

## Tugas 6

Nama : Bayu Aditya R.P.

Kelas : TK 45 01

NIM : 1103210219

### 1. Implementasi dan Simulasi Probabilistic Roadmap (PRM) dengan Visualisasi Matplotlib

Tujuan dan Pendekatan:

- Algoritma PRM bertujuan untuk mencari jalur antara titik awal dan titik tujuan dengan cara membangun graf yang terdiri dari node acak di ruang bebas.
- Node-node ini dihubungkan dengan jarak tertentu dan membentuk roadmap yang memungkinkan pencarian jalur terpendek dengan metode seperti algoritma Dijkstra atau algoritma jalur terpendek berbobot lainnya.

Implementasi dan Simulasi:

- Node Sampling dan Koneksi: Node acak dibangkitkan, lalu dikoneksikan satu sama lain jika mereka berada dalam jarak tertentu, menggunakan metode struktur data seperti KDTree untuk efisiensi.
- Pencarian Jalur: Jalur dari titik awal ke tujuan diperoleh dengan algoritma pencarian jalur di graf.
- Visualisasi Matplotlib: Digunakan untuk menampilkan roadmap dan jalur yang ditemukan. Visualisasi ini bermanfaat untuk memeriksa distribusi node, konektivitas, serta validitas jalur dalam ruang bebas.

Analisis Kelebihan dan Kekurangan:

- Kelebihan: Pendekatan ini relatif mudah diimplementasikan dan cocok untuk lingkungan dengan ruang bebas yang cukup besar. Menggunakan Matplotlib memudahkan pengujian cepat tanpa memerlukan ROS.
- Kekurangan: Simulasi PRM menggunakan Matplotlib kurang fleksibel untuk integrasi dengan lingkungan dinamis dan kurang cocok untuk visualisasi interaktif yang biasanya diperlukan dalam robotika.

## 2. Implementasi dan Simulasi Probabilistic Roadmap (PRM) dengan Visualisasi Rviz di ROS

Tujuan dan Pendekatan:

- Implementasi PRM di ROS dengan visualisasi di Rviz bertujuan untuk membawa roadmap yang dihasilkan ke dalam lingkungan ROS yang memungkinkan integrasi dengan sensor dan sistem navigasi yang lebih kompleks.
- Pada implementasi ini, node dan jalur di PRM dipublikasikan sebagai Marker di ROS, dan divisualisasikan di Rviz.

Implementasi dan Simulasi:

- Integrasi ROS: Node PRM dipublikasikan dalam format ROS sehingga roadmap dan jalur dapat divisualisasikan sebagai Marker di Rviz.
- Visualisasi Rviz: Rviz menawarkan tampilan yang dinamis dan interaktif. Pengguna dapat menambahkan layer informasi lain di atas roadmap, seperti peta lingkungan dan sensor.

Analisis Kelebihan dan Kekurangan:

- Kelebihan: Implementasi ini mendukung aplikasi robotika secara langsung di lingkungan ROS, memungkinkan integrasi dengan sensor dan sistem robotika. Rviz memberikan visualisasi yang dinamis dan interaktif, membantu memantau bagaimana roadmap PRM berinteraksi dengan lingkungan nyata.
- Kekurangan: PRM cenderung tidak efisien dalam lingkungan yang berubah-ubah atau memiliki banyak rintangan bergerak, karena roadmap statis perlu diregenerasi atau diperbarui secara berkala. Juga, setup di ROS dapat memerlukan lebih banyak waktu dan kompleksitas dibandingkan dengan visualisasi sederhana.

## 3. Implementasi Algoritma Rapidly-Expanding Random Trees (RRT) di ROS dengan

Visualisasi Rviz

Tujuan dan Pendekatan:

- Algoritma RRT digunakan untuk mencari jalur dari titik awal ke tujuan dengan pendekatan eksploratif. Algoritma ini memperluas pohon dari titik awal dengan melakukan sampling acak di lingkungan, membuat node baru yang terhubung ke node terdekat dalam tree, dan

melakukan pengecekan untuk mencapai tujuan.

- Implementasi ini memublikasikan node RRT dan jalur akhir di ROS, yang kemudian divisualisasikan di Rviz.

Implementasi dan Simulasi:

- Sampling dan Ekspansi Pohon: Pada setiap iterasi, RRT melakukan sampling acak dan mengekspansi pohon ke arah node acak tersebut. Jika suatu node baru berada dekat dengan tujuan, maka pencarian berhenti, dan jalur dari titik awal ke tujuan dikembalikan.
- Integrasi ROS dan Visualisasi Rviz: Node yang dihasilkan dipublikasikan sebagai Marker, dengan jalur akhir ditampilkan sebagai LINE\_STRIP di Rviz.

Analisis Kelebihan dan Kekurangan:

- Kelebihan: RRT sangat cocok untuk lingkungan yang penuh rintangan, karena tidak memerlukan keseluruhan roadmap, dan bekerja secara langsung dengan lingkungan. Visualisasi Rviz memungkinkan pengamatan langsung dari proses eksplorasi, node yang dihasilkan, dan jalur akhir.
- Kekurangan: RRT sering kali menghasilkan jalur yang kurang optimal karena pohon yang dihasilkan tidak sepenuhnya mengeksplorasi ruang bebas. Algoritma ini juga bisa lambat jika jarak antar node kecil atau lingkungan sangat kompleks.

Perbandingan dan Rekomendasi Implementasi PRM vs RRT di ROS

- Kesesuaian Lingkungan: PRM lebih cocok di lingkungan statis dengan banyak ruang bebas dan sedikit rintangan, karena roadmap yang dihasilkan dapat digunakan kembali tanpa perlu diregenerasi. RRT, di sisi lain, lebih cocok untuk lingkungan dinamis dan lingkungan dengan rintangan padat, karena jalur dicari secara eksploratif tanpa membutuhkan roadmap.
- Visualisasi di ROS dengan Rviz: Kedua algoritma mendapatkan manfaat dari visualisasi

Rviz, yang memungkinkan pengguna untuk memantau proses pencarian jalur secara realtime. Namun, RRT cenderung lebih menarik untuk divisualisasikan, karena memperlihatkan proses eksplorasi yang nyata dari titik awal menuju tujuan.

- Fleksibilitas dan Integrasi: Kedua algoritma diimplementasikan untuk bekerja di ROS,

yang berarti mereka dapat diintegrasikan dengan data sensor robot secara langsung.

Namun, RRT lebih fleksibel dalam menyesuaikan jalur dalam lingkungan yang terus berubah, sedangkan PRM memerlukan perhitungan ulang roadmap bila ada perubahan besar pada peta atau rintangan.

#### Kesimpulan

Kedua algoritma memiliki kelebihan masing-masing tergantung pada sifat lingkungan dan kebutuhan aplikasi:

- PRM dengan Rviz merupakan pilihan yang baik untuk lingkungan statis atau semi-statis, di mana roadmap dapat diandalkan untuk waktu yang lama tanpa regenerasi.
- RRT dengan Rviz lebih disarankan untuk lingkungan dinamis atau berisi rintangan kompleks, karena metode eksplorasinya memungkinkan pencarian jalur tanpa perlu perhitungan roadmap awal.

Visualisasi dengan Rviz meningkatkan pemahaman dan pengawasan terhadap bagaimana masing-masing algoritma bekerja dan bagaimana hasil pencarian jalur dieksekusi dalam skenario nyata di

ROS.