

# Penerapan Algoritma *Branch and Bound* untuk Penyusunan Strategi dalam Permainan Lagori

Dengan Menggunakan Robot Pada Perlombaan  
ABU Robocon 2022

Maharani Ayu Putri Irawan - 13520019

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
rani.api3939@gmail.com

**Abstract**—Algoritma *Branch and Bound* merupakan algoritma pencarian pada pohon ruang status yang biasa digunakan dalam persoalan optimasi. Permainan lagori membutuhkan teknik minimisasi jarak yang diperlukan untuk berjalan dari suatu titik ke titik lain guna mengambil dan menyusun ulang tatanan lagori. Oleh karena itu, pada persoalan minimisasi ini, algoritma *branch and bound* dapat dimanfaatkan. Dalam pengujian program, disimpulkan bahwa solusi yang dihasilkan optimal dengan menggunakan metode penghitungan batas (*bound*) dengan *Manhattan Distance*.

**Keywords**—*branch and bound*; optimasi; minimasi; jarak; lagori; manhattan distance

## I. LATAR BELAKANG

Perlombaan ABU Robocon merupakan kontes internasional bergengsi dalam bidang robotika yang memiliki tema yang bervariasi setiap tahunnya. Berbagai macam permainan, termasuk olahraga, telah menjadi tema perlombaan. Objektif dari perlombaan ini adalah menciptakan robot yang dapat mensimulasikan berbagai permainan yang menjadi tema. Perlombaan ini diikuti oleh berbagai tim dari beragam universitas di dunia. Perwakilan Indonesia sendiri, dipilih melalui Kontes Robot Indonesia dalam bidang lomba Kontes Robot ABU Indonesia, yang biasa disebut dengan KRAI. Perlombaan tahun 2022 memiliki tema “Lagori”, yakni permainan lempar batu yang berasal dari India.

Tugas dari salah satu robot yang dibangun dalam rangka berpartisipasi dalam perlombaan ini adalah melempar bola untuk menghancurkan tumpukan lagori yang bertingkat lima dan memiliki ukuran yang bervariasi. Lagori ini terletak di tengah lapangan yang berukuran relatif luas. Kemudian, robot lain, masih dari tim yang sama, bertugas untuk mengambil dan menyusun ulang silinder-silinder lagori dalam urutan tumpukan yang sesuai dengan waktu sesingkat mungkin. Perolehan poin didasarkan pada jumlah lagori yang berhasil dijatuhkan, kesesuaian susunan tumpukan ulang, dan waktu yang dibutuhkan untuk menyusun *perfect lagori*. *Perfect lagori* merupakan istilah bagi tumpukan kelima silinder lagori dalam urutan yang sesuai, lagori berukuran paling besar berada di

bawah dan yang berukuran lebih kecil berada pada tumpukan di atasnya.

Tumpukan lagori awal, setelah dihancurkan oleh robot pertama (R1), dapat tercecer dalam lokasi yang berjarak relatif jauh terhadap satu sama lain. Sedangkan robot kedua (R2) harus dapat mengumpulkan silinder-silinder lagori yang terlempar dan menyusunnya ulang dalam waktu sesingkat mungkin demi perolehan poin yang tinggi. Oleh karena itu, dibutuhkan suatu strategi dalam penyusunan ulang tumpukan lagori demi memenangkan perlombaan ini.

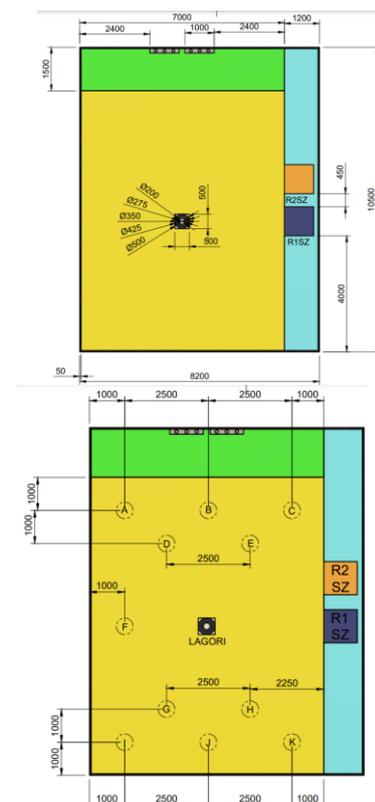


Fig. 1. Ilustrasi arena perlombaan lagori ABU Robocon 2022 (Sumber: [www.aburobocon2022.com](http://www.aburobocon2022.com))



Berikut merupakan algoritma umum dari algoritma *branch and bound*.

Algoritma Branch and Bound
<p><b>Kamus lokal:</b></p> <p><b>pQ:</b> priorityqueue diurutkan sesuai persoalan (minimasi diurutkan menaik, apabila maksimasi diurutkan menurun)</p> <p><b>root:</b> node</p>
<pre> def branchandbound(root):     if (isSolution(root) = true):         write (root)     if (all solution searched):         pQ ← root     while (pQ.isEmpty() = false):         e_node ← pQ[0]         if (isSolution(e_node) = true):             write (e_node)             if (1 solution searched):                 break         if (least-cost search):             for e1 in pQ:                 if (e1.cost &gt; e_node.cost):                     pQ.remove(e1)         else:             if (has child):                 for child in generateChild(e_node):                     child.cost = calculateCost(child)                 pQ ← child                     </pre>

### B. Permainan Lagori

Permainan lagori merupakan permainan tradisional India yang berasal dari bagian Selatan India. Permainan ini dikenal juga sebagai Seven Stones / Pittu Garam. Permainan ini dimainkan oleh dua tim yang beranggotakan minimal 3 pemain dan maksimal 9 pemain. Peralatan yang dibutuhkan dalam permainan ini adalah tujuh buah batu dan sebuah bola karet.

Tujuh buah batu ditumpuk secara vertikal di atas satu sama lain secara seimbang. Dua tim yang bermain berlawanan berusaha untuk melempar bola karet ke arah tumpukan lagori untuk menjatuhkan tumpukan tersebut. Setiap tim mendapatkan 9 kesempatan untuk melempar bola. Dalam permainan dengan 3 pemain, setiap orang akan mendapatkan 3 kali giliran melempar bola dari ketinggian sekitar 6,1 meter. Jika salah satu tim yang sedang mendapat giliran gagal untuk menjatuhkan tumpukan batu, tim lawan mendapatkan gilirannya untuk melempar bola.

Jika salah satu tim berhasil menjatuhkan seluruh tumpukan batu, tim tersebut harus menumpuk ulang ketujuh batu. Tim yang tidak menjatuhkan batu akan melempar bola, menargetkan area lutut hingga kaki tim yang sedang menumpuk ulang lagori. Pemain yang melemparkan bola tidak boleh berlari, hanya boleh berjalan dan menyampaikan bola ke temannya.

Jika tim yang menumpuk batu berhasil menumpuknya tanpa salah satu anggota pun terkena lempar bola, tim tersebut mendapatkan poin dan giliran untuk melempar pada sesi berikutnya. Jika tim yang melempar bola-lah yang berhasil mengenai area lutut hingga kaki anggota tim yang sedang menumpuk bola, tim tersebut mendapatkan giliran melempar bola pada sesi berikutnya.

### C. Permainan Lagori dalam Perlombaan ABU Robocon 2022

Permainan lagori dalam perlombaan ABU Robocon 2022 telah dimodifikasi sedemikian rupa sehingga tidak benar-benar persis dengan aturan permainan yang sebenarnya. Terdapat pula 2 skenario perlombaan, yakni perlombaan secara daring dan luring. Dalam makalah ini, akan dibahas permainan lagori untuk lomba daring.

Dikutip dari laman resmi ABU Robocon 2022, peralatan yang dibutuhkan dalam permainan secara daring adalah 5 buah disk silinder yang terbuat dari foam Polyurethane dengan diameter yang berbeda-beda sebagai lagori, 6 buah bola sepak ukuran 1, 2 buah rak bola, dan 1 buah kotak tempat menyusun tumpukan lagori.

Satu permainan akan dimainkan oleh dua tim yang bertanding secara daring. Setiap tim membuat dua buah robot, masing-masing diberi nama Robot 1 (R1) dan Robot 2 (R2). Robot 1 bertugas untuk melempar bola, baik untuk menjatuhkan lagori maupun untuk menjatuhkan *Ball on Head of Hitter* sedangkan robot 2 bertugas untuk mengambil bola, menyampaikannya pada robot 1, dan menumpuk ulang lagori yang telah dijatuhkan robot 1. Kedua tim kemudian akan disebut sebagai tim merah dan tim biru.

Pada awal permainan, 5 buah silinder lagori sudah ditumpuk di tengah lapangan. R1 tim merah akan menembakkan bola untuk menjatuhkan tumpukan lagori di tengah lapangan, durasi yang diberikan adalah 15 detik. Setelah 15 detik berakhir atau seluruh lagori telah jatuh, R2 tim merah berusaha untuk menumpuk ulang seluruh lagori selama 45 detik, sembari R1 tim biru berusaha menembak bola untuk menjatuhkan 4 buah *Ball on Head of Hitter*, yang ditandai sebagai abjad A-K di lapangan. *Ball on Head of Hitter* mana yang akan ditembak oleh R1 tim biru akan ditentukan oleh tim merah. Apabila tim biru telah selesai menjatuhkan *Ball on Head of Hitter*, maka waktu yang diberikan kepada tim merah untuk menumpuk ulang lagori juga habis.

Setelah babak pertama berakhir, seluruh lagori Kembali ditumpuk seperti semula dan permainan akan dilanjut untuk babak kedua. Permainan dengan alur yang sama seperti sebelumnya akan diulang kembali dengan menukar tim biru dengan tim merah dan tim merah dengan tim biru.

Kemenangan akan ditentukan berdasar skor dan penentu kemenangan. Untuk setiap silinder lagori yang berhasil dijatuhkan, tim yang bersangkutan mendapat 5 poin. Untuk setiap silinder lagori yang berhasil ditumpuk ulang, tim yang bersangkutan mendapat 10 poin. Total poin maksimal yang dapat diperoleh setiap tim adalah 75. Pemenang merupakan tim yang mendapatkan skor paling tinggi. Apabila terjadi seri, pemenang akan ditentukan berdasarkan penentu kemenangan, yakni:

- 1) Waktu penyusunan kembali tumpukan lagori,
- 2) Jumlah *Ball on Head of Hitter* yang berhasil dijatuhkan,
- 3) Tim dengan skor penumpukan lagori tertinggi, atau
- 4) Ditentukan juri.

### III. PERENCANAAN RUTE PENGAMBILAN SILINDER LAGORI MENGGUNAKAN ALGORITMA BRANCH AND BOUND

#### A. Deskripsi Persoalan

Dalam penerapan ini, lapangan dimodelkan dalam petak-petak lokasi pergerakan. *Ball on Head of Hitter* (BOHH) digambarkan sebagai petak abu-abu dengan anotasi huruf A–K. Lagori yang kelimanya berhasil dijatuhkan telah tercecer pada lokasi petak berwarna kuning dengan anotasi angka 1 hingga 5. Angka 1 menandakan lagori yang berukuran paling kecil dan angka 5 merupakan lagori berukuran paling besar.

Robot yang dibuat hanya dapat bergerak dalam arah horizontal atau vertikal, tidak dapat bergerak secara diagonal. Dalam setiap langkah, robot dapat bergerak ke arah Utara, Timur, Selatan, dan Barat.

Untuk memenangkan perlombaan, dibutuhkan skor yang tinggi. Untuk mendapat skor yang tinggi, seluruh lagori harus berhasil ditumpuk ulang. Oleh karena itu, kelima lagori akan ditargetkan untuk dapat seluruhnya diambil dan ditumpuk dalam urutan yang sesuai. Sehingga, strategi yang digunakan dalam penerapan ini adalah dengan mengambil lagori berukuran paling kecil terlebih dahulu, menumpuknya di atas lagori yang berukuran lebih besar dan seterusnya hingga lagori nomor 1 sampai 4 tertumpuk di atas lagori nomor 5. Seluruh lagori yang telah ditumpuk kemudian ditumpukkan di atas *lagori base* yang terdapat pada pusat lapangan.

Dalam implementasi ini, penulis mengambil data contoh skenario permainan yang mungkin terjadi, yang ditampilkan dalam gambar berikut.

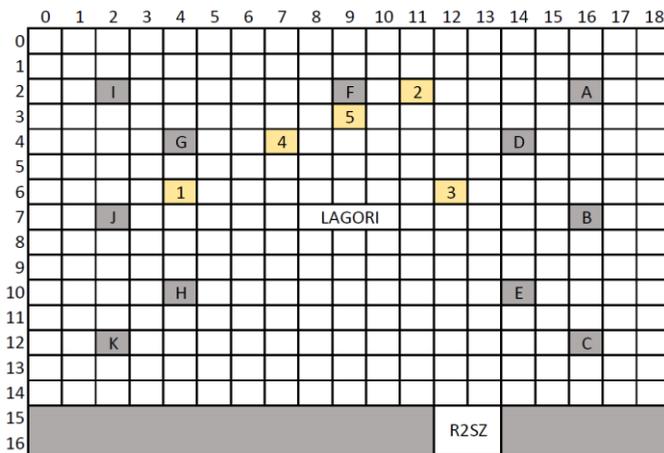


Fig. 4. Salah satu contoh skenario persoalan permainan (Sumber: dok. pribadi)

#### B. Algoritma

Berikut merupakan algoritma yang digunakan dalam pembuatan strategi penyusunan lagori ini.

```

Algoritma Branch and Bound
Penyusunan Strategi Pergerakan Robot dalam Permainan Lagori

Kamus:
    dict: dictionary (key merupakan lokasi poin yang pernah dilewati robot.
    rootNode: node akar
    eNode: expand node
    pq: priority queue yang terurut menaik
    ctrNoNode: jumlah simpul yang dibangkitkan
    field: matriks lapangan
    lagoriPositions: posisi lagori yang terserak di lapangan
    noLagori: kode lagori yang sedang diproses

def solve():
    read(lagoriPositions)
    noLagori ← 0
    field ← defaultField(lagoriPositions)
    while (noLagori < 5):
        rootNode ← node()
        dict[lokasi rootNode] = 0
        pq ← rootNode
        while (pq.isEmpty() = false):
            eNode ← pq[0]
            if (goalReached(eNode)):
                write(node)
                field[posisi X lagori][posisi Y lagori] = 1
                noLagori++
            for i in range(4):
                if (cekPergerakan()):
                    childNode ← deriveNode()
                    if (childNode valid):
                        pq ← childNode
                        dict[lokasi childNode] = calcCost(childNode)
                    ctrNoNode++

```

Berikut merupakan penjelasan secara umum algoritma yang digunakan.

- Dibuat sebuah simpul yang menyatakan status awal permainan di lapangan. Simpul ini dimasukkan ke dalam sebuah *priority queue* yang mengurutkan simpul berdasarkan estimasi biaya dari simpul akar hingga simpul solusi secara menaik.
- Selama masih terdapat simpul dalam *priority queue*, dilakukan pemrosesan sebagai berikut untuk setiap simpul yang ada:
  - Simpul yang memiliki nilai estimasi biaya terkecil diambil sebagai simpul *expand*.
  - Cek apakah simpul *expand* tersebut merupakan simpul solusi. Sebuah simpul dikatakan sebagai simpul solusi jika salah satu sisi robot bersebelahan dengan lagori.
  - Bila simpul tersebut merupakan simpul solusi, dilakukan penulisan ke layar. Pencarian berhenti

- o Bila simpul tersebut bukan merupakan simpul solusi, dibangkitkan simpul anak dengan menggeser posisi robot ke seluruh arah yang mungkin, meliputi arah Utara, Timur, Selatan, dan Barat, dapat disebut juga arah maju, kanan, kiri, dan mundur. Dihitung biaya dari simpul anak. Perhitungan memiliki 2 pilihan, yakni dengan *Euclidean Distance* atau *Manhattan Distance*.
- o Simpul anak yang dibangkitkan perlu dicek validitasnya, yakni apakah robot dapat bergerak ke arah tersebut. Untuk optimasi, pergerakan yang dilakukan tidak boleh saling menihilkan dan tidak boleh kembali ke lokasi yang pernah dilewati.
- o Jika simpul valid dan memenuhi kriteria optimasi, simpul dimasukkan ke dalam *priority queue*.

- Langkah kedua diulang untuk seluruh lagori yang hendak diambil
- Apabila seluruh lagori telah diambil dan ditumpuk pada *lagori base* yang berada di tengah lapangan, ditampilkan solusi langkah pergerakan.

Dalam penghitungan estimasi biaya dari simpul akar menuju simpul solusi, digunakan dua algoritma, yakni *Euclidean Distance* dan *Manhattan Distance*. *Euclidean Distance* menghitung jarak antara dua titik apabila dihubungkan dengan garis lurus. Berikut merupakan ilustrasi dan formula *Euclidean Distance*,

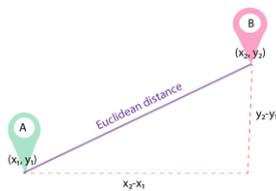


Fig. 5. Ilustrasi *Euclidean Distance* (Sumber: medium.com)

$$d = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

Fig. 6. Formula *Euclidean Distance* (Sumber: dok. pribadi)

*Manhattan Distance* mengukur jarak antara dua titik dengan jalur berpetak. Berikut merupakan ilustrasi dan formula *Manhattan Distance*,

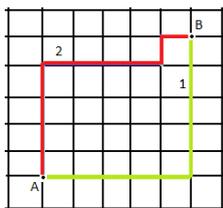


Fig. 7. Ilustrasi *Manhattan Distance* (Sumber: medium.com)

$$d = \sum_{i=1}^n |x_i - y_i| \quad (2)$$

Fig. 8. Formula *Manhattan Distance* (Sumber: dok. pribadi)

Akan dibandingkan performa dari penggunaan kedua metode estimasi biaya tersebut dalam bagian pengujian.

### C. Pengujian

Pengujian pada kasus uji sesuai skenario yang telah dibuat sebelumnya dilakukan 2 kali, masing-masing untuk penghitungan biaya dengan *Euclidean Distance* dan *Manhattan Distance*. Berikut merupakan tampilan hasil pengujian.

```

Input lagori 1 X : 6
Input lagori 1 Y : 4
Input lagori 2 X : 2
Input lagori 2 Y : 11
Input lagori 3 X : 6
Input lagori 3 Y : 12
Input lagori 4 X : 4
Input lagori 4 Y : 7
Input lagori 5 X : 3
Input lagori 5 Y : 9
[[6, 4], [2, 11], [6, 12], [4, 7], [3, 9]]

```

Fig. 9. Tampilan layar memasukkan titik-titik kasus uji (Sumber: dok. pribadi)

Berikut merupakan tampilan ilustrasi lapangan perlombaan. Angka 0 menunjukkan petak-petak yang tidak dapat dilalui oleh robot, baik itu merupakan pembatas lapangan maupun BOHH. Angka 1 menunjukkan petak-petak yang dapat dilewati oleh robot. Sedangkan angka -1 menunjukkan petak-petak yang menjadi tujuan pergerakan robot, yang terdiri atas kelima silinder lagori dan *lagori base*.

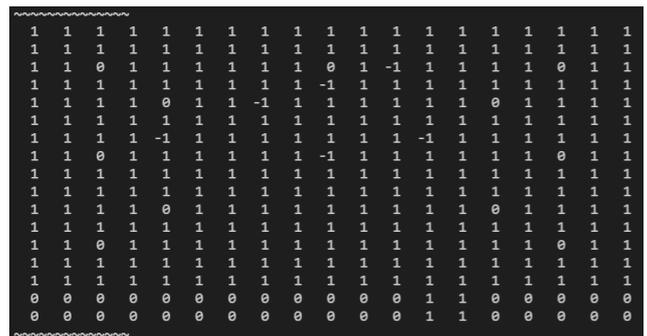


Fig. 10. Ilustrasi lapangan perlombaan (Sumber: dok. pribadi)

Pengujian pertama dilakukan terhadap *Manhattan Distance*. Berikut merupakan hasil pengujian yang dilakukan.

```

Depth : 16
FRONT FRONT LEFT LEFT LEFT LEFT FRONT LEFT LEFT FRONT FRONT LEFT FRONT FRONT LEFT FRONT
Number of nodes generated: 198
Press any key to continue...

```

Fig. 11. Hasil pengujian lintasan R2SZ ke Lagori 1 (Sumber: dok. pribadi)

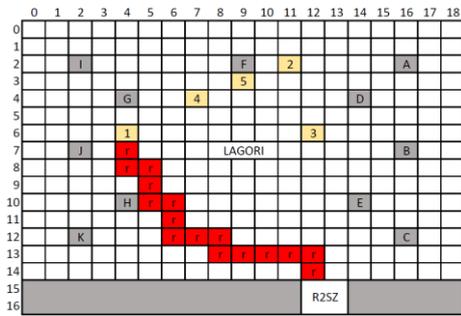


Fig. 12. Ilustrasi hasil pengujian lintasan R2SZ ke Lagori 1 (Sumber: dok. pribadi)

Hasil pengujian yang didapat juga diilustrasikan melalui gambar di atas agar dapat memperjelas pengamatan. Dalam perjalanan dari R2SZ menuju lagori pertama, didapatkan hasil yang optimum, yakni 16 langkah.

```
Depth : 11
FRONT RIGHT RIGHT FRONT RIGHT RIGHT RIGHT RIGHT FRONT RIGHT FRONT
Number of nodes generated: 66
Press any key to continue...
```

Fig. 13. Hasil pengujian Lagori 1 ke Lagori 2 (Sumber: dok. pribadi)

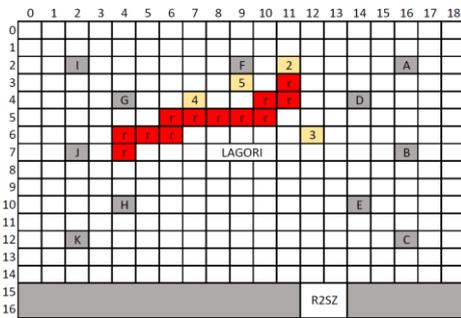


Fig. 14. Ilustrasi hasil pengujian Lagori 1 ke Lagori 2 (Sumber: dok. pribadi)

Pengujian untuk lintasan dari Lagori 1 menuju Lagori 2 juga sudah menunjukkan hasil yang optimal, yakni 11 langkah.

```
Depth : 7
BACK LEFT BACK BACK BACK RIGHT RIGHT
Number of nodes generated: 67
Press any key to continue...
```

Fig. 15. Hasil pengujian Lagori 2 ke Lagori 3 (Sumber: dok. pribadi)

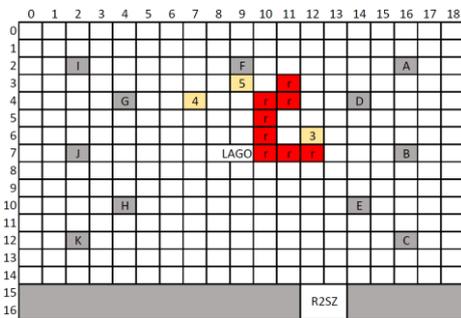


Fig. 16. Ilustrasi hasil pengujian Lagori 2 ke Lagori 3 (Sumber: dok. pribadi)

Pengujian lintasan Lagori 2 ke 3 secara sekilas tampak optimal hanya membutuhkan 3 langkah. Namun, dalam implementasi yang dibuat, hasil yang ditunjukkan sudah optimal karena robot tidak melakukan rotasi. Sisi yang dikalkulasi untuk bersampingan dengan lagori hanya sisi depan dan kiri.

```
Depth : 7
FRONT FRONT LEFT FRONT LEFT LEFT LEFT
Number of nodes generated: 42
Press any key to continue...
```

Fig. 17. Hasil pengujian Lagori 3 ke Lagori 4 (Sumber: dok. pribadi)

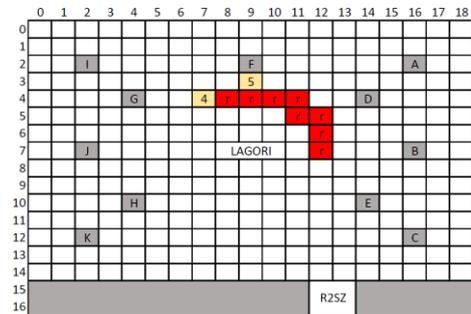


Fig. 18. Ilustrasi hasil pengujian Lagori 3 ke Lagori 4 (Sumber: dok. pribadi)

Pengujian untuk lintasan dari lagori 3 ke 4 juga mendapatkan hasil yang optimal, yakni 7 langkah.

```
Depth : 1
RIGHT
Number of nodes generated: 4
Press any key to continue...
```

Fig. 19. Hasil pengujian Lagori 4 ke Lagori 5 (Sumber: dok. pribadi)

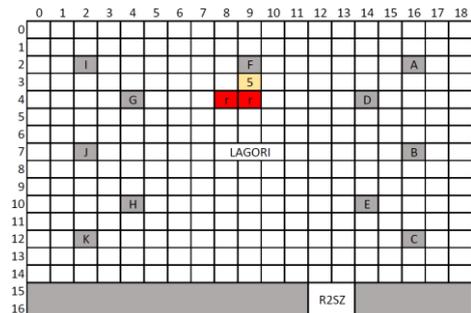


Fig. 20. Ilustrasi hasil pengujian Lagori 4 ke Lagori 5 (Sumber: dok. pribadi)

Pengujian untuk pergerakan dari lagori 4 ke 5 juga sudah optimal, hanya perlu bergeser 1 langkah ke kanan.

```
Depth : 4
BACK RIGHT BACK BACK
Number of nodes generated: 27
Press any key to continue...
```

Fig. 21. Hasil pengujian Lagori 5 ke Lagori base (Sumber: dok. pribadi)

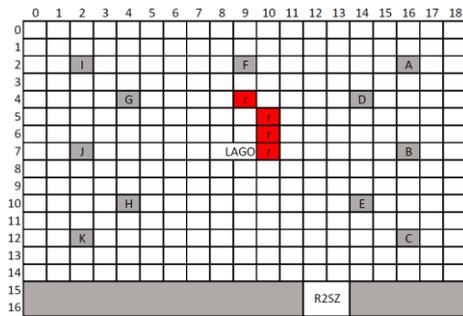


Fig. 22. Ilustrasi hasil pengujian Lagori 5 ke Lagori base (Sumber: dok. pribadi)

Pengujian untuk lintasan dari lagori 5 ke lagori base sekali lagi nampak tidak optimal karena seharusnya dapat ditempuh dalam 2 langkah. Namun sekali lagi, tidak diperhitungkan adanya rotasi robot dalam hal ini, sehingga hanya sisi kiri dan depan robot-lah yang diperhitungkan.

Selanjutnya, dilakukan pengujian terhadap kasus dimana biaya dihitung menggunakan *Manhattan Distance*. Berikut merupakan tangkapan layar hasil pengujian untuk kasus yang sama.

```

Depth : 16
FRONT FRONT FRONT LEFT FRONT FRONT LEFT LEFT LEFT FRONT FRONT LEFT FRONT LEFT LEFT LEFT
Number of nodes generated: 19378
Press any key to continue...
Depth : 11
RIGHT RIGHT RIGHT FRONT RIGHT FRONT FRONT RIGHT RIGHT FRONT RIGHT
Number of nodes generated: 383
Press any key to continue...
Depth : 7
BACK LEFT BACK BACK BACK RIGHT RIGHT
Number of nodes generated: 68
Press any key to continue...
Depth : 7
FRONT FRONT LEFT LEFT LEFT LEFT LEFT
Number of nodes generated: 83
Press any key to continue...
Depth : 3
FRONT RIGHT RIGHT
Number of nodes generated: 18
Press any key to continue...
Depth : 4
BACK RIGHT BACK BACK
Number of nodes generated: 38

```

Fig. 23. Hasil pengujian algoritma dengan perhitungan biaya menggunakan *Euclidean Distance* (Sumber: dok. pribadi)

Pada hasil tangkapan layar, ditunjukkan bahwa hasil dan kedalaman pencarian sama dengan perhitungan menggunakan *Manhattan Distance*. Perbedaan diantara kedua metode perhitungan ini adalah banyaknya simpul yang dibangkitkan. Pada perhitungan menggunakan *Euclidean Distance*, jumlah simpul yang dibangkitkan mencapai 52,4 kali jumlah simpul yang dibangkitkan dalam pengujian menggunakan *Manhattan Distance*. Hal ini dikarenakan pergerakan robot yang lebih sesuai dengan metode perhitungan *Manhattan Distance* sehingga metode perhitungan tersebut menjadi metode heuristik yang lebih tepat untuk mengestimasi nilai batas dalam algoritma *branch and bound*.

#### IV. PENUTUP

##### A. Kesimpulan

Algoritma *branch and bound* dapat memberikan hasil yang optimal apabila perhitungan nilai batas yang digunakan tepat. Dalam penyusunan strategi guna memenangkan permainan

lagori melalui minimasi jarak yang harus ditempuh dalam pengambilan lagori ini, algoritma *branch and bound* sudah dapat bekerja dengan baik.

Implementasi ini tidak terbatas hanya untuk penyusunan strategi dalam permainan lagori, tetapi dapat juga digunakan untuk persoalan-persoalan optimasi lain yang data masukannya berupa titik-titik yang dapat, tidak dapat, dan harus dilalui.

Algoritma yang dibuat sangat bergantung pada manufaktur fisik dan fungsional robot. Seperti contohnya, dalam makalah ini, robot tidak dapat berotasi, sehingga lagori hanya dapat diambil dari sisi depan dan kiri dari robot. Semakin baik manufaktur fungsionalitas robot, asumsi yang digunakan dalam pembuatan algoritma juga akan membaik, sehingga hasil yang didapatkan akan lebih optimal.

##### B. Saran

Kasus uji yang digunakan dalam pengujian algoritma yang digunakan hanya sedikit. Oleh karena itu, untuk keperluan pengamatan guna optimasi, data kasus uji dapat diperbanyak sehingga pengujian dapat dilakukan secara menyeluruh sehingga hasil strategi yang dihasilkan akan lebih baik pula.

#### UCAPAN TERIMA KASIH

Ucapan terima kasih yang sebesar-besarnya hendak saya sampaikan pada Tuhan Yang Maha Esa karena atas berkat-Nya lah saya berhasil menyelesaikan penulisan makalah ini. Tidak lupa ucapan terima kasih saya sampaikan kepada keluarga saya yang telah mendukung dalam proses pendidikan dan perkuliahan yang mana makalah ini merupakan salah satu bagian dari proses tersebut. Tentu saja makalah ini tak dapat selesai tanpa pengajaran dan dedikasi dari dosen-dosen pengampu mata kuliah Strategi Algoritma Semester 2 2021/2022, Ibu Masayu Leylia Khodra, Ibu Nur Ulfa Maulidevi, dan Bapak Rinaldi Munir. Akhir kata, saya sampaikan terima kasih kepada seluruh pihak yang telah membantu dan mendukung selesainya makalah ini, tak terkecuali para pengembang bahasa pemrograman Python dan berbagai dokumentasi yang diperlukan.

#### REFERENCES

- [1] ABU Robocon 2022 Online Rulebook. Diakses melalui [www.aburobocon2022.com](http://www.aburobocon2022.com) pada 20 Mei 2022.
- [2] Gohrani, Kumal. 2019. Different Types of Distance Metrics used in Machine Learning. Diakses melalui [https://medium.com/@kunal\\_gohrani/different-types-of-distance-metrics-used-in-machine-learning-e9928c5e26c7](https://medium.com/@kunal_gohrani/different-types-of-distance-metrics-used-in-machine-learning-e9928c5e26c7) pada 22 Mei 2022.
- [3] Levitin, A. 2012. Introduction to The Design and Analysis of Algorithms (Third Edition). Edinburgh: Pearson.
- [4] Robert Wood, "About Lagori." Topend Sports Website, January 2016, <https://www.topendsports.com/sport/list/lagori.htm>, diakses pada 21 Mei 2022.
- [5] Shokry, Sherif & Tanaka, Shinji & Nakamura, Fumihiko & Ariyoshi, Ryo & Miura, Shino. 2018. Bandwidth Maximization Approach for Displaced Left-Turn Crossovers Coordination under Heterogeneous Traffic Conditions. Journal of Traffic and Transportation Engineering. 6. 183-196. 10.17265/2328-2142/2018.04.004. Diakses melalui [https://www.researchgate.net/figure/A-diagram-explains-the-main-concept-of-branch-and-bound-algorithm\\_fig2\\_327414612](https://www.researchgate.net/figure/A-diagram-explains-the-main-concept-of-branch-and-bound-algorithm_fig2_327414612) pada 22 Mei 2022.

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 20 Mei 2022



Maharani Ayu Putri Irawan