

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan pengujian pada algoritma RRT* dan *Informed-RRT** metode *boundary sampling*, maka dapat disimpulkan sebagai berikut:

1. Pada lingkungan *obstacle narrow*, algoritma *Informed-RRT** memiliki hasil pengujian untuk jarak jalur, waktu komputasi, dan jumlah node. Nilai terbaik jarak jalur yaitu 10.02, waktu komputasi 5.17, jumlah node 50. Sedangkan untuk algoritma RRT* memiliki hasil pengujian terbaik untuk jarak jalur sebesar 10.14, waktu komputasi 5.28, dan jumlah node 51. Pengujian pada lingkungan *obstacle clutter*, algoritma *Informed-RRT** memiliki hasil terbaik untuk jarak jalur sebesar 15.57, waktu komputasi 4.41, dan jumlah node 98. Algoritma RRT* pada lingkungan *obstacle clutter* memiliki hasil pengujian terbaik untuk jarak jalur sebesar 15.87, waktu komputasi 4.48, dan jumlah node 121. Pengujian terakhir dilakukan pada lingkungan *obstacle trap*, algoritma *Informed RRT** memiliki hasil pengujian terbaik untuk jarak jalur sebesar 10.82, waktu komputasi 5.02, dan jumlah node 51. Sedangkan untuk algoritma RRT* memiliki hasil pengujian terbaik untuk jarak jalur sebesar 10.83, waktu komputasi 5.12, dan jumlah node 93. Dari perbandingan keseluruhan data tersebut, dapat disimpulkan bahwa algoritma *Informed RRT** metode *boundary sampling* lebih unggul dari algoritma RRT* metode *boundary sampling*. Hal tersebut dikarenakan hasil pengujian jarak jalur, jumlah node, dan waktu komputasi algoritma *Informed RRT** memiliki nilai lebih sedikit atau lebih kecil dibandingkan algoritma RRT*. Semakin kecil nilai jarak jalur, waktu komputasi, dan jumlah node maka semakin baik performansi algoritma tersebut.
2. Hasil pengujian menunjukkan bahwa algoritma *Informed-RRT** metode *boundary sampling* menjadi algoritma yang optimal untuk digunakan

sebagai algoritma perencanaan jalur, karena memiliki performansi yang baik jika dibandingkan dengan algoritma RRT* metode *boundary sampling*

5.2 Saran

Berdasarkan pengujian dan hasil yang telah dilakukan. Pengembangan harus dilakukan sedemikian rupa agar program yang digunakan dapat lebih baik lagi. Berikut adalah saran agar program dapat lebih baik lagi kedepannya :

1. Pengujian dilakukan terhadap berbagai algoritma perencanaan jalur dengan metode yang berbeda-beda, tidak hanya berfokus di algoritma RRT* saja
2. Metode *sampling* yang digunakan bisa lebih beragam, untuk mendapatkan hasil yang lebih optimal jika dibandingkan dengan hanya menggunakan 1 metode *sampling*