

Muhammad Sidqi Nabhan

1103200179

Terdapat error ketika menginstal ros , jadi saya hanya akan menganalisis paparan tugas yang di berikan asisten

```
sidqi@sidqi-VirtualBox:~$ sudo apt install ros-neotic-desktop-full
[sudo] password for sidqi:
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package ros-neotic-desktop-full
sidqi@sidqi-VirtualBox:~$
```

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

- **Langkah Implementasi:**

- **Generate Node Acak:** Dalam file prm.py, gunakan pustaka numpy untuk menghasilkan titik-titik (node) acak dalam suatu area peta 2D.
- **Bentuk Graf:** Implementasikan fungsi untuk menghubungkan node yang dekat menggunakan jarak Euclidean, pastikan untuk memeriksa adanya hambatan agar graf hanya terbentuk di area bebas.
- **Pencarian Jalur Terpendek:** Terapkan algoritma pencarian jalur terpendek, seperti Dijkstra atau A*, pada graf yang telah dibentuk.
- **Visualisasi dengan Matplotlib:** Visualisasikan hasilnya menggunakan matplotlib dengan menampilkan node, edge, dan jalur terpendek.

- **Parameter params.yaml:** Di dalam file params.yaml, atur parameter seperti:

- Jumlah node
- Radius koneksi maksimum antar node
- Koordinat awal dan tujuan
- Ukuran peta

- **Peluncuran Kode dengan prm.launch:** Buat file prm.launch untuk memudahkan peluncuran PRM di ROS

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

- **Langkah Implementasi:** Mengikuti panduan dari repo github ini :
<https://github.com/nishadg246/Probabilistic-Roadmap>
 1. Instal ROS dan dependensi yang diperlukan.
 2. Gunakan skrip untuk membuat roadmap PRM dan publikasikan informasi ini sebagai sensor_msgs/PointCloud atau visualization_msgs/Marker agar dapat divisualisasikan di Rviz.
 3. Visualisasikan roadmap di Rviz bersama dengan jalur terpendek yang ditemukan antara titik awal dan tujuan.
- **Menjalankan PRM di Terminal:** Jalankan algoritma dengan menggunakan ROS launch file yang disediakan dalam repositori

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

- **Langkah Implementasi:** Berdasarkan panduan di repo github ini :
<https://github.com/malinta/rrt-ros>
 1. Pastikan telah mengkloning repositori dan memenuhi semua dependensi.
 2. Buat visualisasi untuk memperlihatkan node RRT, jalur yang terbentuk, serta titik awal dan tujuan di Rviz. Anda bisa menggunakan visualization_msgs/Marker untuk node dan edge RRT.
- **Menjalankan RRT di Terminal:** Gunakan perintah roslaunch untuk menjalankan RRT dan menampilkan visualisasi di Rviz.

Hasil Analisis Simulasi

Berikut adalah poin-poin untuk analisis dari implementasi dan simulasi yang telah Anda lakukan:

- **Perbandingan Efektivitas PRM dan RRT:**
 1. **PRM:** Algoritma PRM menunjukkan keefektifan di lingkungan yang tetap dengan hambatan kompleks, karena membentuk roadmap global terlebih dahulu yang berguna untuk pencarian jalur berulang kali.
 2. **RRT:** RRT lebih efisien untuk pencarian jalur sekali pakai di lingkungan yang besar atau dinamis, karena secara langsung mencari jalur dari awal ke tujuan tanpa membangun graf global.
- **Performa Visualisasi:**

1. **Matplotlib (PRM):** Cocok untuk visualisasi yang sederhana dan cepat di luar ROS, namun memiliki keterbatasan jika dibandingkan dengan Rviz dalam visualisasi 3D atau lingkungan kompleks.
 2. **Rviz (PRM dan RRT):** Menyediakan visualisasi yang lebih kaya dan interaktif dalam ROS, memungkinkan pengguna untuk melihat roadmap atau pohon RRT yang diperluas secara real-time, memberikan lebih banyak kontrol dalam simulasi.
- **Evaluasi Parameter:**
 1. Jumlah node dan radius koneksi di PRM mempengaruhi kepadatan graf dan kompleksitas komputasi.
 2. Pada RRT, jarak langkah berperan penting dalam menentukan efisiensi pohon dalam menjelajahi ruang tanpa menghabiskan sumber daya terlalu banyak.