

**PENCARIAN RUTE TERPENDEK
ARENA KONTES ROBOT PEMADAM API INDONESIA (KRPAI)
MENGUNAKAN ALGORITMA *HILL CLIMBING***

Pamor Gunoto

Dosen Tetap Program Studi Teknik Elektro
Universitas Riau Kepulauan (UNRIKA) Batam

ABSTRAK

Kontes Robot Indonesia (KRI) adalah ajang bentuk kreativitas dari mahasiswa dalam penerapan teknologi tinggi pada robot. Dalam pertandingan ini menggunakan arena sejenis simulasi interior rumah dengan 4 ruangan. Salah satu kriteria yang dipertandingkan adalah robot dapat memadamkan api dengan cepat. Oleh karena itu diperlukan suatu simulasi untuk mendapatkan rute terpendek yang harus dilalui oleh robot agar supaya waktu yang diperlukan dalam memadamkan api dapat secepat mungkin. Penggunaan algoritma dipakai dalam menentukan rute terpendek yang dapat ditempuh agar supaya waktu yang diperlukan dapat lebih singkat. Pada penelitian ini digunakan algoritma Hill Climbing dengan mengetahui jarak tiap ruangan yang akan dilaluinya. Proses pencarian adalah mendapatkan solusi jarak rute terpendek dengan nilai heuristik yang terbaik. Hasil simulasi dengan menggunakan program Matlab didapatkan bahwa rute terpendek (tercepat) yang harus dilalui oleh robot adalah ruangan 2-1-3-4 (=795,6 cm).

1. PENDAHULUAN

1.1 Latar Belakang

Kontes Robot Pemadam Api Indonesia (KRPAI) dapat merupakan suatu wacana yang sangat menarik untuk mengimplementasikan gagasan dan ide menjadi suatu robot yang fungsional dengan memanfaatkan pengetahuan yang sangat multidisiplin. Robot-robot tersebut harus dirancang dan dibuat sendiri, dengan menggunakan sensor-sensor, aktuator serta mikrokontroler yang ada dan harus diprogramkan sesuai dengan tema kontes setiap tahunnya. Selain itu kerjasama yang baik antara anggota tim peserta juga akan menjadi faktor penting dalam kontes robot. Setiap tim akan memiliki gagasan strategi yang terbaik untuk memenangkan kontes tersebut, sehingga dapat menjadikan tolak ukur suatu perguruan tinggi dalam penerapan teknologi yang tepat guna dan sasaran.

Dalam KRPAI salah satu penentuan kriteria menjadi pemenang adalah dapat memadamkan api di dalam ruangan dengan secepat mungkin. Dalam hal ini maka dengan menentukan rute terpendek di dalam ruangan arena akan diperoleh waktu pemadaman api yang tercepat pula.

Pencarian rute terpendek saat melakukan perjalanan merupakan hal yang perlu dilakukan selain menemukan kota tujuan juga untuk menghemat biaya perjalanan. Algoritma pencarian (*searching algorithm*) yang mendasari kerja dari software atau program banyak modelnya, akan tetapi keefektifan suatu algoritma pencarian dalam menemukan rute atau tujuan tergantung pada proses atau langkah-langkah yang di berikan oleh algoritma itu sendiri. Untuk dapat membuat robot cerdas adalah dengan cara mengimplementasikan metode-metode kecerdasan buatan pada robot tersebut agar robot dapat melakukan penjelajahan dan pencarian sendiri.

Hill Climbing (pendakian bukit) merupakan metode pembangkitan dan pengujian secara fungsi *heuristic*. Fungsi *heuristic* digunakan untuk mengevaluasi keadaan-keadaan problem individual dan menentukan seberapa jauh hal tersebut dapat digunakan untuk mendapatkan solusi yang diinginkan.

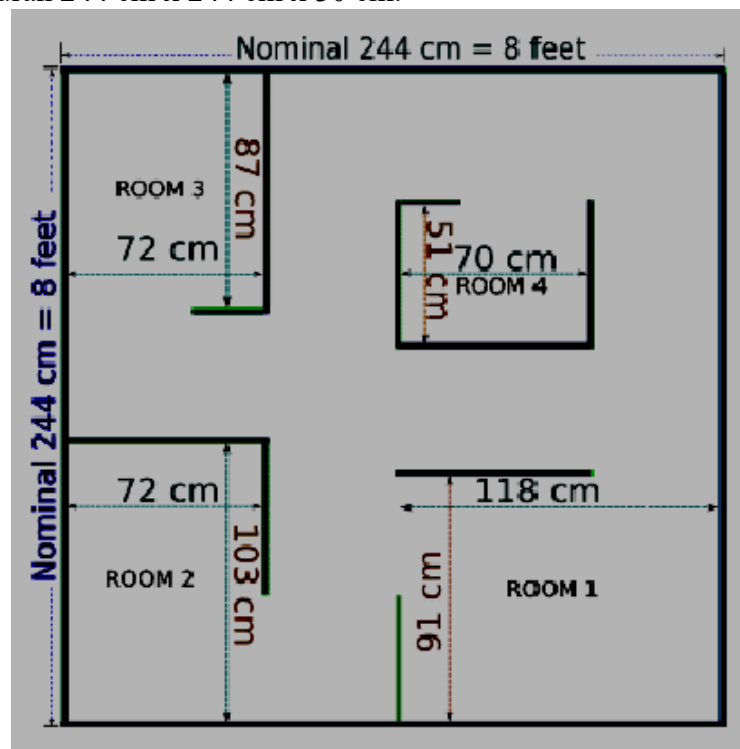
1.2 Tujuan

Menentukan rute yang akan dilalui robot dengan jarak terpendek pada arena Kontes Robot Pemadam Api Indonesia (KRPAI) dengan menggunakan algoritma metode simple *hill climbing*. Sehingga didapatkan waktu yang paling cepat oleh robot dalam melalui ruangan-ruangan yang ada pada arena.

2. TINJAUAN PUSTAKA

2.1 Arena KRPAI

Arena yang digunakan pada KRPAI adalah mensimulasikan interior dari sebuah rumah dengan 4 ruangan. Penomoran ruangan adalah sebagai berikut : ruang 1 (room 1) adalah ruang dengan ukuran terbesar, berturut-turut ruang 2 (room 2), ruang 3 (room 3) dan ruang 4 (room 4) adalah ruangan yang berada searah dengan jarum jam dengan ruang 1 seperti gambar 1. Pada ruang 1 terdapat dua pintu, yang satu lokasinya bisa berubah, yang lain lokasinya tetap. Ruang 4, lokasi pintu juga bisa berubah. Lapangan terbuat dari papan multipleks dengan ketebalan 1,8 s.d 2 cm dan berukuran 244 cm x 244 cm x 30 cm.



Gambar 1. Arena KRPAI

Lapangan memiliki roda agar dapat dipindah atau diputar dengan mudah. Lapangan dibuat sedekat mungkin dengan lantai agar mempermudah dalam pengambilan robot saat pertandingan.

2.2 Metode Pencarian *Hill Climbing*

Teknik *heuristic* adalah teknik yang digunakan untuk mempercepat pencarian solusi. Teknik *heuristic* digunakan untuk mengeliminasi beberapa kemungkinan solusi tanpa harus

mengeksplorasinya secara penuh. Selain itu teknik *heuristik* juga membantu memutuskan kemungkinan solusi mana yang pertama kali dievaluasi. Ada beberapa metode pencarian *heuristik* salah satunya adalah metode *hill climbing*.

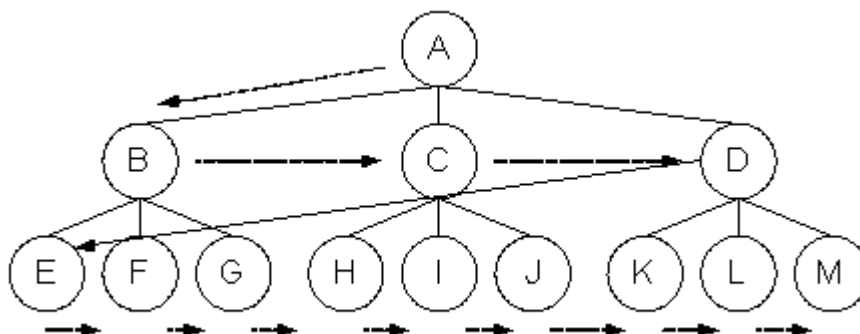
Hill climbing adalah metode yang dikenal untuk pencarian lokal. Gagasan untuk metode *hill climbing* dimulai secara acak dari state yang sudah ada, bergerak ke tetangga dengan nilai evaluasi yang terbaik dan jika suatu minimum lokal telah dicapai lalu memulai lagi secara acak pada state yang berbeda.

Metode Pencarian *Hill Climbing* adalah suatu metode untuk mencari dan menentukan rute yang paling singkat dengan memperkecil jumlah kota atau tempat yang disinggahi dengan menggunakan cara *heuristic*. Cara kerjanya adalah menentukan langkah berikutnya dengan menempatkan node yang akan muncul sedekat mungkin dengan sarasannya.

Langkah-langkah Algoritma Pencarian *Hill Climbing* adalah sebagai berikut :

1. Mulai dari keadaan awal dan dilakukan pengujian, apabila merupakan tujuan maka berhenti dan bila tidak maka dilanjutkan dengan keadaan sekarang sebagai keadaan awal.
2. Dilanjutkan hingga tujuan tercapai atau hingga iterasi tidak memberikan perubahan pada keadaan sekarang.
 - a. Tentukan *successor* sebagai nilai *heuristik* terbaik dari *successor-successor*.
 - b. Kerjakan untuk tiap operator yang digunakan oleh keadaan sekarang
 - Gunakan operator dan bentuk keadaan baru
 - Evaluasi keadaan baru tersebut. Jika merupakan tujuan maka selesai. Jika bukan, bandingkan nilai *heuristiknya* dengan *successor*. Jika lebih baik, jadikan nilai *heuristiknya* keadaan baru tersebut.
 - c. Jika *successor* lebih baik daripada nilai *heuristik* keadaan sekarang, ubah node *successor* menjadi keadaan sekarang.
 - Algoritma akan berhenti kalau mencapainilai optimum lokal
 - Urutan penggunaan operator akan sangat berpengaruh pada penemuan solusi
 - Tidak diijinkan untuk melihat satupun langkah sebelumnya.

Berdasarkan penjelasan Algoritma Pencarian *Hill Climbing* untuk mencari jarak terpendek dapat digambarkan dalam bentuk diagram pohon seperti dibawah ini



Gambar 2. Diagram pohon Pencarian *Hill Climbing*

2.3 Travelling Salesman Problem (TSP) dengan Pencarian *Hill Climbing*

Pada permasalahan TSP, disini ruang keadaan berisi semua kemungkinan lintasan yang mungkin. Operator digunakan untuk menukar posisi kota-kota yang bersebelahan. Apabila ada 4 kota, untuk mencari kombinasi lintasan dengan menukar posisi urutan 2 kota, maka akan mendapatkan kombinasi sebanyak :

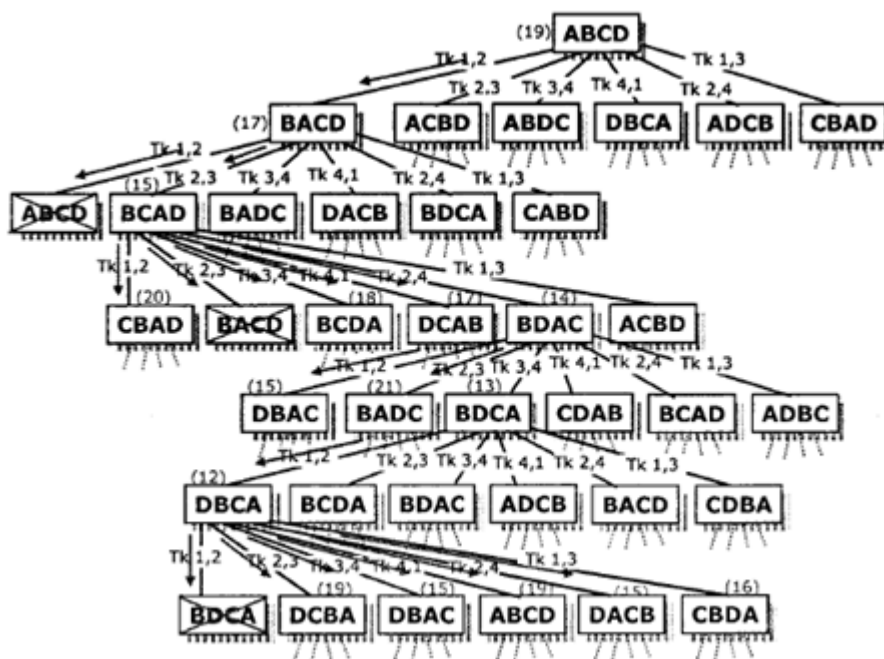
$$\frac{4!}{2!(4-2)!}$$

Atau sebanyak 6 kombinasi. Fungsi *heuristic* yang digunakan adalah panjang lintasan yang terjadi.

Keenam kombinasi ini akan dipakai semuanya, sebagai operator :

1. Tukar 1,2 (menukar urutan posisi kota ke-1 dengan kota ke-2)
2. Tukar 2,3 (menukar urutan posisi kota ke-2 dengan kota ke-3)
3. Tukar 3,4 (menukar urutan posisi kota ke-3 dengan kota ke-4)
4. Tukar 4,1 (menukar urutan posisi kota ke-4 dengan kota ke-1)
5. Tukar 2,4 (menukar urutan posisi kota ke-2 dengan kota ke-4)
6. Tukar 1,3 (menukar urutan posisi kota ke-1 dengan kota ke-3)

Proses pencarian TSP seperti gambar dibawah ini



Gambar 3. Diagram proses pencarian TSP

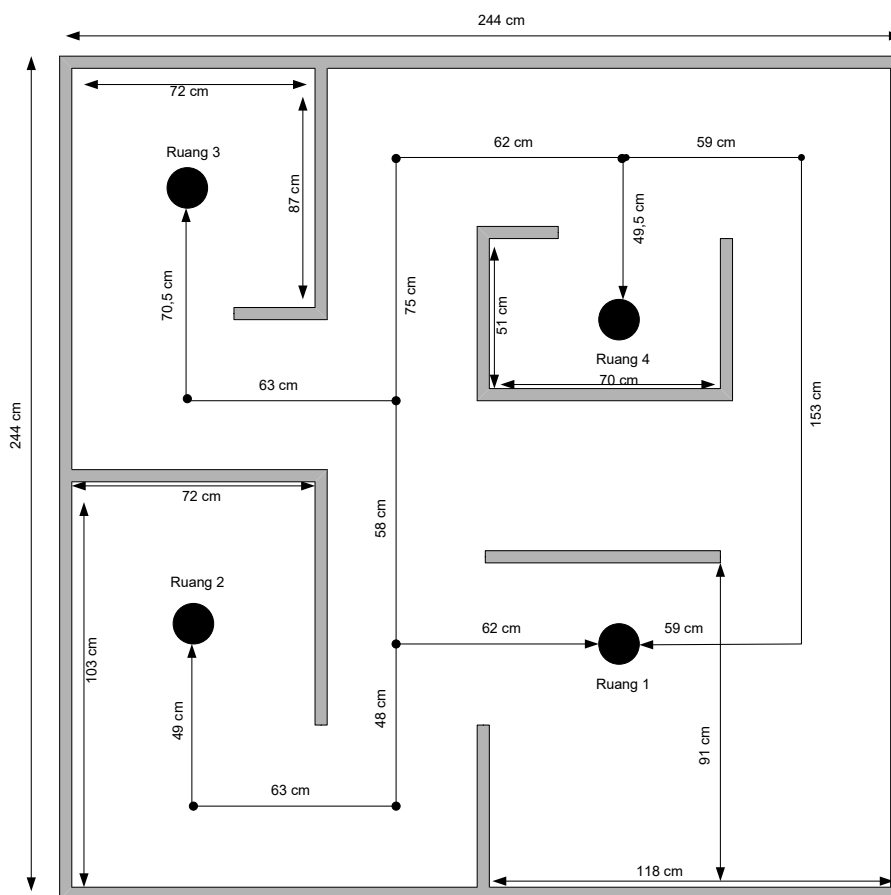
Algoritma TSP

- Keadaan awal, lintasan ABCD (=19)
- Level pertama, *hill climbing* mengunjungi BACD (=17), dimana BACD (=17) < ABCD (=19), sehingga BACD menjadi pilihan selanjutnya dengan operator Tukar 1,2
- Level kedua, mengunjungi ABCD, karena operator Tukar 1,2 sudah dipakai BACD, maka pilih node lain yaitu BCAD (=15), dimana BCAD (=15) < BACD (=17)

- Level ketiga, mengunjungi CBAD (=20), CBAD (=20) > BCAD (=15) maka pilih node lain yaitu BCDA (=18), pilih node lain yaitu DCAB (=17), pilih node lain yaitu BDAC (=14) < BCAD (=15)
- Level keempat, mengunjungi DBAC (=15), dimana DBAC (=15) > BDAC (=14), maka pilih node lain yaitu BADC (=21), pilih node lain yaitu BDCA (=13), dimana BDCA (=13) < BDAC (=14)
- Level kelima, mengunjungi DBCA (=12), dimana DBCA (=12) < BDCA (=13)
- Level keenam, mengunjungi BDCA, karena operator Tukar 1,2 sudah dipakai DBCA, maka ilih node lain yaitu DCBA, pilih ABCD, pilih DACB, pilih CBDA
- Karena sudah tidak ada node yang memiliki nilai heuristik yang lebih kecil dibanding nilai heuristik DBCA, maka node DBCA (=12) adalah lintasan terpendek (SOLUSI)

3. HASIL DAN PEMBAHASAN

Dalam proses pencarian jarak terpendek pada arena robot, dilakukan pemetaan terlebih dahulu. Pemetaan dilakukan untuk mengetahui gambaran area yang akan dilalui sehingga dengan adanya pemetaan ini robot dapat mengetahui rute-rute yang yang dapat diambil. Pemetaan pada arena yang dilakukan dapat dilihat pada gambar 4.



Gambar 4. Pemetaan pada arena KRPAI

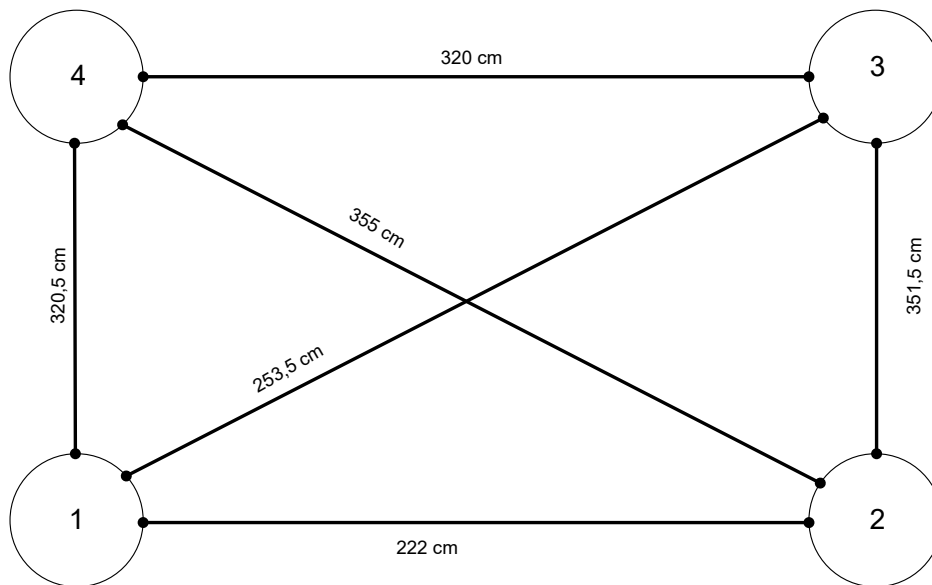
Penentuan rute diambil dengan mengambil posisi tengah dari jalur pada arena, sehingga dapat diketahui jarak masing-masing jalur dari pemetaan yang dilakukan pada arena. Setiap robot diharuskan melewati setiap ruangan yang ada dalam mencari titik api yang akan

dipadankan. Rute yang terpendek menuju tujuan yang diinginkan ditentukan dengan menggunakan metode pencarian *hill climbing*. Dalam proses algoritma ini digunakan bantuan program Matlab untuk mempercepat proses perhitungan dan komputasi data. Tabel 1 memperlihatkan jarak antar ruangan dari hasil pemetaan yang telah dilakukan.

Tabel 1. Jarak rute antar ruang arena

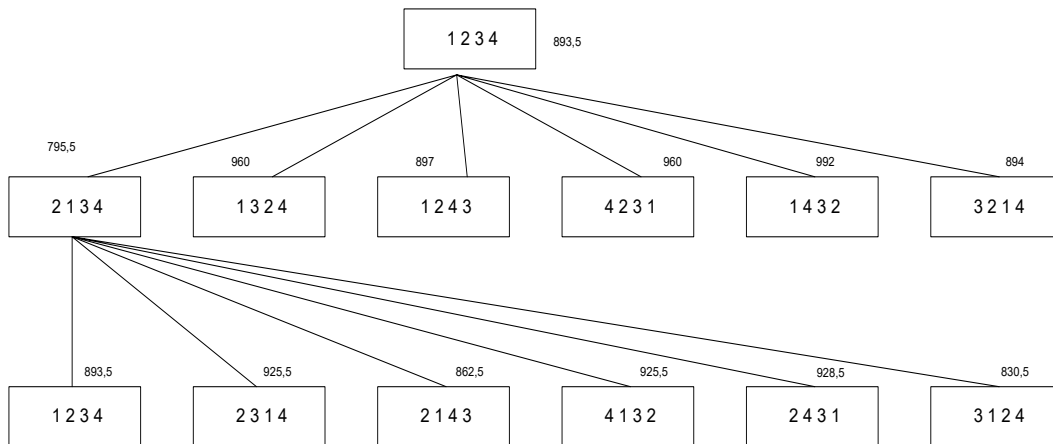
No	Ruang	Jarak Rute (cm)	Jarak total antar ruang (cm)
1	1 - 2	62+48+63+25	222
2	2 - 3	49+63+48+58+63+70,5	351,5
3	3 - 4	70,5+63+75+62+49,5	320
4	4 - 1	49,5+59+153+59	320,5
5	4 - 2	62+75+58+48+63+49	355
6	1 - 3	62+58+63+70,5	253,5

Dari hasil diatas dapat digambarkan secara lebih sederhana jarak antar ruang seperti pada gambar 5.



Gambar 5. Jarak rute antar ruang

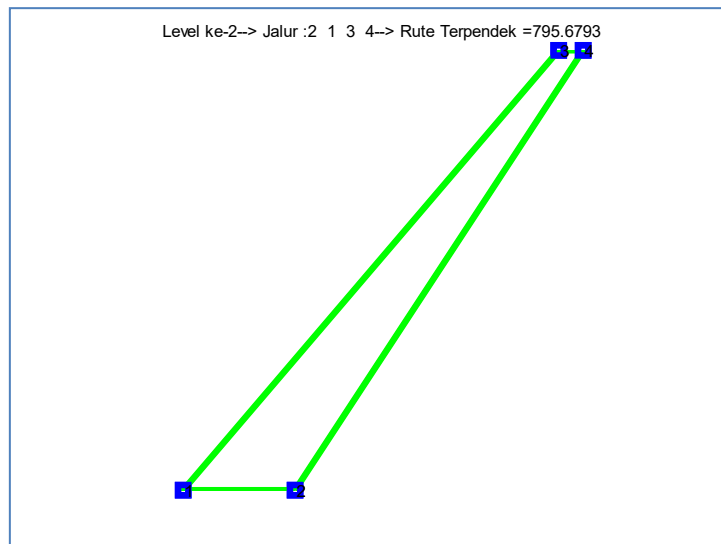
Adapun perhitungan pencarian untuk mendapatkan rute yang terpendek dapat dilakukan dengan membuat diagram seperti gambar 6 dibawah ini



Gambar 6. Proses pencarian rute terpendek dengan metode *hill climbing*

Pada gambar 6 diatas terlihat bahwa, keadaan awal lintasan terpilih adalah ruang 1-2-3-4(893,5). Pada level pertama, *hill climbing* akan memilih nilai *heuristik* terbaik dari keenam successor yang ada, yaitu : 2-1-3-4(795,5), 1-3-2-4(960), 1-2-4-3(897), 4-2-3-1(960), 1-4-3-2(992) dan 3-2-1-4(894) dan yang terpilih adalah 2-1-3-4 karena memiliki nilai heuristik yang paling kecil (=795,5). Kemudian dari 2-1-3-4 ini akan dipilih nilai heuristik terbaik dari successornya yaitu 1-2-3-4(893,5), 2-3-1-4(925,5), 2-1-4-3(862,5), 4-1-3-2(925,5), 2-4-3-1(928,5) dan 3-1-2-4(830,5). Ternyata dari keenam successor tersebut tidak memiliki nilai heuristik yang lebih kecil dari 795,5 sehingga tidak ada perubahan nilai keadaan (2-1-3-4). Hasil yang diperoleh dengan rute yang terpendek adalah 2-1-3-4 dan merupakan solusi.

Hasil pencarian rute terpendek dengan menggunakan program matlab dalam proses komputasi dapat dilihat pada gambar 7 dibawah ini.



Gambar 7. Pencarian rute terpendek dengan Matlab

4. KESIMPULAN

Pencarian rute terpendek menggunakan algoritma *hill climbing* pada arena KRPAI ditentukan oleh pemetaan arena yang dibuat.

Hasil yang diperoleh (solusi) dengan menggunakan algoritma *hill climbing* merupakan nilai perbandingan terkecil dari diantara *successor-successor* yang dibangkitkan dari data yang ada.

Hasil penelitian didapatkan bahwa rute terpendek (tercepat) yang harus dilalui oleh robot adalah ruangan 2-1-3-4 (=795,5 cm)

DAFTAR PUSTAKA

- Kusumadewi, Sri., 2003, *Artificial Intelligence (Teknik & Aplikasinya)*, Graha Ilmu, Jogjakarta
- Purnomo, Hari dan Kusumadewi, Sri., 2005, *Optimasi dengan Teknik Heuristik*, Graha Ilmu, Yogyakarta
- Linfield, G and Penny, J., 1995, *Numerical Methods Using Matlab*, Ellis Horwood Limited
- Nirabel, Ferdi dan Thiang., 2009, *Robot Mobil Pencari Rute Terpendek Menggunakan Metode Steepest Ascent Hill Climbing*, Seminar Nasional Aplikasi Teknologi Informasi, Jogjakarta
- Wibowo, AW, Purwanto, Y dan Purwitasari, D., 2005, *Implementasi dan Analisis Algoritma Pencari Rute Terpendek di Kota Surabaya*, Jurnal Penelitian dan Pengembangan Telekomunikasi