

Nama : Fadhil Dzikri Aqila

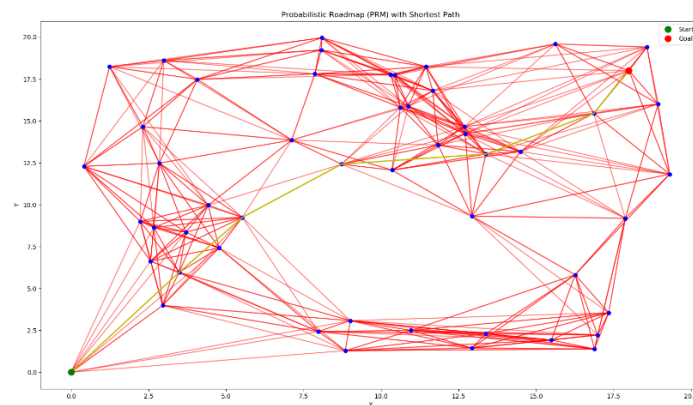
NIM : 1103213136

Kelas : TK-45-G09

## **Analisis Implementasi dan Simulasi Probabilistic Roadmap (PRM) dan Rapidly-Expanding Random Trees (RRT) di ROS.**

### **1. Analisis Implementasi dan Simulasi PRM dengan Python**

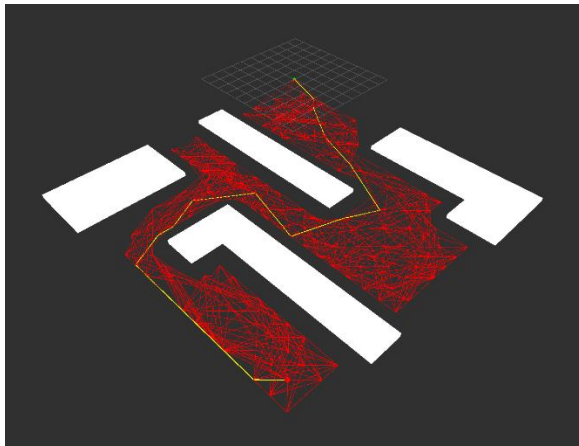
Implementasi PRM dengan Python berfokus pada pembangunan roadmap yang terdiri dari node acak yang terhubung dalam lingkungan bebas hambatan. Dalam pendekatan ini, algoritma PRM membangun graf dari node yang terhubung oleh edge jika jaraknya berada dalam radius tertentu. PRM biasanya digunakan dalam lingkungan statis. Pada simulasi PRM di Python, matplotlib digunakan untuk menampilkan roadmap dan jalur yang ditemukan, sehingga pengguna dapat melihat jalur dari titik awal hingga tujuan dalam tampilan dua dimensi sederhana. Namun, simulasi ini terbatas pada tampilan statis dan kurang fleksibel dibandingkan simulasi berbasis ROS, yang mendukung integrasi yang lebih mendalam.



Pada gambar di atas, node dan edge dari roadmap ditampilkan di lingkungan dua dimensi tanpa hambatan fisik yang signifikan. Titik awal dan tujuan diberi warna hijau dan merah, sedangkan garis kuning menunjukkan jalur terpendek yang ditemukan di antara keduanya. Garis merah mewakili koneksi antar-node yang membentuk roadmap PRM, dan node-node tersebut ditampilkan sebagai titik biru. Visualisasi ini membantu menganalisis efektivitas PRM dalam menemukan jalur optimal di lingkungan terbuka dengan pengaturan dua dimensi.

### **2. Analisis Implementasi dan Simulasi PRM dengan RViz**

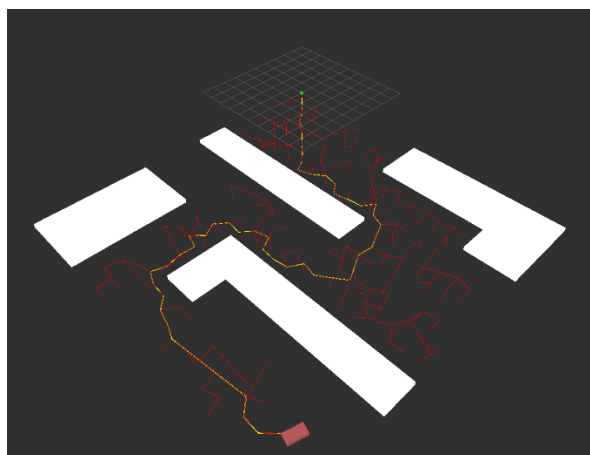
Implementasi PRM di ROS dengan visualisasi di RViz memberikan pendekatan yang lebih komprehensif dan realistis dibandingkan dengan Python standar. Di RViz, node dan edge yang membentuk roadmap divisualisasikan sebagai marker dalam lingkungan tiga dimensi, memberikan gambaran yang lebih mendekati aplikasi di dunia nyata. Penggunaan RViz memudahkan integrasi PRM dengan sistem lain di ROS dan memungkinkan pengguna untuk memvisualisasikan serta menyesuaikan parameter PRM secara langsung. Implementasi ini juga menawarkan kontrol yang lebih baik dalam pengaturan lingkungan yang lebih dinamis, dibandingkan dengan PRM berbasis Python yang hanya mengandalkan matplotlib.



Pada gambar di atas, terlihat jalur roadmap yang dibentuk oleh node dan edge di lingkungan dengan beberapa hambatan (diwakili oleh bentuk putih). Garis merah menunjukkan koneksi antar-node dalam roadmap, sementara garis kuning menunjukkan jalur terpendek dari titik awal hingga tujuan. Visualisasi ini memungkinkan analisis efektivitas PRM dalam menemukan jalur optimal di lingkungan yang kompleks dengan adanya hambatan.

### 3. Analisis Implementasi RRT di ROS dengan RViz

Implementasi RRT di ROS dengan visualisasi di RViz memiliki pendekatan berbeda dari PRM, karena RRT dirancang untuk eksplorasi ruang secara progresif. Pohon jalur dibentuk dari titik awal menuju target dengan memperluas node secara acak ke arah ruang bebas hambatan. Visualisasi RRT di RViz memungkinkan pengguna melihat perkembangan jalur secara dinamis, di mana pohon terbentuk melalui proses ekspansi bertahap. Algoritma RRT unggul dalam lingkungan dinamis atau tidak pasti karena tidak memerlukan roadmap awal seperti PRM, menjadikannya lebih fleksibel dan adaptif dalam lingkungan dengan hambatan yang berubah atau bergerak.



Pada gambar di atas, terlihat peta dengan beberapa hambatan (diwakili oleh blok putih), sedangkan jalur RRT ditampilkan dengan garis berwarna merah yang bercabang secara acak untuk mengeksplorasi ruang di sekitar hambatan. Garis kuning menunjukkan jalur yang dihasilkan dari titik awal menuju tujuan, menghindari hambatan. RRT cocok untuk lingkungan dengan hambatan karena dapat mengeksplorasi ruang secara efisien dengan menciptakan cabang-cabang dari titik awal menuju area tujuan. Visualisasi di RViz ini memungkinkan pengguna memantau bagaimana algoritma RRT mencari jalur bebas hambatan di antara objek-objek yang ada.