

METODE LOKALISASI ROBOT OTONOM DENGAN MENGGUNAKAN ADOPTI ALGORITMA *HEURISTIC SEARCHING* DAN *PRUNING* UNTUK PEMBANGUNAN PETA PADA KASUS *SEARCH-AND-SAFE*

W. Jatmiko¹, M. S. Alvissalim², A. Febrian³, dan Dhiemas R.Y.S⁴

Fakultas Ilmu Komputer, Universitas Indonesia, Depok, Indonesia
wisnuj@cs.ui.ac.id

Abstrak

Permasalahan *search-and-safe* merupakan salah satu contoh robot otonom dapat disimulasikan untuk menggantikan pekerjaan manusia di lingkungan berbahaya, misalnya pada kegiatan evakuasi manusia dari ruang tertutup yang terbakar. Dalam contoh ini, robot otonom harus dapat menemukan objek manusia untuk diselamatkan, serta objek api untuk dipadamkan. Lebih jauh lagi, untuk dapat menyelesaikan permasalahan seperti ini dengan baik, robot otonom harus dapat mengetahui keberadaannya, bukan hanya posisi dalam sistem koordinat global saja tetapi juga posisi relatif terhadap posisi tujuan dan keadaan lingkungan itu sendiri. Permasalahan ini kemudian dikenal juga sebagai lokalisasi yang menjadi bagian penting dari proses navigasi pada robot otonom. Salah satu metode yang dapat digunakan untuk menyelesaikan permasalahan lokalisasi adalah dengan menggunakan representasi internal peta lingkungan kerja dalam pengetahuan robot otonom. Pada kondisi ketika tidak tersedia informasi mengenai konfigurasi lingkungan, atau informasi yang tersedia sifatnya terbatas, robot harus dapat membangun sendiri representasi petanya dengan dibantu oleh komponen sensor yang dimilikinya. Pada paper ini kemudian dibahas salah satu metode yang dapat diterapkan dalam proses pembangunan peta seperti yang dijelaskan, yaitu melalui adopsi algoritma *heuristic searching* dan *pruning* yang sudah dikenal pada bidang kecerdasan buatan. Selain itu juga akan dijabarkan desain robot otonom yang digunakan, serta konfigurasi lingkungan yang digunakan pada studi kasus *search-and-safe* ini. Diharapkan nantinya hasil yang diperoleh dari penelitian ini dapat diterapkan untuk skala yang lebih besar.

Kata kunci: lokalisasi robot, robot otonom bergerak, pembangunan peta, algoritma *heuristic searching*.

1. Pendahuluan

Navigasi adalah salah satu permasalahan penting yang harus diselesaikan dalam pengembangan teknologi robot otonom bergerak, agar dapat mendukung mobilitasnya. Masalah ini menyangkut beberapa komponen penting di dalamnya, dimulai dari masalah persepsi, yaitu metode atau cara agar suatu robot otonom dapat memperoleh data tertentu dari lingkungan di sekitarnya, kemudian untuk dapat menginterpretasikan data tersebut menjadi informasi yang berguna bagi proses selanjutnya. Berkaitan dengan masalah persepsi sangat erat kaitannya dengan masalah sensor dan rekognisi pada robot otonom tersebut.

Masalah berikutnya adalah lokalisasi, yaitu metode atau cara agar robot otonom dapat mengetahui posisi atau keberadaannya dalam suatu lingkungan tempat robot tersebut harus menyelesaikan misi atau mencapai tujuan yang dibebankan kepadanya. Masalah lokalisasi ini biasanya erat dengan penentuan posisi robot otonom dalam suatu sistem koordinat absolut

yang bersifat global (biasanya mengacu pada kedudukan benda pada sistem koordinat di bumi). Proses penentuan kedudukan ini bisa berasal dari pengetahuan yang diberikan manusia kepada robot otonom, berdasarkan informasi yang diperoleh melalui komponen sensor yang digunakan robot otonom, ataupun dari kedua metode tersebut.

Sedangkan dua masalah terakhir yang terkait dengan pengembangan modul navigasi pada robot otonom adalah masalah kognisi dan kontrol gerak, yang berkaitan dengan metode atau cara agar robot otonom dapat bergerak untuk mencapai suatu posisi tujuan dalam rangka menyelesaikan misinya. Kedua hal ini menyangkut pengembangan algoritma penyelesaian masalah secara komputasional untuk menentukan langkah yang harus ditempuh robot otonom dari posisi awalnya menuju posisi tujuan akhir, hingga penentuan bagaimana robot berinteraksi dan mengendalikan komponen motornya dalam rangka mencapai posisi tujuannya tersebut [1].

Pada kenyataannya, dalam beberapa waktu terakhir ini, masalah lokalisasi telah menyita banyak perhatian banyak peneliti dalam berbagai

riset di teknologi bidang robotika. Hal ini disebabkan karena masalah lokalisasi seringkali menjadi suatu masalah rumit karena keterbatasan yang dimiliki sensor robot dalam mendapatkan informasi dari lingkungannya. Pada suatu lingkungan kerja yang sama sekalipun, pada waktu dan kondisi yang berbeda, suatu robot otonom dapat memperoleh informasi yang berbeda, sebagai akibat adanya *noise* pada komponen sensor robot tersebut. Lebih jauh lagi, pada saat bergerak robot otonom juga dapat mengalami *error* sebagai akibat dari *noise* pada komponen motor. Kedua hal ini pada akhirnya menyebabkan kedudukan robot otonom tidak dapat diperhitungkan secara benar. Padahal pada tahap selanjutnya kognisi dan kontrol gerak, perbedaan posisi robot dalam perhitungan dan posisi robot yang sebenarnya dapat memberikan bias pada proses pencapaian posisi tujuan. Hal ini pada akhirnya dapat menghalangi robot untuk menemukan hasil akhir yang diharapkan. Hal inilah yang kemudian menjadi alasan pentingnya lokalisasi untuk dapat ditangani dengan baik pada pengembangan robot otonom bergerak.

Dalam perkembangannya yang telah banyak menyita perhatian publik ini, berbagai penelitian yang dilakukan telah mendapatkan berbagai alternatif solusi yang dapat diterapkan dalam upaya untuk menangani proses lokalisasi pada robot otonom bergerak [2-6]. Salah satunya adalah dengan membentuk suatu representasi internal dari lingkungan kerja tempat robot otonom berada dalam suatu basis pengetahuan berupa peta atau bentuk topologi dari lingkungan tersebut, atau dikenal dengan teknik lokalisasi atau navigasi berbasis peta. Dengan adanya peta yang menggambarkan kondisi lingkungan tempat robot otonom bekerja, robot kemudian dapat menentukan kedudukannya, tidak hanya posisi dalam suatu koordinat tertentu saja tetapi juga posisi relatif terhadap posisi tujuan yang akan dicapainya. Dengan demikian, hal ini memberikan keunggulan tersendiri bagi permasalahan kognisi dan kontrol gerak yang harus diselesaikan berikutnya. Robot otonom secara eksplisit menentukan posisinya berdasarkan representasi peta yang dimilikinya, kemudian memperbaharui posisinya pada peta setiap kali robot bergerak.

Selain itu teknik seperti ini juga memiliki keunggulan lainnya, yaitu representasi peta yang dimiliki robot otonom juga dapat digunakan sebagai sarana berkomunikasi dengan operator manusia. Posisi pergerakan robot otonom dapat dimonitor oleh operator manusia melalui representasi peta; serta pada saat memasuki lingkungan kerja baru, robot cukup diberikan representasi peta yang baru untuk mencapai posisi tujuan baru pada lingkungan tersebut. Pada saat informasi mengenai kondisi lingkungan tidak diketahui sebelumnya atau tidak lengkap, robot otonom dapat membangun peta

dengan menggunakan sensornya.

2. Motivasi

Berdasarkan penjabaran di atas disebutkan manfaat dari representasi peta secara internal pada robot otonom bergerak untuk menyelesaikan masalah lokalisasi. Representasi peta sendiri terbagi atas beberapa jenis [2], dengan informasi yang terdapat di dalamnya bisa jadi sudah diberitahukan sebelumnya, sehingga proses kognisi dan kontrol gerak dapat lebih mudah dilakukan. Akan tetapi, pada kondisi tertentu ketika informasi dari lingkungan tidak diketahui secara lengkap atau bahkan tidak tersedia sama sekali, menimbulkan tantangan tersendiri dalam proses lokalisasi. Hal ini disebabkan karena robot otonom harus secara inisiatif membangun representasi peta lingkungan kerjanya sendiri, kemudian menentukan posisinya dalam peta yang dibuat.

Pada *paper* ini akan dibahas salah satu contoh studi kasus yang melakukan penyelesaian masalah lokalisasi dengan menggunakan peta secara eksplisit untuk menentukan posisi robot otonom, untuk kemudian dimanfaatkan dalam rangka mencapai posisi tujuan. Contoh kasus yang dimaksud adalah permasalahan *search-and-safe* pada suatu lingkungan yang tertutup dan terbatas. Pada contoh permasalahan ini, robot tidak hanya diminta untuk mencapai posisi tertentu sebagai tujuan akhirnya, akan tetapi untuk menemukan suatu objek tertentu dan kemudian membawa objek tersebut kembali ke tujuan asal. Dari penjelasan ini, dapat dilihat bahwa metode lokalisasi yang dituntut tidak hanya untuk mengenali posisi robot pada setiap waktu beserta posisi tujuan akhirnya tetapi juga posisi awal robot otonom memulai pekerjaannya. Penggunaan objek untuk ‘dicari dan diselamatkan’ memberikan alasan yang lebih nyata, agar robot harus kembali ke posisi awalnya nanti, selain juga untuk memberikan fitur lain pada robot otonom.

Fitur yang dimaksud adalah robot harus dapat mengenali suatu objek berdasarkan karakteristik yang dimilikinya, untuk menentukan objek mana yang harus dibawa kembali ke posisi awal, objek mana yang hanya merupakan rintangan dalam perjalanan robot mencapai tujuan akhirnya, dan sebagainya. Selain itu, kegiatan pengembangan robot otonom kemudian juga harus memperhatikan aspek desain yang lebih baik, karena robot tidak hanya dituntut untuk dapat berjalan dan menghindari rintangan, tapi juga untuk berinteraksi dengan objek yang harus diselamatkannya. Untuk keperluan pemeriksaan keberhasilan proses robot otonom dalam mengenali objek yang akan diselamatkan, pada lingkungan kemudian juga ditambahkan objek lain yang alih-alih diselamatkan, harus dilakukan tindakan lain yang berbeda terhadapnya. Misalnya keberadaan api

yang harus dipadamkan. Contoh kasus seperti kemudian dapat menjadi simulasi dalam kejadian yang dapat ditemui dalam kehidupan nyata, yaitu proses evakuasi manusia yang terperangkap pada suatu bangunan yang sedang terbakar. Penggunaan robot otonom bergerak di sini tentunya akan membawa keunggulan, karena bangunan yang sedang terbakar merupakan lingkungan yang berbahaya untuk dimasuki petugas penyelamat sedangkan proses evakuasi tetap harus dilakukan. Selain itu, api juga dapat dengan mudah dibedakan dari objek fisik melalui pendeteksian kepanasan, sehingga robot otonom dapat memilih tindakan yang akan dilakukan pada saat menemukan objek ataupun api.

Adapun metode lokalisasi yang diterapkan pada penelitian ini menggunakan representasi peta yang dibangun oleh robot otonom, artinya belum tersedia informasi secara pasti mengenai keadaan lingkungan. Untuk penyederhanaan masalah, diberikan asumsi bahwa lingkungan tempat robot otonom berada terbatas pada beberapa kemungkinan konfigurasi saja. Semua kemungkinan konfigurasi ini diberitahukan kepada robot, sehingga robot akan mempunyai informasi parsial terhadap lingkungan kerjanya. Robot selanjutnya akan melakukan pencarian terhadap semua kemungkinan konfigurasi ini untuk menentukan representasi peta seperti apakah yang akan digunakannya untuk menentukan posisi awal, posisi saat ini, serta posisi akhir tempat objek berada. Perlu diperhatikan bahwa walaupun posisi objek yang akan diselamatkan diketahui, robot tetap harus mengenali keberadaan objek pada ruangan tersebut, karena robot harus berinteraksi dengan objek dengan cara mengambil dan lalu membawanya.

Untuk alasan penyederhanaan masalah juga, konfigurasi lingkungan dibuat berupa *maze* atau labirin untuk memudahkan representasi peta pada robot otonom dan dapat mengakomodasi keberadaan rintangan dengan baik. Rintangan secara seragam berupa dinding atau *jamming* yang terdapat di lantai. Selain itu, kondisi lingkungan yang seperti ini juga untuk melihat seberapa besar keberhasilan metode lokalisasi kan membawa pengaruh bagi keberhasilan proses kognisi dan kontrol gerak pada navigasi robot otonom nantinya.

Berdasarkan garis besar permasalahan yang dijelaskan di atas, maka kegiatan penelitian ini secara garis besar dibagi menjadi dua. Tahapan pertama adalah desain robot dan perangkat yang akan digunakan. Kegiatan ini juga mencakup pemilihan komponen sensor yang dapat mendukung navigasi robot, hingga perakitannya. Tahap kedua adalah perancangan algoritma penyelesaian masalah yang menyangkut dengan pembangunan peta internal pada robot otonom berdasarkan semua kemungkinan konfigurasi lingkungan yang ada. Hal ini juga mencakup perencanaan jalur yang akan

ditempuh dari posisi awal menuju akhir, mengambil objek, dan kemudian kembali lagi ke posisi awal. Parameter keberhasilan penelitian kemudian dilihat dari kemampuan robot dalam menemukan dan menyelamatkan objek yang dicari.

3. Disain Robot Otonom

3.1. Desain Perangkat Keras

Tahapan yang pertama kali dilakukan pada penelitian ini adalah perancangan desain robot otonom yang dapat digunakan untuk menyelesaikan masalah seperti yang dijabarkan di atas. Berikut adalah beberapa komponen perangkat keras yang terdapat pada robot otonom.

1. Komponen badan (*body*) robot yang dirakit dengan menggunakan Platform Traxter II Trossen Robotics. Komponen ini memiliki dua roda di kiri dan dua roda di kanan yang masing-masingnya terhubung oleh *belt*. Selain itu juga ditambahkan dua tingkatan atau *layer* pada badan robot dengan menggunakan bahan *acrylic*.
2. Komponen pemadam-api (*fire extinguisher*) yang dipasang pada bagian badan robot. Komponen ini terdiri dari *propeller* yang digerakkan oleh servo-motor sebagai penggerak utama. Selain itu, komponen ini juga dilengkapi oleh sensor TPA81 yang berfungsi untuk mendeteksi temperatur objek dan Uvtron Hamamatsu yang berfungsi untuk mendeteksi radiasi ultraviolet dari sumber api. Kedua sensor ini diterapkan dalam rangka menemukan sumber api. Komponen pemadam-api ini dapat bergerak sejauh 135 derajat di bidang horizontal untuk menjangkau posisi sumber api.
3. Komponen jari-pencengkram (*gripper*) untuk mengambil objek yang dicari dan membawanya bersama robot otonom. Komponen ini juga dipasang pada badan robot otonom dan digerakkan oleh servo-motor, termasuk untuk melipat jari-pencengkram saat komponen ini tidak digunakan atau untuk memindahkan objek ke bagian atas robot. Hal ini dilakukan agar dimensi panjang dan lebar robot tidak bertambah saat objek dibawa.
4. Komponen sensor jarak untuk mendeteksi keberadaan objek lain di sekitar robot otonom dengan rentang 90 derajat. Sensor yang digunakan adalah sensor sonar SRF08 Devantech dan sensor inframerah GP2D120 Sharp. Sensor sonar yang digunakan sebanyak delapan unit yang dipasang mengelilingi badan robot, bersesuaian dengan posisi arah mata angin, agar robot dapat mengenali objek di sekelilingnya.
5. Komponen *limit-switch* yang dipasang untuk mendeteksi terjadinya tumbukan. Pada saat terjadi tumbukan, akan menyebabkan

komponen ini tertekan oleh objek yang menabraknya, sehingga robot otonom dapat melakukan gerakan yang menjauhi tumbukan. Penggunaan sensor ini dimaksudkan agar robot dapat tetap melanjutkan perjalanan pada saat terjadi tumbukan dengan halangan sebagai akibat kegagalan sensor jarak mendeteksi keberadaan objek.

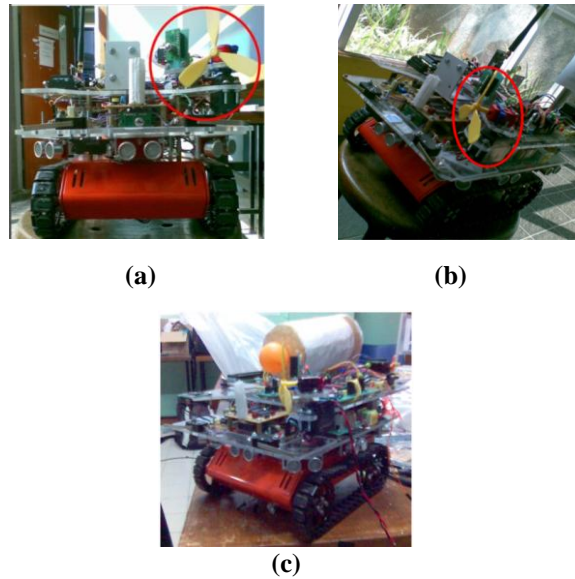
6. Komponen sensor cahaya atau *fototransistor* yang digunakan untuk mendeteksi perbedaan karakteristik warna lantai yang dilalui oleh robot. Hal ini nantinya dapat dimanfaatkan dengan cara membedakan warna lantai pada posisi awal, sehingga robot dapat melakukan verifikasi bahwa misi telah berhasil diselesaikan pada saat mencapai posisi ini, tidak hanya berdasarkan posisi yang diperoleh berdasarkan representasi peta.
7. Komponen sensor arah atau kompas yang berguna untuk mengetahui arah orientasi robot di dalam ruangan. Komponen sensor arah yang digunakan adalah kompas digital CMPS03 Devantech.
8. Komponen aktuator yang berfungsi untuk output pergerakan robot otonom. Adapun beberapa motor yang digunakan adalah DC Motor *Traxter II*, sebagai motor penggerak utama; *Standard Servo Paralaks*, sebagai motor penggerak komponen pemadam-api dan motor penggerak komponen jari-pencengkram; serta DC Motor Audley, sebagai motor penggerak *Propeller*. Selain itu, robot otonom juga ditambahkan dengan komponen LCD yang berfungsi sebagai *debugger* untuk menampilkan hasil informasi yang diperoleh oleh sensor, misalnya jarak dari objek, dan sebagainya.

Pada Gambar 1 terdapat tampilan spesifikasi robot otonom bergerak yang digunakan pada penelitian ini, sesuai dengan penjabaran komponen-komponen yang telah disebutkan. Gambar 1.a dan 1.b merupakan desain robot secara umum dengan komponen pemadam-api. Sedangkan pada Gambar 1.c robot otonom membawa objek dengan meletakkannya pada bagian atas badan dengan menggunakan komponen jari-pencengkram.

1.2. Kebutuhan Perangkat Lunak

Kegiatan yang dilakukan pada tahap selanjutnya adalah implementasi kode program dengan menggunakan perangkat lunak yang dibutuhkan dalam rangka mendapat sistem kontrol pada robot otonom yang dikembangkan. Pada penelitian ini, kode program dibuat dengan menggunakan CodeVisionAvrC Compiler, dengan bahasa pemrograman C sebagai basisnya. Beberapa fitur yang ditawarkan oleh perangkat lunak ini adalah adanya fasilitas untuk *download* kode program langsung ke dalam *microcontroller*, sehingga akan memudahkan kegiatan pengembangan. Selain itu

juga tersedia fasilitas Terminal yang berfungsi untuk memberi dan menerima perintah pada *microcontroller*. Adapun algoritma penyelesaian masalah yang diterapkan pada tahapan ini akan dijelaskan pada Bagian 4.



Gambar 1. Tampilan robot otonom yang digunakan pada penelitian. (a) tampak depan, (b) tampak samping, (c) pada saat membawa objek

1.3. Desain Konfigurasi Ruang

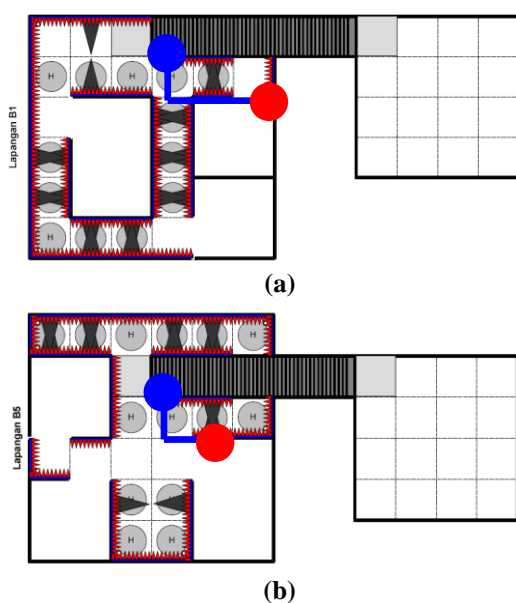
Seperti yang dijelaskan sebelumnya, walaupun pada penelitian ini kondisi lingkungan tempat robot otonom diminta untuk menemukan objek tidak diketahui secara jelas tetapi dalam rangka memudahkan permasalahan, maka digunakan asumsi bahwa konfigurasi lingkungan kerja hanya terbatas pada beberapa kemungkinan yang terhingga. Robot kemudian diberitahukan mengenai semua kemungkinan ini. Kemudian dengan kemampuannya secara mandiri robot dituntut untuk dapat menentukan konfigurasi peta yang sesuai dengan kondisi yang ditemui sensornya. Dengan demikian, dapat juga dikatakan bahwa sesungguhnya informasi mengenai lingkungan tersedia tetapi bersifat terbatas.

Dalam penelitian ini kemudian digunakan ruangan yang berbentuk labirin atau *maze* yang dibagi atas grid atau kotak-kotak tertentu. Keseluruhan ruangan ini selanjutnya terbagi menjadi dua bagian dengan robot otonom berada dari suatu bagian ruangan sedangkan objek yang akan diselamatkan berada di bagian ruangan lainnya. Terdapat 24 kemungkinan konfigurasi lingkungan dengan 15 kemungkinan posisi awal yang berbeda pada bagian ruangan pertama, serta 12 kemungkinan konfigurasi lingkungan pada bagian ruangan kedua. Berdasarkan perhitungan secara manual, kombinasi yang seperti ini akan menyebabkan terdapat $24 * 15 * 12$ kemungkinan konfigurasi lingkungan beserta posisi awal robot

otonom, atau lebih kurang sebanyak 4000 kemungkinan.

Adanya pembagian lingkungan kerja menjadi dua bagian ini dimaksudkan untuk memudahkan proses pembangunan representasi peta nantinya, selain juga untuk menguji kinerja algoritma yang dikembangkan pada berbagai kemungkinan konfigurasi lingkungan yang lebih banyak. Untuk keperluan inilah kemudian di antara kedua bagian ruangan akan dihubungkan oleh suatu lorong (jalur dengan dinding pada bagian kanan dan kirinya) yang memiliki ukuran lebar persis satu grid dan mempunyai karakteristik warna lantai yang dibedakan dari *grid-grid* lainnya. Hal ini menyebabkan, untuk suatu konfigurasi ruangan bagian kedua yang tertentu, robot akan memasuki ruangan dalam posisi yang sama. Hal ini nantinya dapat dimanfaatkan untuk mempermudah pencarian konfigurasi lingkungan yang sesuai, dalam rangka pembangunan representasi peta internal pada robot otonom. Untuk lebih jelasnya, pada Gambar 2 dapat dilihat beberapa contoh kemungkinan konfigurasi lingkungan yang digunakan pada penelitian ini.

Dari Gambar 2 dapat dilihat bahwa bagian ruangan tempat posisi awal robot otonom berada, terdapat pada sebelah kiri. Bagian yang diarsir hitam merupakan lorong yang menghubungkan ruangan dengan bagian lain tempat objek berada, yaitu bagian pada sebelah kanan. Dapat dilihat bahwa lantai pada bagian lorong sengaja dibedakan karakteristiknya agar mudah dikenali. Kemungkinan posisi awal robot otonom pada bagian ruangan yang pertama digambarkan oleh lingkaran abu-abu sedangkan pada saat robot memasuki bagian ruangan yang kedua digambarkan oleh kotak abu-abu yang akan selalu sama untuk suatu konfigurasi tertentu. Selanjutnya proses pembangunan peta pada robot otonom.



Gambar 2. Contoh beberapa konfigurasi lingkungan yang digunakan pada penelitian

2. Pembangunan Peta

Pada bagian sebelumnya telah disebutkan desain robot otonom yang digunakan, serta konfigurasi lingkungan tempat simulasi akan dilakukan. Pada bagian ini akan dijelaskan algoritma yang digunakan pada proses pembangunan peta dalam rangka menyelesaikan masalah navigasi pada robot otonom, khususnya masalah lokalisasi.

Algoritma yang digunakan mengadopsi algoritma pencarian (*searching*) pada struktur data pohon atau *tree* dengan menggunakan *heuristic* agar proses pencarian dapat berlangsung lebih cepat. Selain itu, akan diterapkan juga konsep *pruning* yang akan mengeliminasi semua *node* pada *tree* yang sudah tidak mungkin dikunjungi pada pencarian. Dengan melakukan hal ini, diharapkan robot otonom dapat menemukan konfigurasi lingkungan simulasi dengan lebih cepat, sehingga robot dapat menyelesaikan tugasnya dengan lebih cepat. Berikut akan dibahas mengenai konsep *searching* yang digunakan, serta *heuristic* dan *pruning*, sebelum dijelaskan mengenai bentuk adopsi yang digunakan untuk menyelesaikan lokalisasi robot otonom.

4.1. Strategi Searching, Heuristic dan Pruning

Proses pembangunan peta pada robot otonom dapat diselesaikan dengan cara merepresentasikan seluruh kemungkinan langkah yang dapat ditempuh dari posisi robot otonom hingga posisi tujuan pada struktur data *tree*, kemudian melakukan penelusuran terhadapnya. Pada saat simulasi dimulai, robot otonom tidak memiliki informasi posisi awal tempat dia berada. Berdasarkan perhitungan manual yang dilakukan sebelumnya, terdapat hingga ribuan kemungkinan posisi robot otonom dan konfigurasi lingkungannya. Jika proses penelusuran pada *tree* (*searching*) dilakukan dengan cara mengunjungi semua *node* yang ada, maka waktu yang diperlukan untuk menemukan konfigurasi lingkungan yang tepat akan menjadi sangat lama. Ditambah dengan adanya keterbatasan kapasitas memori robot otonom untuk dapat menyimpan semua *node* yang ada pada *tree* tersebut. Oleh karena itu, proses *searching* harus dilakukan dengan menggunakan strategi tertentu.

Pada setiap konfigurasi lingkungan yang digunakan terdapat sebuah lorong yang memiliki karakteristik lantai yang dapat dibedakan dari bagian ruangan lain. Hal ini dapat dimanfaatkan sebagai strategi untuk membagi proses pencarian ke dalam dua tahap yang berbeda. Pada tahap pertama, *searching tree* dilakukan untuk menemukan posisi lorong yang menghubungkan kedua bagian ruangan. Kemudian, pada saat robot otonom memasuki bagian ruangan yang kedua, hanya terdapat satu kemungkinan posisi pada setiap konfigurasi lingkungan, sehingga *searching* tahap kedua untuk menemukan konfigurasi lingkungan

yang sebenarnya dapat dilakukan dengan *node* lebih sedikit.

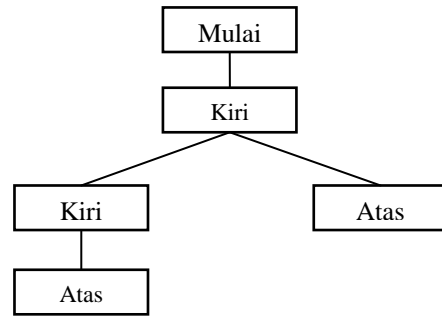
Selain itu, jenis penelusuran *tree* yang dilakukan adalah *informed searching* agar mendapatkan hasil pencarian yang lebih cepat. Searching jenis ini memanfaatkan informasi tambahan dari luar (*heuristic*) yang digunakan untuk memberikan bobot pada setiap *node* yang ada, kemudian penelusuran dilakukan dengan memprioritaskan kunjungan terhadap *node* dengan bobot yang paling tinggi [7]. *Heuristic* yang digunakan adalah banyaknya ujung jalur yang dimiliki oleh suatu posisi (atau direpresentasikan sebagai *node* dalam *tree*), serta jumlah jalur yang dimiliki oleh posisi yang direpresentasikan oleh *node* yang berada satu level di bawah *node* tersebut.

Dalam rangka memberikan hasil pencarian yang lebih optimum, selanjutnya diterapkan konsep *pruning* pada setiap *node* yang tidak mungkin dikunjungi karena adanya faktor fisik yang tidak memungkinkan robot mengunjungi *node* tersebut, misalnya karena adanya dinding. Informasi seperti ini dapat diperoleh dari bacaan sensor yang dimiliki oleh robot. Akan tetapi, perlu diperhatikan, bahwa dengan adanya error yang dimiliki oleh komponen sensor ini, memungkinkan terjadinya *pruning* yang tidak seharusnya, ataupun tidak dieliminasi *node* yang tidak dapat dikunjungi. Hal ini akan berdampak pada kesalahan dalam penelusuran *tree*. Oleh karena itu, untuk menangani hal ini, proses *searching tree* dilakukan dengan prinsip *backtracking* yang biasa dikenal pada pencarian jenis DFS.

4.2. Searching Tree untuk Menemukan Lorong

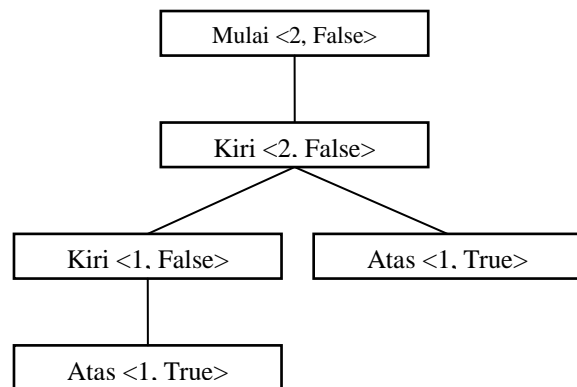
Misalkan posisi awal robot otonom dinyatakan oleh lingkaran merah pada Gambar 2.a dan 2.b sebelumnya sedangkan posisi tujuan dinyatakan oleh lingkaran biru, maka robot otonom harus dapat menemukan jalur yang dapat membawanya dari posisi awal menuju posisi tujuan tersebut. Pada Gambar 2.a salah satu solusi jalur yang dapat ditempuh adalah bergerak ke kiri sejauh dua langkah, lalu ke atas satu langkah. Sedangkan pada Gambar 2.b solusi jalur yang dapat ditempuh adalah bergerak ke kiri satu langkah, dan ke atas satu langkah.

Sebagai catatan tambahan, definisi arah kiri, kanan, atas dan bawah dalam ruang lingkup penelitian ini adalah berdasarkan orientasi pengamat, bukan berdasarkan orientasi robot. Hal ini dilakukan untuk menghindari ambiguitas pada orientasi arah yang dapat muncul karena robot perlu melakukan rotasi dalam bergerak ke kanan ataupun ke kirinya. Selanjutnya dari langkah-langkah yang dapat diambil tersebut direpresentasikan dalam bentuk *tree* seperti pada Gambar 3 (hanya ditampilkan sebagian).



Gambar 3. Langkah yang ditempuh untuk mencapai posisi lorong dengan konfigurasi lingkungan pada Gambar 2.a dan 2.b

Pada Gambar 3 terdapat *tree* yang digunakan pada proses *searching* tahap pertama, yaitu pencarian yang dilakukan untuk menemukan posisi lorong. Setiap *leaf* pada *tree* merupakan langkah terakhir yang akan membawa robot otonom pada posisi lorong. Kemudian dengan menggunakan *heuristic*, representasi *tree* di atas akan seperti pada Gambar 4 (hanya ditampilkan sebagian).



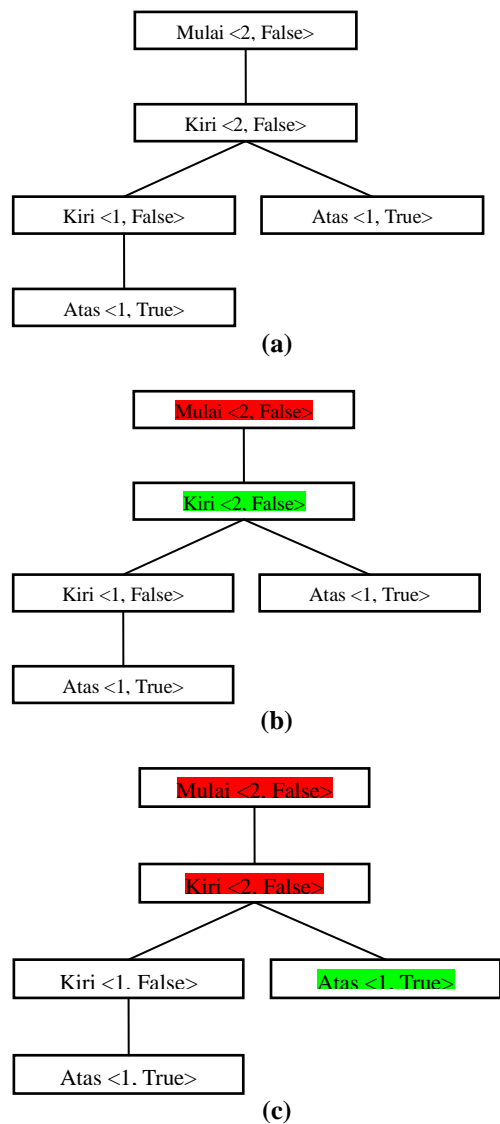
Gambar 4. Representasi *tree* pada gambar 3 dengan menggunakan informasi *heuristic*

Pada Gambar 4, mula-mula robot otonom berada pada posisi *root* pada *tree* (Gambar 5.a). *Node* pada *tree* mempunyai struktur A<B,C>. A memuat informasi kemungkinan arah gerak yang dapat diambil oleh robot. B memuat informasi banyaknya kemungkinan jalur yang dapat diambil oleh robot. Sedangkan C memuat informasi apakah *node* tersebut merupakan posisi tujuan atau tidak. *Root* dalam kasus ini hanya memiliki satu anak, yaitu Kiri<2, False> sehingga robot mengambil tindakan untuk bergerak ke arah kiri (Gambar 5.b). Penentuan keputusan untuk mengambil tindakan ke arah mana ditentukan oleh banyaknya jalur yang dapat diambil oleh robot pada suatu *node*. Atau dengan kata lain dilihat dari besar nilai B-nya. Semakin besar nilai B-nya, maka skala prioritas untuk mengunjungi *node* tersebut semakin besar. Namun apabila ditemukan anak-anak dari suatu *node* tersebut mempunyai nilai B yang sama besar,

maka penentuan- nya akan dilakukan secara acak (Gambar 5.c).

Jika ditampilkan secara keseluruhan, bentuk *tree* yang menyimpan semua kemungkinan langkah robot otonom pada proses *searching* tahap pertama untuk menemukan posisi lorong adalah seperti pada Gambar 6.

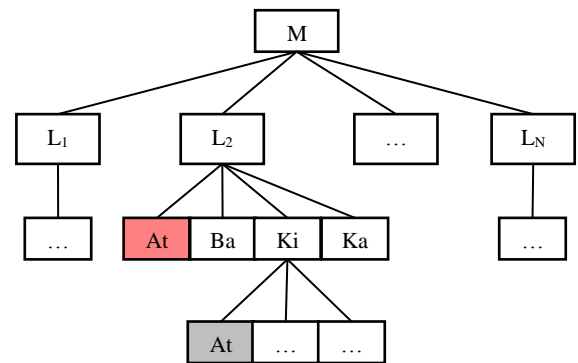
Dari Gambar 6 dapat dilihat bahwa pertama kali dilakukan pengelompokan semua kemungkinan konfigurasi lingkungan ke dalam n -kelompok yang berbeda, yaitu kelompok L_1 hingga kelompok L_n . Pengelompokan ini bertujuan agar dapat dilakukan *pruning* lebih awal dalam rangka mengurangi jumlah *node* pada *tree* yang harus ditelusuri selama proses *searching*. Pengelompokan ini dilakukan berdasarkan karakteristik dinding yang terdapat di sekeliling robot otonom pada suatu kondisi tertentu, yaitu apakah terdapat dinding di bagian depan, kiri, kanan ataupun belakang robot.



Gambar 5. Proses pengunjungan *node-node* pada *tree* oleh robot.

Selanjutnya, pada saat simulasi dimulai robot otonom dengan menggunakan sensornya akan berupaya mencocokkan keadaan yang ditemuinya dengan kelompok-kelompok yang telah ditentukan sebelumnya. Melalui cara ini kemudian dapat dilakukan *pruning* terhadap *node-node* yang merepresentasikan kelompok konfigurasi lingkungan yang tidak memenuhi karakteristik yang sama dengan yang ditemukan oleh robot otonom. *Searching* selanjutnya dilakukan pada kelompok konfigurasi lingkungan yang tersisa. Dari semua *node* yang tersisa kemudian diberikan nilai *heuristic* yang menentukan kebaikan *node* untuk lebih dahulu dikunjungi.

Langkah *pruning* yang selanjutnya dapat dilakukan adalah untuk mengeliminasi *node-node* yang tidak mungkin dikunjungi karena adanya faktor fisik dinding. Dengan berasumsi bahwa Gambar 2.a merupakan contoh konfigurasi yang tergabung dalam kelompok L_2 , maka *node* yang tereliminasi karena adanya dinding pada contoh ini direpresentasikan oleh warna merah pada Gambar 6. *Node* yang tidak perlu ditelusuri ini kemudian akan dapat mengurangi jumlah *node* yang ada pada *tree* untuk proses *searching* yang lebih cepat. Ketika robot otonom menemukan *node* yang merupakan posisi lorong (direpresentasikan oleh warna abu-abu), maka semua *node* dari root hingga *node* tersebut akan disimpan sebagai jalur dari posisi awal hingga posisi tujuan. Selanjutnya proses *searching* dapat dilanjutkan pada tahap selanjutnya untuk menentukan konfigurasi lingkungan yang sebenarnya.



Gambar 6. Representasi keseluruhan kemungkinan langkah robot untuk mencapai posisi lorong pada *searching* tahap pertama

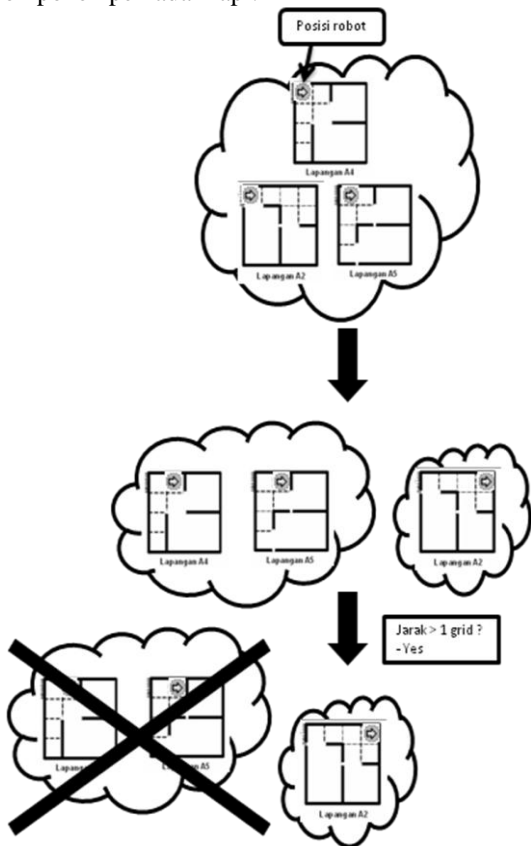
4.3. Searching Tree untuk Metode Lokalisasi

Proses *searching* tahap kedua bertujuan untuk menentukan konfigurasi yang sebenarnya dari lingkungan tempat robot otonom berada. Melalui pembangunan peta secara internal ini, robot otonom kemudian dapat menentukan jalur terbaik yang dapat membawanya untuk menemukan dan menyelamatkan objek sesuai dengan yang diharapkan.

Searching tree pada tahap kedua sebenarnya

tidak jauh berbeda dari tahap pertama. Hanya saja, pada tahap ini proses pencarian dilakukan dengan cara mencocokkan keadaan yang ditemui robot di lapangan dengan konfigurasi lingkungan yang ada, sampai akhirnya tersisa satu kemungkinan konfigurasi lingkungan yang berarti merupakan solusi dari konfigurasi lingkungan yang sebenarnya.

Proses pencocokan konfigurasi lingkungan ini dimulai dari bagian ruangan yang kedua karena memiliki jumlah kemungkinan yang lebih sedikit. Jika pada pencarian tahap pertama, robot otonom diminta untuk menemukan posisi lorong, pada tahap ini posisi tujuan adalah ruangan tempat objek akan diselamatkan berada. Selain itu, robot otonom juga diminta untuk menemukan objek api dan berusaha memadamkannya dengan menggunakan komponen pemadam api.



Gambar 7. Proses *pruning* dan pencocokan konfigurasi lingkungan

Pencocokan konfigurasi lingkungan dilakukan dengan mengecek konfigurasi dinding yang ada pada setiap posisi, dimulai dari posisi pertama tempat robot otonom memasuki bagian ruangan kedua (Gambar 7). Robot kemudian bergerak ke salah satu posisi yang dipilih pada representasi *tree*-nya, dan melakukan pengecekan pada pada posisi tersebut. Setiap konfigurasi lingkungan yang tidak memiliki konfigurasi dinding yang sama pada posisi yang bersesuaian kemudian akan dibuang dari daftar kemungkinan. Begitu seterusnya sampai tersisa satu konfigurasi lingkungan terakhir.

Setelah robot otonom berhasil menentukan konfigurasi lingkungan yang sesuai, maka konfigurasi tersebut akan dijadikan sebagai representasi peta internal yang dapat memandu pergerakan robot otonom tanpa harus melakukan searching seperti sebelumnya. Seandainya objek yang akan diselamatkan pun ditemukan sebelum representasi peta internal selesai dibangun, peta yang diperoleh pada akhirnya dapat digunakan untuk menemukan jalan kembali menuju lorong, atau pun untuk menentukan posisi api yang harus dipadamkan.

Robot otonom kemudian diharapkan kembali ke posisi awal dengan membawa objek yang ditemukannya. Saat posisi objek ditemukan, robot otonom akan menggunakan komponen jaring-pencengkram yang telah dikembangkan untuk mengambil dan mengangkut objek yang akan diselamatkan tersebut. Sesuai dengan spesifikasi pada Gambar 1 sebelumnya, objek yang diselamatkan akan ditempatkan pada bagian atas robot, agar tidak memberikan penambahan dimensi yang berarti sehingga dapat menghindari perbedaan pada proses komputasi saat sebelum dan sesudah menemukan objek.

4.4. Kembali pada Posisi Awal

Setelah berhasil menemukan posisi lorong dengan menggunakan representasi peta internal yang sudah ditemukan sebelumnya, robot harus berpindah kembali pada bagian ruangan yang pertama. Misi selanjutnya adalah untuk membawa objek yang ditemukan kembali ke posisi awal tempat robot memulai simulasi. Akan tetapi, perlu diingat bahwa robot sesungguhnya belum memiliki representasi peta internal mengenai konfigurasi lingkungan yang ada pada bagian ruangan pertama ini. Sebelumnya robot otonom hanya menyimpan informasi jalur yang menghubungkan antara posisi lorong dan posisi awal.

Untuk dapat menentukan jalur kembali ke posisi awal sebenarnya dapat dilakukan dengan dua cara. Cara yang pertama adalah dengan memanfaatkan informasi jalur yang menghubungkan posisi lorong dengan posisi awal simulasi yang telah ditemukan sebelumnya. Sedangkan cara yang kedua adalah dengan membangun representasi peta internal bagian ruangan yang pertama.

Cara yang pertama mungkin merupakan metode yang paling cepat untuk dilakukan tetapi tidak ada jaminan bahwa informasi jalur yang disimpan sebelumnya merupakan jalur tercepat yang dapat membawa robot otonom kembali ke posisi awal simulasi. Cara kedua merupakan metode yang lebih baik secara keseluruhan. Dengan membangun representasi peta internal terlebih dahulu, robot otonom dapat menentukan jalur terbaik yang dapat ditempuh menuju posisi awal simulasi. Proses penentuan jalur terbaik ini dapat dilakukan salah satunya dengan memanfaatkan representasi peta

internal dalam bentuk *graph*, kemudian menggunakan metode *Shortest Path* Dijkstra [8].

Walaupun proses pembangunan peta internal di sini terkesan dapat memperlambat proses penyelesaian masalah, pada kenyataannya tidak ada jaminan bahwa waktu yang diperlukan untuk melewati rute sesuai informasi sebelumnya akan lebih cepat dibandingkan dengan membangun representasi peta internal serta menentukan jalur terbaik menuju posisi awal simulasi. Pada penelitian ini, kami menerapkan cara kedua. Proses pembangunan peta internal dilakukan seperti proses pembangunan peta internal pada bagian ruangan kedua sebelumnya.

3. Kesimpulan dan Tahap Penelitian Selanjutnya

Pada penelitian ini telah dibangun suatu aplikasi yang bertujuan untuk mensimulasikan kasus *search-and-safe* yang sering dijumpai dalam kehidupan sehari-hari. Pada kasus ini, robot otonom bergerak diharapkan dapat menggantikan tugas manusia dalam menemukan dan menyelamatkan suatu objek tertentu dalam lingkungan kerja yang berbahaya. Salah satu contohnya adalah pada proses evakuasi korban gedung atau ruangan yang terbakar. Penggunaan robot otonom pada kasus seperti ini akan dapat menghindari resiko kecelakaan yang mungkin dialami petugas evakuasi.

Salah satu hal yang kemudian menjadi fokus pada penelitian ini adalah masalah navigasi robot otonom dalam lingkungan yang tidak dikenal sebelumnya. Robot otonom diminta untuk dapat secara mandiri membangun representasi peta internal mengenai lingkungan tempat dia bekerja agar dapat menyelesaikan masalah lokalisasi. Hal ini akan berdampak untuk mendukung proses penyelesaian masalah kognisi dan kontrol gerak selanjutnya.

Salah satu tantangan dalam masalah lokalisasi adalah masalah ketidakpastian pada lingkungan. Untuk alasan penyederhanaan masalah, robot otonom sebelumnya akan diberikan beberapa kemungkinan konfigurasi lingkungan yang ada. Selanjutnya diasumsikan bahwa konfigurasi lingkungan yang sebenarnya tidak akan berbeda dari salah satu kemungkinan konfigurasi tersebut. Dengan demikian, pada penelitian ini bertujuan untuk mengembangkan suatu metode pada robot otonom dalam menentukan konfigurasi lingkungan dari beberapa kemungkinan yang ada. Proses pencarian solusi konfigurasi lingkungan yang sebenarnya ini bergantung pada kemampuan persepsi robot otonom, sekaligus kemampuan untuk melakukan proses komputasi yang diperlukan. Untuk itu juga dibahas desain perangkat keras robot otonom yang digunakan, hingga kepada komponen pemadam-api dan jari-penggenggam walaupun

masih dalam ruang lingkup yang sederhana.

Penelitian ini telah berhasil menyelesaikan studi kasus *search-and-safe* pada lingkungan yang tertutup dan terbatas. Penerapan adopsi algoritma *searching tree* dengan metode heuristik dan *pruning* diharapkan dapat digunakan pada kehidupan yang sebenarnya. Akan tetapi, masih banyak hal yang harus dilakukan sebelum hal tersebut dapat direalisasikan.

Penelitian ini masih memiliki beberapa keterbatasan yang diharapkan dapat diselesaikan pada tahap pengembangan yang selanjutnya. Selain masalah keterbatasan perangkat keras, serta ukuran dan kondisi lingkungan yang digunakan, metode algoritma *searching tree* dengan heuristik dan *pruning* masih menggunakan asumsi bahwa kemungkinan konfigurasi lingkungan diketahui sebelumnya. Pada kenyataannya, dalam kehidupan yang sebenarnya kemungkinan konfigurasi lingkungan bersifat tidak terbatas dan beragam. Selain itu, pada penelitian ini konfigurasi lingkungan yang digunakan berbentuk lorong-lorong atau model labirin dengan dinding sebagai rintangannya. Hal ini dimaksudkan untuk memudahkan masalah persepsi robot otonom untuk mendapatkan informasi yang akan diperlukan selama proses pembangunan peta internal. Dalam kehidupan sehari-hari kenyataannya masih banyak terdapat rintangan lain yang dapat menghalangi pergerakan robot otonom. Diharapkan masalah-masalah seperti ini dapat menjadi pertimbangan pada tahap penelitian yang selanjutnya.

4. Ucapan Terima Kasih

Penelitian ini dapat terwujud atas dasar kerjasama Fasilkom UI dengan Tim Robot UI ketika mengikuti Kontes Robot Cerdas Indonesia (KRCI) 2009 Divisi Expert Single.

REFERENSI

- [1] Siegwart, Roland dan Illah R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*, London: MIT Press, 2004.
- [2] Meger, David, Ioannis Rekleitis, and Gregory Dudek, *Simultaneous Planning, Localization, and Mapping in CameraSensor Network*, Minneapolis: Springer, 2006.
- [3] Howard, Andrew, Maja J Mataric, and Gaurav S. Sukhatme, "Localization for mobile robot teams using maximum likelihood estimation". Switzerland, in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.
- [4] R. Kurazume and S. Hirose, *An Experimental Study of a Cooperative Positioning System*. Autonomous Robot, 2000.
- [5] Rekleitis, Ioannis M., Gregory Dudek, and Evangelos E. Millios. 2002. *On the Positional*

Uncertainty of Multi-robot Cooperative Localization. Washington DC: Multi Robot Systems Workshop, Naval Research Laboratory.

- [6] Roumeliotis, Stergios I., and George A. Berkey, "Distributed multirobot localization", *IEEE Transaction on Robotics and Automation*, 2002
- [7] Russell, J Stuart and Peter Norvig. *Artificial Intelligence – A Modern Approach Second Edition*. New Jersey: Prentice Hall, 2003.
- [8] Kenneth H. Rose, *Discrete Mathematics and Its Applications Sixth Edition*, McGraw Hill International Edition, 2007.