

HASIL ANALISIS

1. PRM dengan Matplotlib (From Scratch):

PRM (Probabilistic Roadmap) menggunakan pendekatan graf probabilistik untuk menemukan jalur bebas hambatan. Dengan membuat node acak dan menghubungkannya, algoritma ini mencoba mencari jalur optimal.

Kelebihan:

- **Visualisasi yang jelas:** Pengguna dapat melihat distribusi node dan jalur dalam ruang bebas.
- **Cocok untuk ruang besar:** PRM efektif untuk masalah dengan ruang yang tidak terhalang atau terstruktur.

Kekurangan:

- **Tidak Efisien di Ruang Terbatas:** Jika ruang terlalu kecil atau padat, PRM bisa menghabiskan banyak waktu untuk membuat graf yang terhubung.
- **Penggunaan Memori:** Membangun graf yang besar memerlukan penggunaan memori yang lebih tinggi.

2. PRM dengan Visualisasi Rviz:

Menggunakan Rviz untuk visualisasi memungkinkan integrasi dengan ROS 2, memudahkan interaksi dan debugging.

Kelebihan:

- **Interaktivitas dan Real-Time:** Rviz memungkinkan visualisasi jalur secara langsung dan interaktif, mempermudah pemahaman dan analisis hasil.
- **Integrasi ROS:** Membuka peluang untuk penggunaan dalam sistem robotik nyata.

Kekurangan:

- **Kompleksitas Setup:** Memerlukan setup ROS yang lebih rumit dibandingkan dengan menggunakan Matplotlib.
- **Keterbatasan Realisme:** Kadang PRM bisa menghasilkan jalur yang tidak realistis jika distribusi node tidak optimal.

3. RRT (Rapidly-Expanding Random Trees):

RRT membangun pohon secara acak dari titik start dan berkembang ke arah tujuan. RRT lebih efisien dalam eksplorasi ruang besar dengan banyak hambatan.

Kelebihan:

- **Ekspansi Cepat:** RRT sangat baik dalam eksplorasi ruang besar dengan hambatan. Ini bisa digunakan dalam ruang yang lebih kompleks dan lebih cepat dalam mencapai tujuan.
- **Simplicity:** Sederhana dalam implementasi dan membutuhkan lebih sedikit parameter dibandingkan PRM.

Kekurangan:

- **Kualitas Jalur:** Jalur yang dihasilkan sering kali kurang halus dan tidak optimal, dan bisa berkeluk-luku.
- **Keberlanjutan Pohon:** Dapat terjebak dalam ruang yang sempit jika tidak ada ekspansi yang cukup dalam arah yang tepat.

Perbandingan PRM vs RRT:**Kecepatan dan Efisiensi:**

- **RRT** lebih cepat dalam eksplorasi ruang besar dan lebih efisien ketika ruang penuh dengan hambatan. RRT dapat mencapai titik tujuan lebih cepat daripada PRM.
- **PRM** membutuhkan banyak node dan koneksi, sehingga prosesnya lebih lambat di ruang yang besar. Tetapi PRM lebih baik untuk ruang yang luas dan terbuka, serta lebih stabil dalam pencarian jalur optimal.

Kualitas Jalur:

- **PRM** cenderung menghasilkan jalur yang lebih halus karena membangun graf dengan node yang lebih banyak dan terhubung lebih baik.
- **RRT** sering menghasilkan jalur yang lebih kasar karena pengembangan pohon secara acak, meskipun dengan eksplorasi lebih cepat.

Penggunaan dalam Lingkungan Dinamis:

- **RRT** lebih cocok digunakan dalam lingkungan yang dinamis dengan hambatan bergerak karena pendekatan ekspansinya yang cepat dan adaptif.
- **PRM** lebih cocok untuk ruang statis atau lebih terstruktur.

Analisis Visualisasi:**Matplotlib:**

- **Kelebihan:** Cocok untuk eksplorasi dan analisis hasil secara offline. Mudah digunakan untuk visualisasi algoritma sederhana.
- **Kekurangan:** Tidak interaktif dan kurang ideal untuk real-time monitoring seperti yang dibutuhkan dalam sistem robotik yang lebih kompleks.

Rviz:

- **Kelebihan:** Menyediakan visualisasi 3D dan interaktivitas yang memungkinkan pemantauan secara langsung dan lebih realistis. Cocok untuk simulasi robot di lingkungan nyata.
- **Kekurangan:** Memerlukan pengaturan lingkungan ROS yang lebih kompleks, dan kurang cocok untuk analisis offline.

Kesimpulan:

Pilihan antara PRM dan RRT bergantung pada ukuran ruang dan kompleksitas masalah.

- **PRM** lebih cocok untuk ruang besar dan terbuka di mana kualitas jalur sangat penting.
- **RRT** lebih efisien untuk ruang dengan banyak hambatan dan dinamika yang tinggi.

Visualisasi menggunakan **Rviz** menawarkan lebih banyak keuntungan dalam interaktivitas dan analisis real-time dibandingkan dengan **Matplotlib**, tetapi membutuhkan pengaturan yang lebih rumit di ROS.