

**Nama:Anda Figo Haq**

**NIM:1103213098**

### **1. Analisis Implementasi Algoritma Dijkstra**

- **Keunggulan:** Algoritma Dijkstra bekerja dengan baik dalam menemukan jalur terpendek dari titik awal ke tujuan di dalam graf yang berbobot. Algoritma ini menjamin bahwa jalur terpendek ditemukan, karena menghitung biaya secara menyeluruh pada setiap node sebelum memutuskan jalur.
- **Keterbatasan:** Karena algoritma ini mengeksplorasi seluruh node, maka untuk graf besar, Dijkstra membutuhkan waktu yang lebih lama, khususnya jika graf tidak dioptimalkan atau memiliki banyak node.
- **Kesimpulan:** Dijkstra cocok untuk aplikasi di mana pemetaan lingkungan statis dan graf sederhana atau cukup kecil, seperti di grid statis tanpa perubahan signifikan dalam rintangan.

### **2. Analisis Implementasi Algoritma A\***

- **Keunggulan:** Algoritma A\* menggunakan heuristic yang membantu mempercepat proses pencarian dengan menilai jarak antara titik saat ini dan tujuan. Heuristic ini memungkinkan A\* bekerja lebih cepat dibandingkan Dijkstra karena memprioritaskan jalur yang tampak lebih menjanjikan menuju tujuan.
- **Keterbatasan:** Hasil pencarian sangat bergantung pada fungsi heuristic. Jika heuristic kurang akurat, algoritma ini dapat mengeksplorasi terlalu banyak node yang tidak relevan, memperlambat pencarian.
- **Kesimpulan:** A\* sangat efisien untuk jalur yang membutuhkan heuristic yang lebih kompleks, misalnya pada grid yang besar dengan rintangan dinamis atau aplikasi robotik real-time yang memerlukan kecepatan dan keakuratan dalam penghindaran rintangan.

### **3. Analisis Implementasi Cell Decomposition**

- **Keunggulan:** Metode Cell Decomposition memecah ruang kerja menjadi cell atau sel yang memungkinkan perencanaan jalur di ruang yang kompleks dengan lingkungan penuh rintangan. Ini berguna terutama ketika lingkungan berisi zona terbuka yang luas dan cluster rintangan.
- **Keterbatasan:** Pendekatan ini dapat menjadi tidak efisien dalam grid yang sangat besar atau pada lingkungan dengan banyak rintangan kecil, di mana pemecahan cell berlebihan dapat memperlambat proses.
- **Kesimpulan:** Cell Decomposition efektif dalam lingkungan dengan rintangan tetap yang kompleks, tetapi kurang cocok untuk lingkungan dinamis atau lingkungan yang sangat besar yang memerlukan pembaruan cell terus-menerus.

#### 4. Analisis Rencana Gerak dan Optimasi Jalur dengan MoveIt! dan Gazebo di ROS

- **Path Searching (Pencarian Jalur):** Dengan MoveIt!, kita bisa melakukan perencanaan jalur langsung di ROS melalui visualisasi di Rviz, yang memungkinkan pemilihan titik awal dan tujuan secara intuitif. Integrasi ini memudahkan pengguna untuk merencanakan jalur dengan cepat dan melakukan simulasi untuk pengujian awal. Rviz juga menawarkan berbagai optimizer seperti CHOMP dan STOMP yang berguna dalam berbagai kasus penggunaan.
- **Trajectory Optimization (Optimasi Trajektori):** Optimasi memungkinkan jalur robot dihaluskan dan disesuaikan dengan dinamika robot, membuatnya lebih efisien untuk diimplementasikan pada robot nyata. Optimasi ini sangat penting terutama dalam mengurangi penggunaan energi robot dan waktu eksekusi, serta meningkatkan keselamatan dalam menghindari rintangan.
- **Integrasi Gazebo untuk Simulasi Fisik:** Dengan Gazebo, perencanaan jalur dapat diuji dalam simulasi fisik, memungkinkan pengujian realistis dari skenario pergerakan robot dan interaksi dengan lingkungan. Integrasi ini penting untuk memastikan bahwa jalur yang direncanakan dan dioptimalkan di MoveIt! juga sesuai di lingkungan nyata.

#### 5. Kesimpulan dan Rekomendasi

- Untuk pengaplikasian dalam lingkungan statis atau semi-statis dengan rintangan tetap, Dijkstra dan Cell Decomposition adalah pilihan yang cocok. A\* lebih cocok untuk lingkungan dinamis yang memerlukan waktu respon cepat.
- MoveIt! dan Gazebo adalah kombinasi ideal untuk perencanaan jalur dan pengujian simulasi, sehingga memungkinkan pengguna untuk mengoptimalkan jalur dan menguji pergerakan fisik sebelum implementasi pada robot fisik.
- **Rekomendasi Umum:** Untuk mencapai performa optimal, gunakan A\* atau metode hybrid (seperti Hybrid A\*) pada robot yang bergerak di lingkungan yang dinamis. Gunakan CHOMP atau STOMP untuk trajektori yang lebih halus pada robot yang membutuhkan optimasi jalur berkelanjutan di lingkungan yang padat dengan rintangan.

Analisis ini memberikan gambaran lengkap tentang implementasi, keunggulan, dan keterbatasan dari algoritma dan perencanaan jalur yang tersedia dalam ROS untuk berbagai jenis aplikasi robotik.