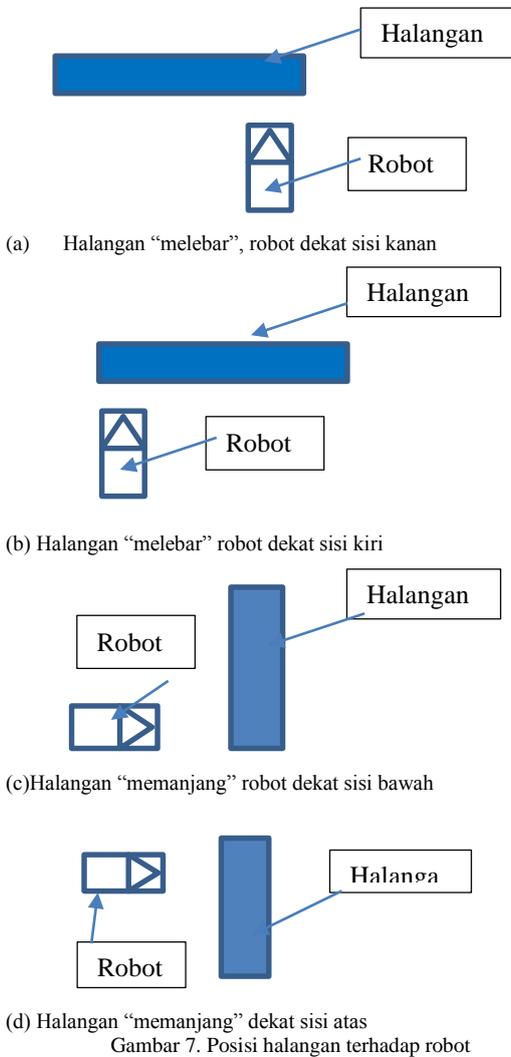
Untuk masukannya, pertama, adalah bila halangan yang dihadapi robot, berupa kotak biru, yang secara umum akan dilihat oleh robot halangannya berupa "melebar" (diberi nilai "-1") atau "memanjang" (diberi nilai "1"), seperti diilustrasikan pada gambar 6.



Gambar 6. Tipe halangan

Masukan kedua, bila halangan yang dihadapi robot, untuk halangan "melebar", apakah robot lebih

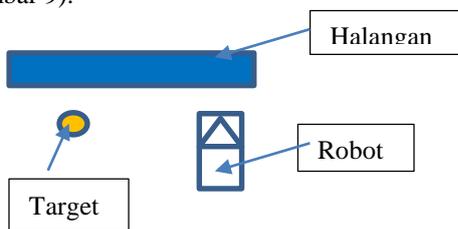
dekat dengan sisi kiri (diberi nilai "1"), atau kanan (diberi nilai "-1"). Bila robot dengan halangan "memanjang", apakah robot lebih dekat dengan daerah atas (diberi nilai "-1") atau bawah (diberi nilai "1"). Lihat Gambar 7.



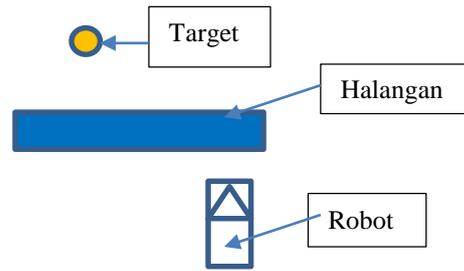
Gambar 7. Posisi halangan terhadap robot

Masukan ke tiga, akan dibedakan apakah target terletak di sisi yang sama dengan robot (diberi nilai "1"), ataukah letaknya berseberangan (diberi nilai "-1"). Lihat gambar 8.

Masukan ke empat, akan dibedakan apakah target lebih dekat dari sisi kiri (diberi nilai "-1") atau kanan (diberi nilai "1") (untuk halangan "melebar") atau lebih dekat atas (diberi nilai "1") atau bawah (diberi nilai "-1") untuk halangan "memanjang" (lihat gambar 9).

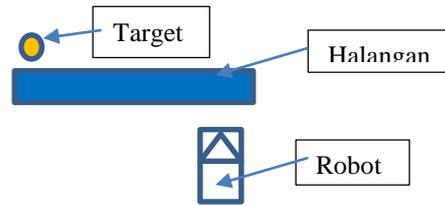


(a) Target berada pada sisi yang sama dengan robot

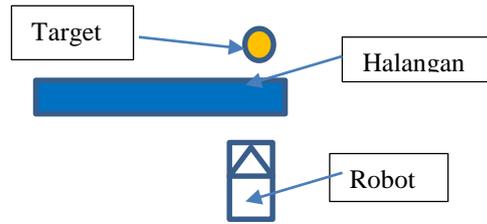


(b) Target berada pada sisi berseberangan dengan robot

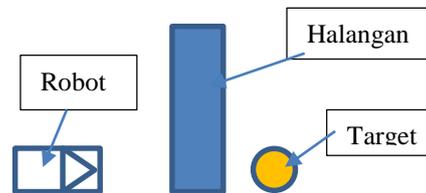
Gambar 8. Letak target apakah berada pada sisi yang sama dengan robot atau berseberangan



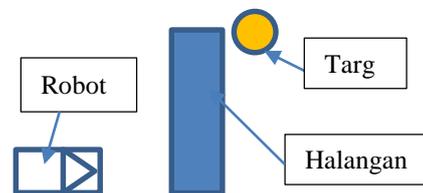
(a) Halangan "melebar" target dekat sebelah kiri



(b) Halangan "melebar" target dekat sebelah kanan



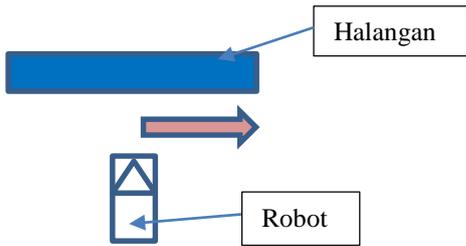
(c) Halangan "memanjang" target berada di bawah



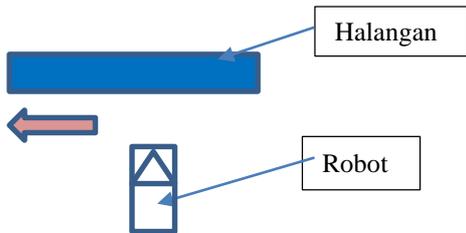
(d) Halangan "memanjang" target berada di atas

Gambar 9. Letak target terhadap halangan

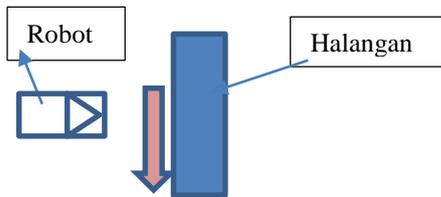
Sedangkan untuk keluarannya, bila halangannya "melebar", bila belok kanan diberi nilai "-1" sedangkan bila belok kiri diberi nilai "1". Bila halangannya memanjang, maka bila gerakannya ke bawah diberi nilai "-1" sedangkan untuk gerakan ke atas diberi nilai "1" (lihat Gambar 10).



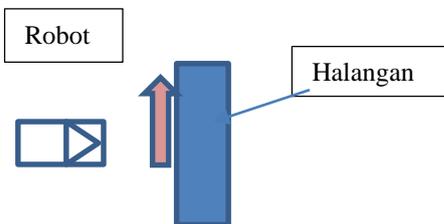
(a) Halangan “melebar” robot bergerak ke kanan



(b) Halangan “melebar” robot bergerak ke kiri



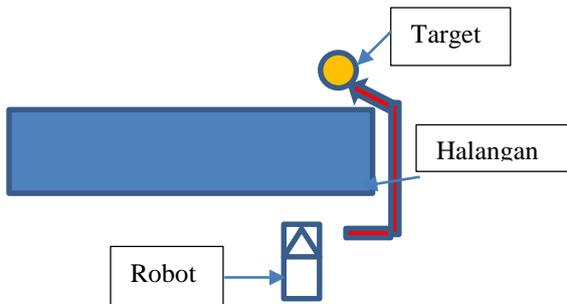
(c) Halangan “memanjang” robot bergerak ke bawah



(d) Halangan “memanjang” robot bergerak ke atas

Gambar 10. Definisi keluaran untuk gerakan robot

Sebagai ilustrasi, bila robot menghadapi halangan dan target seperti Gambar 11, maka perhitungan masukan keluaran (pembelajaran “supervised”) adalah:



Gambar 11. Ilustrasi situasi lintasan terpendek dari robot untuk mencapai target

Masukan: (1) Halangan “melebar” → “-1” (2) Robot berada di sisi kanan halangan → “-1” (3) Letak target

terhalang oleh halangan dengan robot → “-1” (4) Target terletak di sisi kanan, → “1”

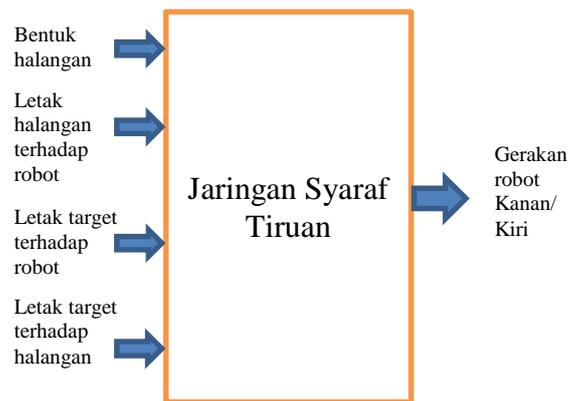
Keluaran: Untuk mencapai lintasan terpendek, Robot bergerak ke kanan → “-1”

Dengan menggunakan prinsip tersebut maka dapat diturunkan tabel masukan dan keluaran JST untuk pembelajaran “supervised” seperti tampak pada tabel 1.

Tabel 1. Tabel masukan dan keluaran JST

Masukan 1	Masukan 2	Masukan 3	Masukan 4	Keluaran
-1	-1	-1	-1	-1
-1	-1	-1	1	-1
-1	1	1	-1	-1
-1	1	1	1	1
1	-1	-1	-1	1
1	-1	1	-1	1
1	1	1	-1	1
1	1	1	1	-1

Sedangkan struktur masukan dan keluaran JST tampak pada Gambar 12.



Gambar 12. Struktur JST untuk pengendalian lintasan robot

#### 4. SIMULASI DAN EKSPERIMEN

Simulasi dilakukan dengan asumsi bahwa medan yang dihadapi robot adalah tampak pada gambar 13. Robot diuji kemampuannya untuk mampu mencapai target, meski pun dengan halangan, menggunakan (a) algoritma *bug* dan (b) kombinasi algoritma *bug* dan dioptimalkan menggunakan JST. Perbandingan dari kedua skenario tersebut tampak pada gambar 14.

Tampak bahwa algoritma kombinasi *bug* dan JST bisa menghasilkan lintasan lebih pendek dibandingkan hanya menggunakan algoritma *bug* saja, atau paling tidak sama lintasannya.

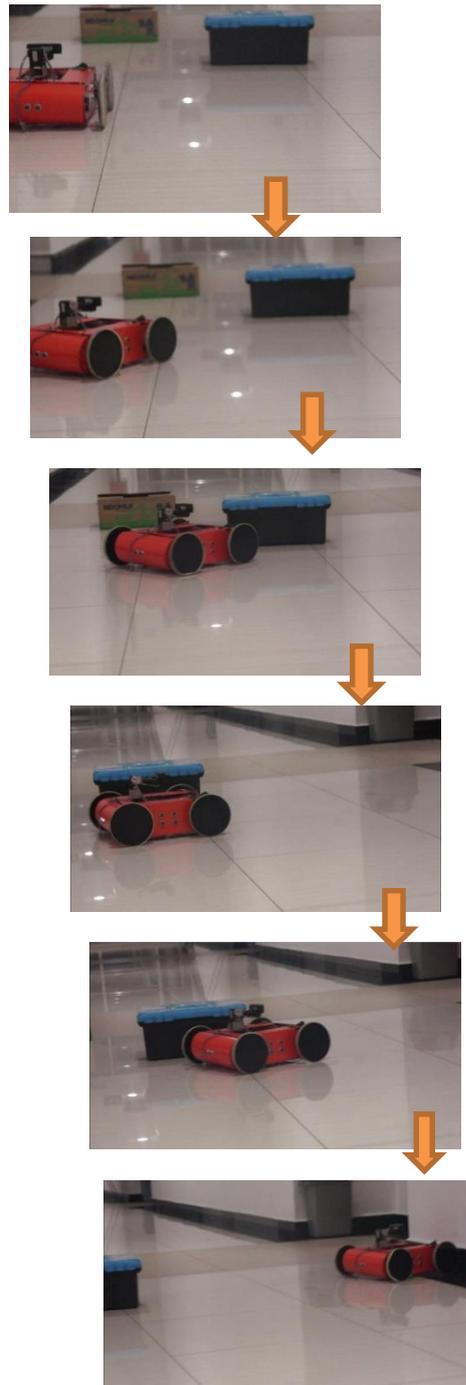
Eksperimen dilakukan pada suatu daerah seperti pada gambar 15 (a), di mana robot (berwarna merah) dihadapkan pada situasi adanya dua halangan di depannya. gambar 15 (b) menunjukkan simulasi lintasan, dengan menggunakan algoritma *bug* dan JST.



Gambar 13. Daerah medan uji, robot dan target posisi yang dikehendaki

	Hasil Bug Algorithm	Hasil kombinasi Bug Algorithm - JST
1		
2		
3		
4		

Gambar 14. Perbandingan lintasan robot menggunakan algoritma bug dan kombinasi algoritma bug dan optimasi menggunakan JST.



Gambar 16. Snapshot hasil eksperimen lintasan robot



(a) Ruang eksperimen.

(b) Simulasi lintasan

Gambar 15. Situasi eksperimen vs simulasi

Dari gambar 16 di atas nampak bahwa lintasan yang dilakukan pada eksperimen dengan robot sebenarnya sudah sesuai dengan hasil simulasi. Meski pun halangan tidaklah berbentuk persegi panjang, tetapi algoritma tetap bisa diimplementasikan dan menghasilkan lintasan yang cukup bisa diterima, dalam arti meski lintasannya tidak terlalu persisi tetapi masih bisa sampai kepada titik tujuan yang dikehendaki.

Eksperimen dengan medan yang yang tak beraturan seperti layaknya medan bencana, sedang dalam proses untuk dilakukan. Meski pun demikian, dengan asumsi bahwa halangan robot dalam bentuk

persegi panjang tidak mengurangi validitas hasil simulasi ini, karena semua bentuk halangan yang ada pada daerah bencana dapat dibawa ke bentuk persegi panjang (melalui pendekatan).

## 5. KESIMPULAN

Telah diusulkan metode navigasi perencanaan lintasan robot dengan memadukan algoritma *bug* dan JST (Jaringan Syaraf Tiruan). Algoritma *bug* memiliki sifat cepat dan sederhana, sehingga mampu memilih jalur yang bebas dari halangan dalam waktu cepat. JST dapat mengoptimalkan jalur yang dipilih dengan memilih jalur terpendek. Masukan JST ada empat, yaitu bentuk halangan, letak halangan terhadap robot, letak target terhadap robot dan letak halangan terhadap robot.

Untuk validasi metode yang diusulkan, telah dilakukan simulasi perencanaan lintasan robot dengan memilih medan dengan lintasan yang beragam, meletakkan posisi robot dan tujuan yang berbeda-beda. Hasil simulasi telah mengkonfirmasi bahwa metode yang diusulkan lebih baik (atau paling tidak sama) dengan algoritma *bug*. Eksperimen kemudian dilakukan pada medan yang sebenarnya, dan nampak bahwa algoritma ini memiliki potensi untuk bisa diimplemmentasikan dengan baik, mengingat aspek cepat dan sederhananya.

## UCAPAN TERIMA KASIH

Penelitian ini mendapat dukungan dana dari Direktorat Riset dan Pengabdian Kepada Masyarakat, Direktorat Jenderal Penguatan Riset dan Pengembangan, Kemristekdikti, dengan Nomer Kontrak 10/PL14/PG.1/SP2P/2018 Tanggal 26 Maret 2018.

## DAFTAR PUSTAKA

- BLITCH J.G. & R. MAUER, 2000. KNOBSAR: A Knowledge Based System Prototype for Robot Assisted Urban Search and Rescue, Simulation, Vol. 66, pp. 375-391
- BLITCH J., N. SIDKI, & T. DURKIN, 2000. Tactical Mobile Robots for Urban Search and Rescue, Proceedings of the SPIE - The International Society for Optical Engineering, vol. 4024, pp. 201-211
- BURKE J., R. MURPHY, M. COOVERT, D. RIDDLE, 2004. Moonlight in Miami: An Ethnographic Study of Human-Robot Interaction in USAR, Human-Computer Interaction, Special Issue on Human-Robot Interaction, vol. 19, pp. 85-116, 2004.
- BUNIYAMIN, N.W., A. J. WAN NGAH, Z. MOHAMAD, 2014. PointsBug Versus TangentBug Algorithm, A Performance Comparison In Unknown Static Environment, 2014 IEEE on Sensors Applications Symposium (SAS), pp. 278 - 282, February
- CASPER J., M. MICIRE, R. MURPHY, 2000. Issues in Intelligent Robots for Search and Rescue, SPIE Ground Vehicle Technology II, Vol. 4, pp. 41-46
- DELMERICO J., E. MUEGLER, J. NITSCH, D. SCARAMUZZA, 2017. Active autonomous aerial exploration for ground robot path planning, IEEE Robotics and Automation Letter, Vol. 2, No. 2, April.
- GEOFFREY, A.H., & S. SINGH, 2012. Multirobot coordination with periodic connectivity: Theory and experiment, IEEE Trans on Robotics, Vol 28, No.4, August
- GLASIUS R., A. KOMODA, S.C.A.M. GIELEN, 1995. Neural Network Dynamics for Path Planning and Obstacle Avoidance, Neural Networks, Vol. 8
- HIROSE S., 1993. Biologically Inspired Robots: Snake-like Locomotors and Manipulators, Oxford University Press, Oxford.
- JENNINGS, J.S., G. WHELAN, W.F. EVANS, 1997. Cooperative Search and Rescue with a Team of Mobile Robots, 8th Int. Conf. on Advanced Robotics, pp. 193-200
- LOPES L., J. CONNELL, P. DARIO, R. MURPHY, P. BONASSO, B. NEBEL, and R. BROOKS, 2001. Sentience in Robots: Applications and Challenges," IEEE Intelligent Systems, vol. 16, pp. 66-69.
- MASUDA, R., T. OINUMA and A. MURAMATSU, 1996. "Multi-Sensor Control System for Rescue Robot," in 1996 IEEE/SICE/RSJ Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems, pp. 381-387
- SCHOLTZ J., J. YOUNG, J.L. DRURY, H.A. YANCO, 2004. Evaluation of human-robot interaction awareness in search and rescue Proceeding International Conference Robotics and Automation (ICRA 2014), 26 April – 1 May 2004
- SERON J., J.L. MARTINEZ, A. MANDOW, A.J. REINA, J. MORALES, A.J. GARCIA-

CEREZO, 2014. Automation of the arm-aided climbing maneuver for tracked mobile manipulator, *IEEE Trans. On Industrial Electronics*, Vol. 61, No 7, July, pp. 3638-3647

SON KUSWADI, RIYANTO SIGIT, INDRA ADJI S., M. NASYIR TAMARA, DZIKRI ADITYA SAHANAS, GALANG ILMAN ISLAMI, 2016a). Adaptive Morphology-based design of multi-locomotion flying and crawling robot “PENS-FlyCrawl”, *Proceeding IEEE Knowledge Creation and Intelligent Computing, Manado 18-20 November 2016, Manado, in press*

SON KUSWADI, RIYANTO SIGIT, INDRA ADJI S., M. NASYIR TAMARA, RAKA CHANDRADITYA, RONA ROBERTY NAILUS SHOFI, 2016b. Optimal Cone of Relative Position Acquisition Module of Multi Mobile Robot, *Proceeding IEEE Knowledge Creation and Intelligent Computing, Manado 18-20 November 2016, Manado.*

TADOKORO S., R. VERHOEVEN, M. MILLER, T. TAKAMORI, 1999. A Portable Parallel Manipulator for Search and Rescue at Large-Scale Urban Earthquakes and an Identification Algorithm for the Installation in Unstructured Environments, *Proc. 1999 IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems*, pp. 1222-1227

VOYLES, RM, 2000. TerminatorBot: A Robot with Dual-Use Arms for Manipulation and Locomotion, *Proc. Of the 2000 Int. Conf. on Robotics and Automation*, pp. 61-66