

## **Analisis Performansi Algoritma *Informed Rapidly-exploring Random Tree\** dengan Metode *Boundary Sampling* dan Metode *Goal Biasing Sampling***

### ***Performance Analysis of Informed Rapidly-Exploring Random Tree\* Algorithm using Boundary Sampling Method and Goal Biasing Sampling Method***

**Kamilia<sup>\*</sup>, Tri Rahajoeningroem**

Program Studi Teknik Elektro, Fakultas Teknik dan Ilmu Komputer

Universitas Komputer Indonesia Jl. Dipati ukur No 112, Bandung

\*Email : [kamilia.13118802@mahasiswa.unikom.ac.id](mailto:kamilia.13118802@mahasiswa.unikom.ac.id)

**Abstrak** - Perencanaan jalur merupakan salah satu program penunjang untuk robotika atau kendaraan. Pada perencanaan jalur dibutuhkan algoritma yang tepat agar mendapatkan hasil yang optimal. Algoritma *Informed Rapidly-exploring Random Tree\** (*Informed-RRT\**) merupakan algoritma untuk perencanaan jalur yang berbasis *sampling* atau pengambilan sampel acak pada ruang pencarian. Pada penelitian ini akan dilakukan analisis performansi algoritma *Informed-RRT\** dengan metode *boundary sampling* dan *Informed-RRT\** metode *goal biasing sampling*. Tujuan penelitian ini yaitu membandingkan kedua algoritma tersebut untuk mendapatkan algoritma dengan performansi terbaik dalam melakukan perencanaan jalur. Pengujian dilakukan berbasis simulasi menggunakan *software* Labview. Dalam pengujian ini digunakan beberapa kasus lingkungan *obstacle* yang ada, yaitu lingkungan *narrow*, *trap* dan *clutter*. Kriteria untuk menentukan performansi yang optimal yaitu jarak jalur (*path cost*), waktu komputasi, dan jumlah node. Dari hasil yang didapatkan, nilai rata-rata algoritma *Informed-RRT\** metode *boundary sampling* menghasilkan waktu komputasi yang lebih cepat dan juga jumlah node yang lebih sedikit dibandingkan dengan *Informed-RRT\** metode *goal biasing sampling*, sedangkan untuk jarak jalur algoritma dengan kedua metode tersebut memiliki hasil yang hampir sama. Dapat disimpulkan bahwa algoritma *Informed-RRT\** metode *boundary sampling* memiliki performansi lebih optimal dibandingkan dengan algoritma *Informed-RRT\** metode *goal biasing sampling*.

**Kata kunci** : *Boundary Sampling*, *Goal Biasing Sampling*, *Informed-RRT\**, Labview, Perencanaan jalur

**Abstract** - *Path planning is one of the supporting programs for robotics or vehicles. In path planning, the right algorithm is needed in order to get optimal results. Informed Rapidly-exploring Random Tree\* (Informed-RRT\*) algorithm is an algorithm for path planning based on random sampling in the search space. In this study, an analysis of the performance of the Informed-RRT\* algorithm will be carried out using the boundary sampling method and the goal biasing sampling method. The purpose of this study is to compare the two algorithms to get the algorithm with the best performance in planning paths. The test is based on simulation using Labview software. In this test, several cases of existing obstacle environments are used, namely narrow, trap and clutter. The criteria for determining optimal performance are path cost, computation time, and number of nodes. From the results obtained, the average value of the Informed-RRT\* boundary sampling method produces faster computation time and also fewer nodes than the Informed-RRT\* goal biasing sampling method, while for the distance of the algorithm path with both methods have almost the same result. It can be concluded that the Informed-RRT\* boundary sampling method has more optimal performance than the Informed-RRT\* goal biasing sampling method.*

**Keyword** : *Boundary Sampling*, *Goal Biasing Sampling*, *Informed-RRT\**, Labview, Path Planning

## **I. PENDAHULUAN**

### **A. Latar Belakang**

Perencanaan jalur atau *path planning* telah banyak diteliti dan dikembangkan untuk algoritma pada program kendaraan atau robotika. Tujuan dari

perencanaan jalur adalah merancang jalur dari posisi awal ke posisi tujuan pada lingkungan yang memiliki hambatan statis maupun dinamis [1]. Pada perencanaan jalur dibutuhkan algoritma yang tepat agar mendapatkan hasil yang optimal.

Algoritma perencanaan jalur memungkinkan robot atau kendaraan menemukan jalur terpendek atau jalur optimal antara dua titik. Jalur optimal dapat berupa jalur yang meminimalkan jumlah belokan, jumlah pengereman, jumlah tabrakan, atau apa pun yang diperlukan dalam program tersebut [2]. Terdapat beberapa metode yang digunakan untuk melakukan perencanaan jalur ini, diantaranya yaitu metode perencanaan jalur berbasis sensor, perencanaan jalur berbasis *graph*, dan perencanaan jalur berbasis pengambilan sample acak [3].

## B. Tinjauan *State of Art*

Pada metode perencanaan jalur berbasis sensor, robot bekerja di lingkungan yang belum dipetakan. Robot mengumpulkan berbagai informasi mengenai lingkungan atau area yang akan dilewati, lalu dari informasi itu robot dapat bergerak langkah demi langkah. Lalu terdapat juga perencanaan jalur berbasis *graph* yang terbagi lagi menjadi beberapa bagian, yaitu algoritma perencanaan jalur *Dijkstra* [4], *A\** [5], *D\** [6] dan *AD\** [7]. Algoritma tersebut termasuk dalam perencanaan jalur berbasis *graph* yang paling banyak digunakan. Namun, algoritma-algoritma tersebut masih memiliki beberapa kekurangan. Seperti algoritma *AD\** yang hanya bisa diterapkan pada ruang kerja yang telah didiskritisasi, yang akan menyebabkan terjadinya penurunan kinerja dalam kasus dimensi yang tinggi.

Metode perencanaan jalur lainnya yang sering digunakan adalah algoritma *Sampling Based Planning* (SBP). Algoritma SBP menjadi salah satu metode perencanaan jalur yang dapat memberikan solusi paling optimal dibandingkan dengan metode lainnya. Algoritma SBP menggunakan pengambilan sampel acak pada ruang pencarian, dan juga mempunyai keunggulan yaitu dapat memberikan solusi yang cepat pada masalah yang kompleks dan memiliki dimensi yang tinggi [8]. Algoritma *Probabilistic Roadmap Method* (PRM) [9] dan algoritma *Rapidly-exploring Random Tree* (RRT) [10] merupakan algoritma bagian dari SBP yang paling umum digunakan. Algoritma PRM memiliki keunggulan dapat memberikan solusi yang bersifat asimptotik optimal, yaitu algoritma PRM dapat memberikan solusi yang optimal jika waktu yang diberikan mencukupi. Tetapi algoritma PRM memiliki waktu komputasi yang cukup tinggi. Algoritma RRT yang diusulkan oleh Lavelle [10] berhasil menerapkan algoritma RRT pada masalah yang kompleks. Keunggulan algoritma RRT lainnya adalah waktu komputasi yang rendah, tetapi hanya dapat memberikan solusi yang bersifat kurang optimal

[8]. Hal tersebut yang menyebabkan algoritma RRT kemudian dikembangkan menjadi *Rapidly-exploring Random Tree\** (RRT\*). Karaman dan Frazzoli [11] mengusulkan variasi dari algoritma RRT yang kemudian diberi nama RRT\*. Algoritma RRT\* dapat memberikan kualitas keluaran yang bersifat asimptotik optimal dengan menggunakan dua fitur tambahan, yaitu *ChooseParent* dan *Rewire*. Tetapi, waktu komputasi dari algoritma RRT\* masih tergolong cukup tinggi.

Gammell dkk. [2] mengusulkan algoritma *Informed-RRT\**. Algoritma *Informed-RRT\** bekerja dengan cara membatasi wilayah pengambilan sampel baru (*sampling*) agar lebih efisien. Proses pengambilan sampel pada algoritma *Informed-RRT\** dilakukan dalam wilayah elips yang mengelilingi *start node* dan *goal node*. Teknik pengambilan sampel pada algoritma *informed-RRT\** ini diklaim dapat mempercepat laju konvergensi dari algoritma RRT\* [2].

Pada penelitian ini digunakan metode *sampling* yang berbeda dari algoritma *Informed-RRT\** pada umumnya. Terdapat beberapa metode *sampling* yang dapat digunakan pada algoritma *Informed-RRT\**, beberapa diantaranya adalah metode *boundary sampling* dan *goal biasing sampling*. Metode *boundary sampling* adalah metode *sampling* yang melakukan pencarian jalur dengan cara membangkitkan node pada batasan-batasan *obstacle* [12], sedangkan metode *goal biasing* adalah metode *sampling* yang bekerja dengan cara mengambil sampel acak pada ruang pencarian, kemudian sesekali akan mengambil titik sampel langsung disekitar titik tujuan [13]. Pada pengujian ini dilakukan perbandingan algoritma *Informed-RRT\** menggunakan metode *boundary sampling* dan metode *goal biasing sampling*.

## C. Tujuan

Tujuan dari penelitian ini yaitu mengimplementasikan sistem algoritma *Informed-RRT\** metode *boundary sampling* dan metode *goal biasing sampling* sebagai program perencanaan jalur pada labview. Selain itu juga bertujuan untuk membuat analisa performansi dan membandingkan hasil dari analisa performansi algoritma *Informed-RRT\** dengan metode *boundary sampling* dan metode *goal biasing sampling*

## D. Sistematika Pembahasan

Makalah ini diorganisasikan sebagai berikut, yaitu pada bagian 1 akan menjelaskan mengenai latar belakang, tinjauan *state of art*, tujuan, dan sistematika pembahasan. Bagian 2 akan menyajikan metodologi penelitian serta rancangan program yang telah dibuat dalam bentuk blok

diagram Labview. Program Labview ini, bisa digunakan sebagai perangkat untuk simulasi maupun dihubungkan langsung dengan perangkat keras [14]. Bagian 3 berisi tentang pemaparan dan analisis data-data yang didapatkan dari hasil pengujian, dimana data-data tersebut nantinya akan dibandingkan satu dengan yang lainnya. Adapun kesimpulan akan disajikan pada Bagian 4.

**II. METODOLOGI**

Pada penelitian ini dilakukan analisis untuk mengetahui performansi algoritma *Informed-RRT\** metode *boundary sampling* dan *Informed-RRT\** metode *goal biasing sampling*. Program untuk algoritma tersebut dibuat pada *software* Labview yang nantinya dapat memvisualisasikan proses pencarian jalur. Program dari algoritma *Informed-RRT\** ini terdiri dari beberapa subprogram, yaitu subprogram *check collision*, *check segment collision*, *choose parent*, *find near nodes*, *get path len*, *get final course*, *get nearest list index*, *get new node*, *is near goal*, *line cost*, *rewire*, *sample unit ball*, *draw graph*, *draw path*, *distance squared point to segment*, dan *boundary sampling*.

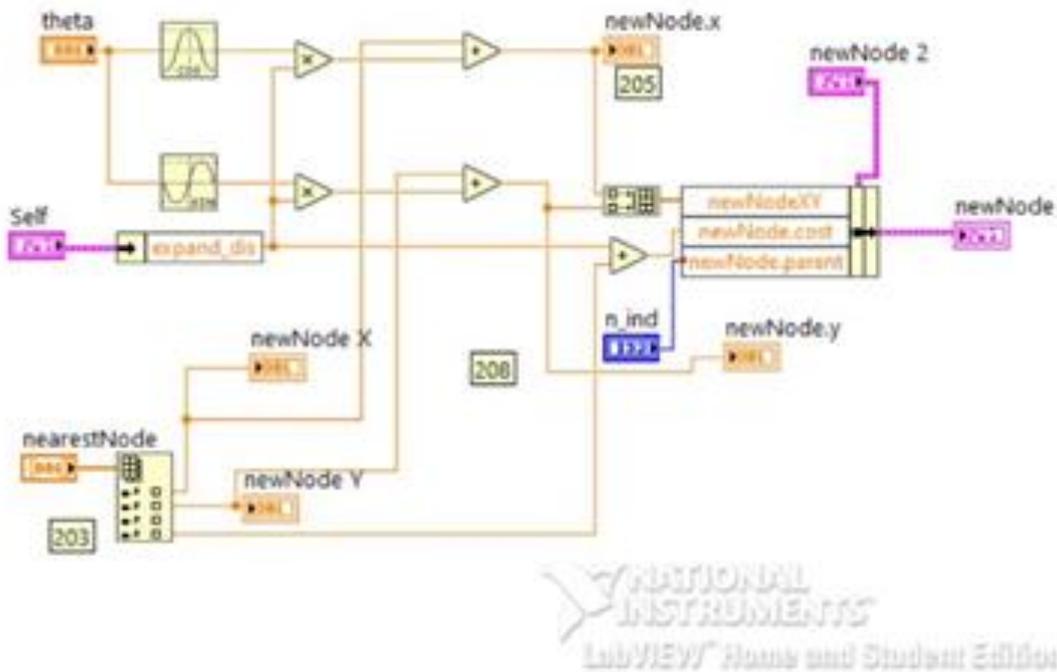
Subprogram *get new node* merupakan subprogram yang terdapat pada program *Informed-RRT\** metode *goal biasing sampling*. Subprogram tersebut berfungsi untuk mencari node-node baru yang akan dibangkitkan, dimana node-node tersebut acak pada ruang pencarian,

tetapi ada kalanya program memunculkan node-node disekitar titik tujuan. Blok diagram untuk subprogram *get new node* dapat dilihat pada **Gambar 1**.

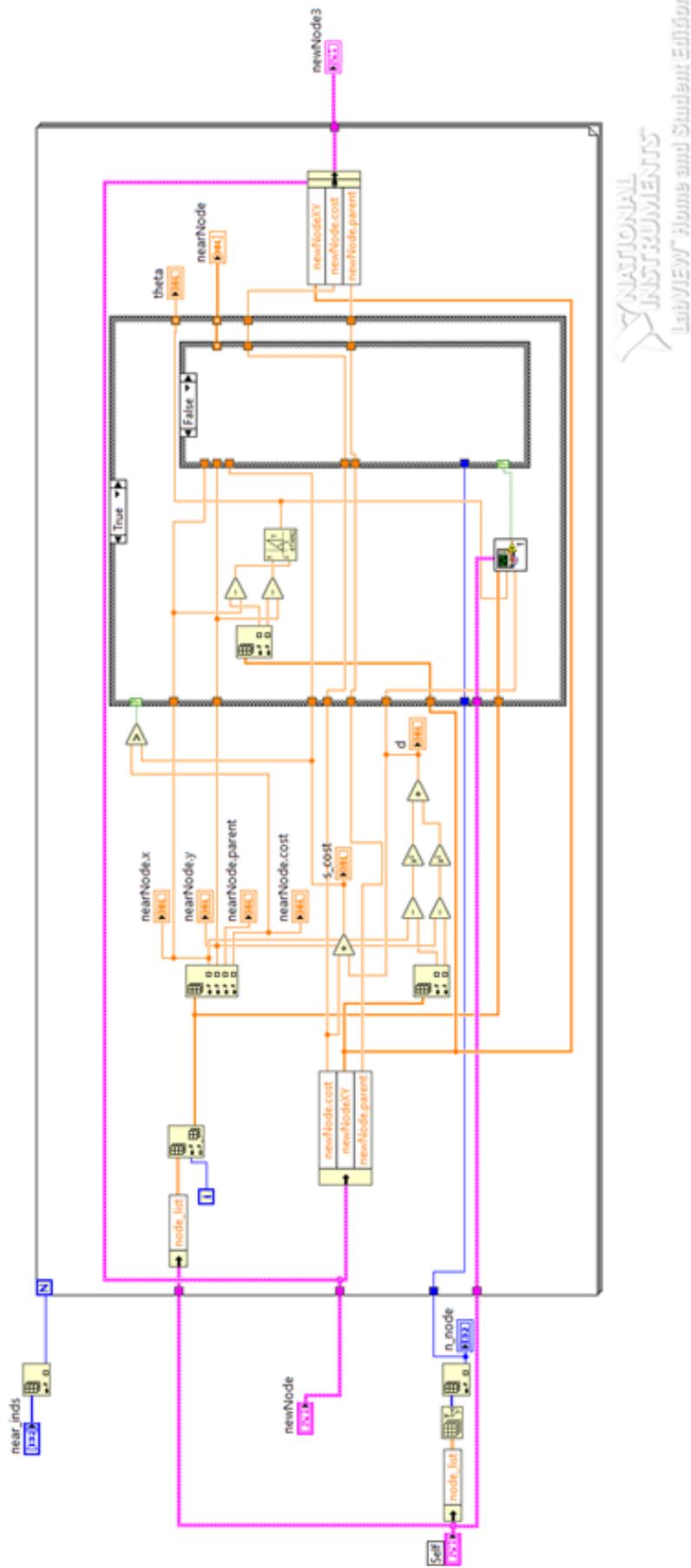
Pada subprogram *rewire* akan dicari node-node yang terdekat dengan *goal node* dimana jika node tersebut dihubungkan ke node awal akan diperoleh jalur yang lebih dekat dari jalur yang sebelumnya telah dibangkitkan. Subprogram *rewire* dapat dilihat pada **Gambar 2**.

Pada subprogram *choose parent*, dicari node-node yang berjarak terdekat dari node awal. Diantara node-node tersebut akan dicari node yang bila dihubungkan ke node awal akan diperoleh jalur paling minimal atau jalur terdekat. Node-node tersebut yang nantinya akan menjadi *node parent*. Blok diagram untuk subprogram *choose parent* dapat dilihat pada **Gambar 3**. Subprogram *rewire* dan *choose parent* termasuk pada subprogram inti diantara keseluruhan program, karena kedua subprogram tersebut yang membedakan *RRT\** dengan program *RRT* lainnya.

Subprogram *boundary sampling* merupakan subprogram untuk proses *sampling* atau pencarian node yang akan dibangkitkan. Pada subprogram ini node-node akan dibangkitkan, dimana node tersebut harus terletak pada perbatasan *obstacle* yang ada pada ruang pencarian. Blok diagram untuk subprogram *boundary sampling* dapat dilihat pada **Gambar 4**.

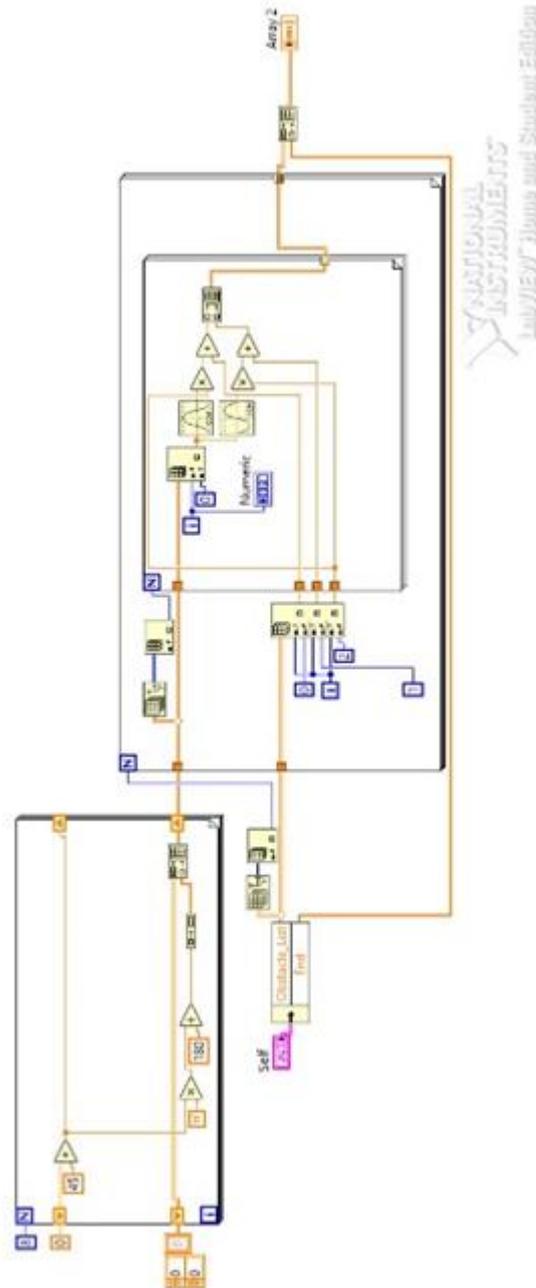


**Gambar 1.** Blok Diagram Get New Node



Gambar 2. Blok Diagram Rewire



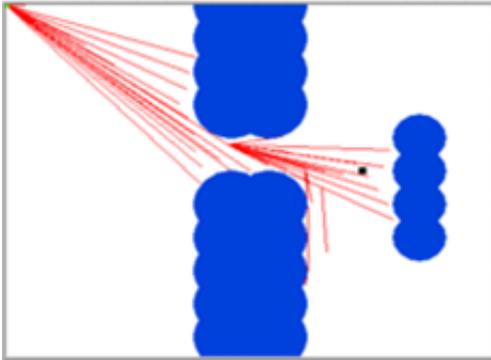


Gambar 4. Blok Diagram *Boundary Sampling*

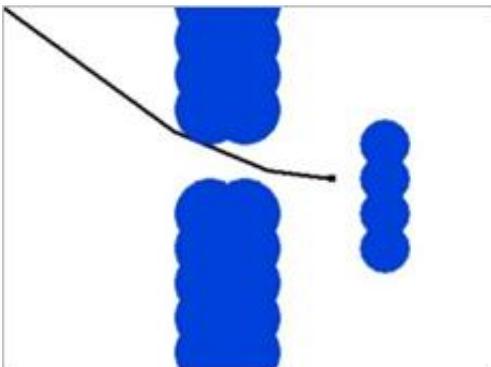
### III. HASIL DAN PEMBAHASAN

Pada pengujian ini, algoritma *Informed-RRT\** metode *boundary sampling* dan *Informed-RRT\** metode *goal biasing sampling* masing-masing dijalankan sebanyak 30 kali pada kasus lingkungan *obstacle trap*, *narrow*, dan *clutter*. Hal ini bertujuan untuk membandingkan *Informed-RRT\** metode *boundary sampling* dan *Informed-RRT\** metode *goal biasing sampling*. Data yang diambil dalam pengujian tersebut yaitu waktu komputasi, jumlah node, dan jarak jalur.

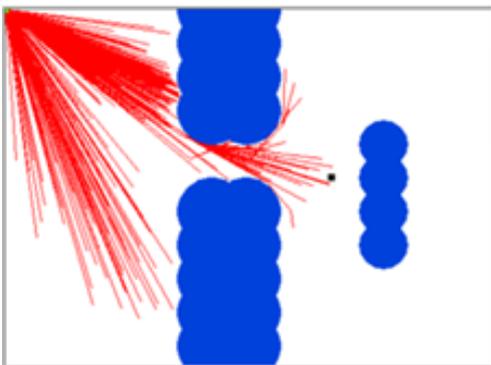
Gambar 5 dan Gambar 6 menunjukkan hasil dari program algoritma *Informed-RRT\** metode *boundary sampling* sedangkan Gambar 7 dan Gambar 8 menunjukkan algoritma *Informed-RRT\** metode *goal biasing*, dimana keduanya diuji pada lingkungan *obstacle narrow*. Lingkungan *obstacle narrow* terbentuk menjadi celah sempit, dan jalur yang dibangkitkan nantinya akan melewati celah tersebut untuk mencapai *goal node*. Dari kedua gambar tersebut terlihat bahwa kedua program berhasil menemukan solusi berupa jalur yang dibangkitkan dari *start node* menuju *goal node*.



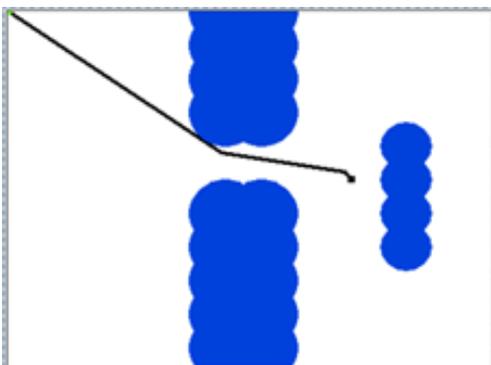
**Gambar 5.** Hasil pengujian algoritma *Informed-RRT\** metode *boundary sampling* pada lingkungan *narrow*



**Gambar 6.** Hasil jalur yang didapatkan dari pengujian algoritma *Informed-RRT\** metode *boundary sampling* pada lingkungan *narrow*

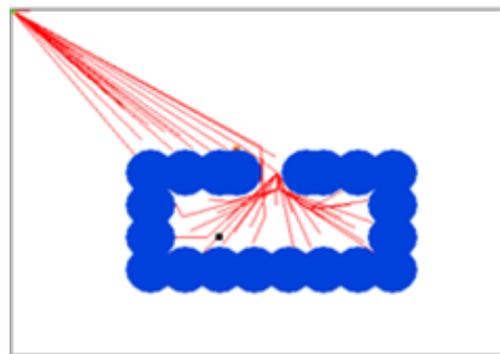


**Gambar 7.** Hasil pengujian algoritma *Informed-RRT\** metode *goal biasing sampling* pada lingkungan *narrow*

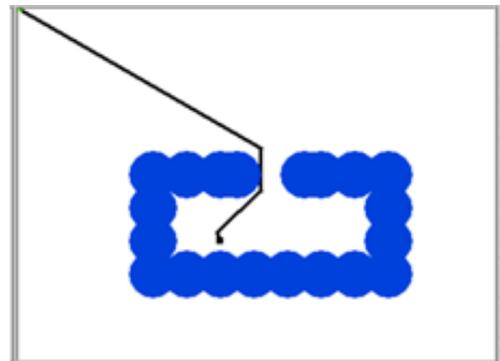


**Gambar 8.** Hasil jalur yang didapatkan dari pengujian algoritma *Informed-RRT\** metode *goal biasing sampling* pada lingkungan *narrow*

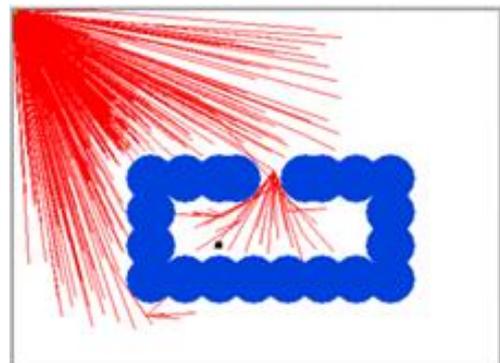
Pada **Gambar 9** dan **Gambar 10** terdapat hasil dari program algoritma *Informed-RRT\** metode *boundary sampling* pada lingkungan *obstacle trap*, sedangkan pada **Gambar 11** dan **Gambar 12** terdapat hasil dari program algoritma *Informed-RRT\** metode *goal biasing sampling* pada lingkungan *obstacle trap*. Lingkungan *obstacle trap* terbentuk menjadi sebuah perangkap, dimana *start node* harus membangkitkan jalur menuju *goal node* yang berada didalam kotak perangkap. Kedua program baik algoritma *Informed-RRT\** metode *boundary sampling* maupun *Informed-RRT\** metode *goal biasing sampling* dapat menemukan jalur dari *start node* menuju *goal node*.



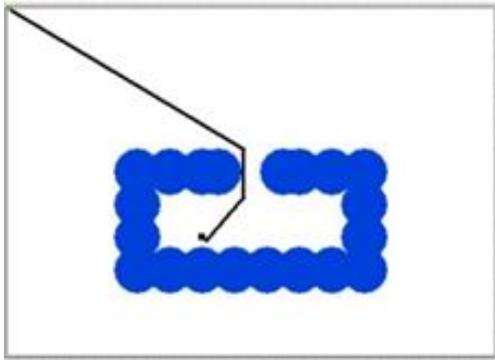
**Gambar 9.** Hasil pengujian algoritma *Informed-RRT\** metode *boundary sampling* pada lingkungan *trap*



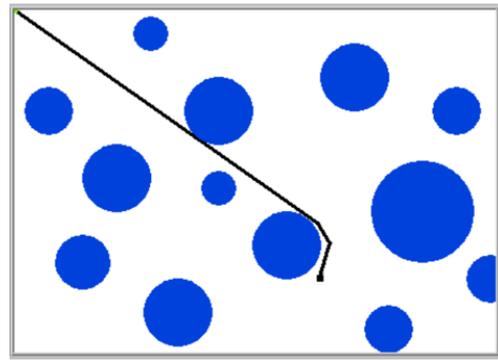
**Gambar 10.** Hasil jalur yang didapatkan dari pengujian algoritma *Informed-RRT\** metode *boundary sampling* pada lingkungan *trap*



**Gambar 11.** Hasil pengujian algoritma *Informed-RRT\** metode *goal biasing sampling* pada lingkungan *trap*

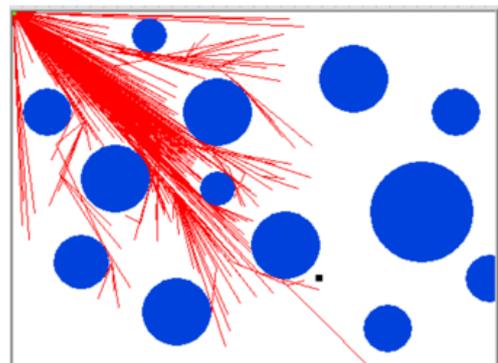


**Gambar 12.** Hasil jalur yang didapatkan dari pengujian algoritma *Informed-RRT\** metode *goal biasing sampling* pada lingkungan *trap*



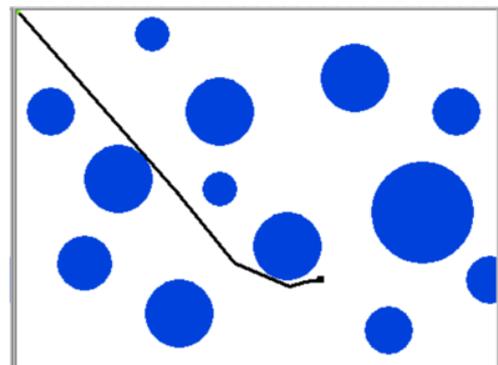
**Gambar 14.** Hasil jalur yang didapatkan dari pengujian algoritma *Informed-RRT\** metode *boundary sampling* pada lingkungan *clutter*

Pengujian terakhir dilakukan pada lingkungan *obstacle clutter*. Pada **Gambar 13** dan **Gambar 14** terdapat hasil dari program algoritma *Informed-RRT\** metode *boundary sampling* pada lingkungan *obstacle clutter*, sedangkan **Gambar 15** dan **Gambar 16** terdapat hasil dari program algoritma *Informed-RRT\** metode *goal biasing sampling* pada lingkungan *obstacle clutter*. Lingkungan *clutter* terbentuk dari beberapa *obstacle* yang berantakan dan tidak menentu. Dari kedua gambar dapat dilihat bahwa masing-masing program baik algoritma *Informed-RRT\** metode *boundary sampling* maupun *Informed-RRT\** metode *goal biasing sampling* dapat menemukan jalur dari *start node* menuju *goal node* dengan melewati *obstacle* yang terletak secara acak.

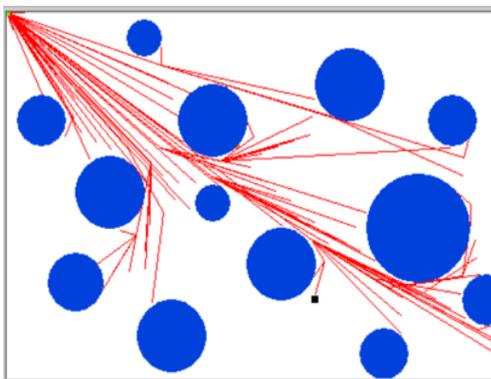


**Gambar 15.** Hasil pengujian algoritma *Informed-RRT\** metode *goal biasing sampling* pada lingkungan *clutter*

Dari keseluruhan hasil yang didapatkan, terlihat perbedaan yang cukup signifikan antara *Informed-RRT\** metode *boundary sampling* dengan *Informed-RRT\** metode *goal biasing sampling*. Pada *Informed-RRT\** metode *goal biasing sampling* banyak dimunculkan node-node di keseluruhan area pencarian [13]. Berbeda dengan *Informed-RRT\** metode *boundary sampling* yang hanya memunculkan sedikit node, dan juga node-node tersebut hanya dimunculkan di sekitar batas-batas area *obstacle* [12].



**Gambar 16.** Hasil jalur yang didapatkan dari pengujian algoritma *Informed-RRT\** metode *goal biasing sampling* pada lingkungan *clutter*



**Gambar 13.** Hasil pengujian algoritma *Informed-RRT\** metode *boundary sampling* pada lingkungan *clutter*

Data lengkap hasil dari pengujian yang berupa nilai maksimum, nilai minimum, dan nilai rata-rata terdapat pada **Tabel I** sampai dengan **Tabel IX**. Pada **Tabel I**, **Tabel IV**, dan **Tabel VII**, terdapat perbandingan jarak jalur atau *path cost* *Informed-RRT\** metode *boundary sampling* dan *Informed-RRT\** metode *goal biasing sampling*. Jarak jalur kedua algoritma tersebut tidak memiliki perbedaan yang signifikan, karena nilai dari jarak jalur yang dihasilkan hampir sama, atau tidak memiliki perbedaan yang terlalu jauh.

Pada **Tabel II**, **Tabel V**, dan **Tabel VIII** terdapat perbandingan jumlah node pada algoritma *Informed-RRT\** metode *boundary sampling* dan

*Informed-RRT\** metode *goal biasing sampling*. Kedua algoritma tersebut menghasilkan jumlah node yang jauh berbeda, dimana hasil pengujian menunjukkan bahwa jumlah node maksimum, minimum, dan rata-rata algoritma *Informed-RRT\** metode *goal biasing sampling* lebih banyak dibandingkan algoritma *Informed-RRT\** metode *boundary sampling*.

**Tabel I.** Perbandingan jarak jalur pada lingkungan *narrow*

	Jarak jalur	
	<i>Informed-RRT* Goal biasing Sampling</i>	<i>Informed-RRT* Boundary Sampling</i>
Maksimum	22.65	11.15
Minimum	10.89	10.02
Rata-rata (Mean)	11.39	10.55

**Tabel II.** Perbandingan jumlah node pada lingkungan *narrow*

	Jumlah Node	
	<i>Informed-RRT* Goal biasing Sampling</i>	<i>Informed-RRT* Boundary Sampling</i>
Maksimum	394	69
Minimum	315	50
Rata-rata (Mean)	355.83	56.87

**Tabel III.** Perbandingan waktu pada lingkungan *narrow*

	Waktu Komputasi	
	<i>Informed-RRT* Goal biasing Sampling</i>	<i>Informed-RRT* Boundary Sampling</i>
Maksimum	28.27	6.84
Minimum	14.89	5.17
Rata-rata (Mean)	22.30	5.74

**Tabel IV.** Perbandingan jarak jalur pada lingkungan *clutter*

	Jarak jalur	
	<i>Informed-RRT* Goal biasing Sampling</i>	<i>Informed-RRT* Boundary Sampling</i>
Maksimum	12.83	12.99
Minimum	12.50	12.50
Rata-rata (Mean)	12.66	12.73

Perbandingan waktu komputasi pada algoritma *Informed-RRT\** metode *boundary sampling* dan *Informed-RRT\** metode *goal biasing sampling* terdapat pada **Tabel III**, **Tabel VI**, dan **Tabel IX**. Algoritma *Informed-RRT\** metode *boundary sampling* memiliki waktu komputasi yang lebih optimal dibandingkan dengan algoritma *Informed-RRT\** metode *goal biasing sampling*. Hal tersebut dikarenakan *Informed-RRT\** metode *boundary sampling* lebih cepat dalam menemukan solusi atau membangkitkan jalur. Hasil pengujian menunjukkan bahwa waktu komputasi maksimum, minimum, dan rata-rata algoritma *Informed-RRT\** metode *boundary sampling* lebih cepat dibandingkan algoritma *Informed-RRT\** metode *goal biasing sampling*.

**Tabel V.** Perbandingan jumlah node pada lingkungan *clutter*

	Jumlah Node	
	<i>Informed-RRT* Goal biasing Sampling</i>	<i>Informed-RRT* Boundary Sampling</i>
Maksimum	417	117
Minimum	380	100
Rata-rata (Mean)	399.97	108.43

**Tabel VI.** Perbandingan waktu pada lingkungan *clutter*

	Waktu Komputasi	
	<i>Informed-RRT* Goal biasing Sampling</i>	<i>Informed-RRT* Boundary Sampling</i>
Maksimum	21.82	5.37
Minimum	16.95	5.11
Rata-rata (Mean)	19.02	5.22

**Tabel VII.** Perbandingan jarak jalur pada lingkungan *trap*

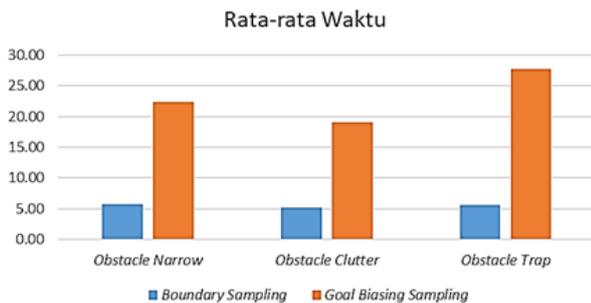
	Jarak jalur	
	<i>Informed-RRT* Goal biasing Sampling</i>	<i>Informed-RRT* Boundary Sampling</i>
Maksimum	12.24	11.89
Minimum	11.44	10.82
Rata-rata (Mean)	11.74	11.41

**Tabel VIII.** Perbandingan jumlah node pada lingkungan *trap*

	Jumlah Node	
	<i>Informed-RRT* Goal biasing Sampling</i>	<i>Informed-RRT* Boundary Sampling</i>
Maksimum	425	65
Minimum	362	51
Rata-rata (Mean)	394.13	56.17

**Tabel IX.** Perbandingan waktu pada lingkungan *trap*

	Waktu Komputasi	
	<i>Informed-RRT* Goal biasing Sampling</i>	<i>Informed-RRT* Boundary Sampling</i>
Maksimum	34.45	7.21
Minimum	21.66	5.02
Rata-rata (Mean)	27.67	5.60

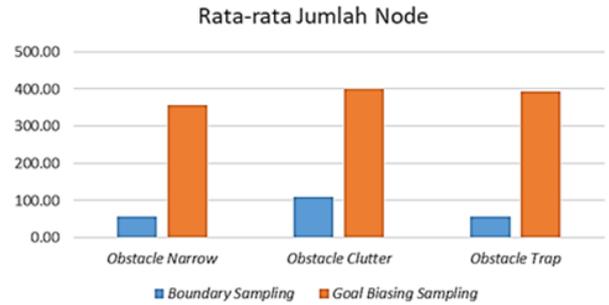


**Gambar 17.** Perbandingan rata-rata waktu komputasi algoritma *Informed-RRT\** metode *boundary sampling* dan *Informed-RRT\** metode *goal biasing sampling* pada lingkungan *obstacle narrow*, *trap*, dan *clutter*

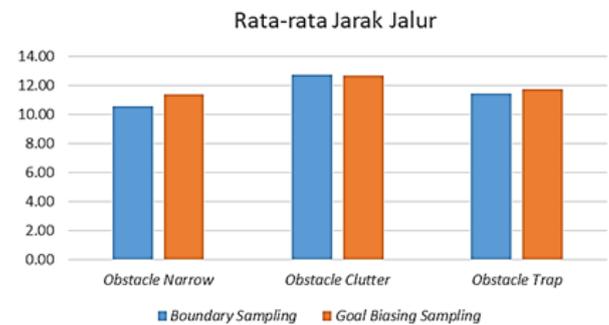
Pada **Gambar 17** terdapat grafik perbandingan rata-rata waktu komputasi algoritma *Informed-RRT\** metode *boundary sampling* dan *Informed-RRT\** metode *goal biasing sampling*. Dari grafik tersebut dapat terlihat bahwa untuk seluruh kasus lingkungan *obstacle*, algoritma *Informed-RRT\** metode *boundary sampling* memiliki performansi yang lebih optimal karena menghasilkan waktu komputasi yang lebih cepat dibandingkan dengan *Informed-RRT\** metode *goal biasing sampling*.

Grafik perbandingan rata-rata jumlah node algoritma *Informed-RRT\** metode *boundary sampling* dan *Informed-RRT\** metode *goal biasing sampling* terdapat pada **Gambar 18**. Dari grafik tersebut dapat terlihat bahwa algoritma *Informed-RRT\** metode *goal biasing sampling* memiliki jumlah node yang jauh lebih banyak dibandingkan dengan *Informed-RRT\** metode *boundary sampling*. Hal tersebut dikarenakan *Informed-*

*RRT\** metode *goal biasing sampling* membuat sampel acak pada seluruh ruang pencarian, sementara *Informed-RRT\** metode *boundary sampling* hanya memunculkan node di sekitar batasan *obstacle* [15].



**Gambar 18.** Perbandingan rata-rata jumlah node algoritma *Informed-RRT\** metode *boundary sampling* dan *Informed-RRT\** metode *goal biasing sampling* pada lingkungan *obstacle narrow*, *trap*, dan *clutter*



**Gambar 19.** Perbandingan rata-rata jarak jalur algoritma *Informed-RRT\** metode *boundary sampling* dan *Informed-RRT\** metode *goal biasing sampling* pada lingkungan *obstacle narrow*, *trap*, dan *clutter*

**Gambar 19** menunjukkan grafik perbandingan rata-rata jarak jalur atau *path cost* algoritma *Informed-RRT\** metode *boundary sampling* dan *Informed-RRT\** metode *goal biasing sampling*. Berbeda dengan rata-rata waktu komputasi dan jumlah node, hasil dari rata-rata jarak jalur pada algoritma *Informed-RRT\** metode *boundary sampling* dan *Informed-RRT\** metode *goal biasing sampling* memiliki hasil yang tidak jauh berbeda atau hampir sama. Hal tersebut dikarenakan pemakaian algoritma yang sama, yaitu algoritma *Informed-RRT\**.

Dari pengujian yang dilakukan menunjukkan bahwa algoritma *Informed-RRT\** metode *boundary sampling* dan *Informed-RRT\** metode *goal biasing sampling* dapat berfungsi dengan baik sebagai program pencarian jalur, yang mana program tersebut jika dikembangkan dapat diaplikasikan pada robot atau kendaraan.

#### IV.KESIMPULAN

Berdasarkan pengujian pada algoritma *Informed-RRT\** metode *boundary sampling* dan

*Informed-RRT\** metode *goal biasing sampling*, maka dapat disimpulkan untuk hasil jumlah node dan waktu komputasi algoritma *Informed-RRT\** metode *boundary sampling* lebih optimal dibandingkan dengan *Informed-RRT\** metode *goal biasing sampling*, karena dari nilai rata-rata algoritma *Informed-RRT\** metode *boundary sampling* menghasilkan waktu komputasi yang lebih cepat dan juga jumlah node yang lebih sedikit. Dari hasil jarak jalur, algoritma *Informed-RRT\** metode *boundary sampling* dan *Informed-RRT\** metode *goal biasing sampling* memiliki performansi yang sama, karena nilai rata-rata jarak jalur yang tidak berbeda jauh. Penelitian ini dapat digunakan atau dikembangkan sebagai algoritma pencarian jalur untuk robot maupun kendaraan. Selain itu juga dapat digunakan sebagai ilmu untuk penelitian yang akan datang dari segi metode sampling maupun algoritmanya.

## DAFTAR PUSTAKA

- [1] A. Gasparetto, P. Boscariol, A. Lanzutti, dan R. Vidoni, "Path planning and Trajectory Planning Algorithms: A General Overview," dalam *Motion and Operation Planning of Robotic Systems*, vol. 29, G. Carbone dan F. Gomez-Bravo, Ed. Cham: Springer International Publishing, hlm. 3–27, 2015.
- [2] J. D. Gammell, S. S. Srinivasa, dan T. D. Barfoot, "Informed-RRT\*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," dalam *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, IL, hlm. 2997–3004, 2014.
- [3] M. Aria, "Algoritma Perencanaan Jalur Kendaraan Otonom berbasis Hibridisasi Algoritma BFS dan Path Smoothing," *Jurnal Ilmiah Telekomunikasi, Kendali dan Elektronika Terapan*, vol. 8, no. 1, hlm. 13–22, Jun 2020.
- [4] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, hlm. 269-271, 1959.
- [5] P. E. Hart, N. J. Nilsson, dan B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, hlm. 100-107, 1968.
- [6] A. Stentz, "Optimal and efficient path planning for unknown and dynamic environments," *Carnegie-Mellom Univ Pittsburgh PA Robotics Inst*, 1993.
- [7] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, dan S. Thrun, "Anytime search in dynamic graphs," *Artificial Intelligence*, vol. 172, no. 14, hlm. 1613-1643, 2008.
- [8] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, dan S. Teller, "Anytime motion planning using the RRT," *2011 IEEE International Conference on Robotics and Automation*, hlm. 1478-1483, 2011.
- [9] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, M. H. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, hlm. 566-580, 1996.
- [10] S. M. LaValle, "Rapidly-exploring Random Trees: A new tool for path planning," *Tech. Rep., Computer Science Dept., Iowa State University*, hlm. 1-4, 1998.
- [11] S. Karaman, dan E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no 7, hlm. 846-894, 2011.
- [12] N. M. Amato dan Y. Wu, "A randomized roadmap method for path and manipulation planning," dalam *Proceedings of IEEE International Conference on Robotics and Automation*, Minneapolis, MN, USA, vol. 1, hlm. 113–120, 1996.
- [13] G. Kang et al., "Sampling-based path planning with goal oriented sampling," *IEEE/ASME Int. Conf. Adv. Intell. Mechatronics, AIM*, hlm.1285–1290, 2016.
- [14] J. Utama, dan M.D.Saputra. Design of electric wheelchair controller based on brainwaves spectrum EEG sensor. In *IOP Conference Series: Materials Science and Engineering (Vol. 407, No. 1, p. 012080)*. IOP Publishing, 2018.
- [15] M. Elbanhawi dan M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access*, vol. 2, hlm. 56–77, 2014.