

Pada simulasi perencanaan jalur robot menggunakan ROS Motion Planning yang mencakup Path Searching dan Trajectory Optimization dengan Rviz dan Gazebo, beberapa algoritma populer telah diuji dan dianalisis, yakni Dijkstra, A*, dan GBFS. Setiap algoritma memiliki karakteristik dan keunggulan tersendiri dalam menemukan jalur yang efisien dari titik awal ke titik tujuan, sambil menghindari rintangan yang ada dalam simulasi. Implementasi Dijkstra menunjukkan performa yang baik dalam menghasilkan jalur terpendek secara konsisten. Algoritma ini bekerja dengan mengevaluasi seluruh simpul di sepanjang graf hingga menemukan jalur paling optimal, memastikan bahwa setiap langkah menuju tujuan memiliki biaya yang minimum. Meskipun efektif dalam memberikan jalur paling ekonomis, Dijkstra dikenal membutuhkan waktu pencarian yang relatif lebih lama dibandingkan algoritma lain, terutama dalam graf atau grid besar. Hal ini menjadikan Dijkstra kurang ideal untuk aplikasi yang membutuhkan respons cepat, namun tetap unggul dalam situasi yang mengutamakan jalur termurah.

Sementara itu, algoritma A menawarkan solusi yang lebih efisien dengan menggabungkan pendekatan heuristik. Algoritma ini tidak hanya mempertimbangkan jarak yang telah ditempuh, tetapi juga memperkirakan jarak yang tersisa ke tujuan, yang memungkinkannya untuk lebih fokus pada jalur yang menjanjikan. Penggunaan heuristik membantu mempercepat proses pencarian dan menghasilkan jalur yang mendekati optimal. Pada simulasi ini, A* berhasil mengurangi waktu pencarian jalur dibandingkan Dijkstra, meskipun terkadang jalur yang dihasilkan memiliki sedikit perbedaan dalam optimalitas biaya. Penggunaan A* sangat tepat dalam aplikasi yang menginginkan keseimbangan antara waktu respons cepat dan jalur yang mendekati optimal.

Greedy Best-First Search (GBFS) adalah algoritma lain yang berfokus pada kecepatan dalam menemukan jalur menuju tujuan. Berbeda dengan Dijkstra dan A*, GBFS memprioritaskan simpul yang secara langsung mendekati titik tujuan berdasarkan perkiraan heuristik tanpa mempertimbangkan total biaya dari titik awal. Hal ini membuat GBFS sangat cepat, namun kurang akurat dalam menghasilkan jalur paling efisien, terutama dalam lingkungan dengan banyak rintangan atau jalur sempit. Pada simulasi, GBFS menunjukkan kemampuan respons yang cepat, tetapi jalur yang dihasilkan sering kali lebih panjang atau tidak seoptimal algoritma lainnya. Algoritma ini cocok untuk aplikasi yang lebih mementingkan waktu respons cepat daripada akurasi jalur, seperti dalam situasi darurat atau aplikasi real-time yang kurang sensitif terhadap biaya perjalanan.

Selain perencanaan jalur, simulasi juga mencakup Trajectory Optimization menggunakan algoritma CHOMP (Covariant Hamiltonian Optimization for Motion Planning) dan STOMP (Stochastic Trajectory Optimization for Motion Planning). Kedua algoritma ini bertujuan untuk menyempurnakan jalur yang dihasilkan agar lebih aman dan efisien. CHOMP berfungsi dengan mengurangi belokan tajam dalam jalur dan menciptakan pergerakan yang lebih halus, sedangkan STOMP menggunakan pendekatan stokastik untuk menghindari rintangan dengan cara yang lebih fleksibel. Pada simulasi ini, CHOMP dan STOMP terbukti efektif dalam menghilangkan ketidakaturan pada jalur yang ditemukan oleh algoritma pencarian jalur dasar. Hasil optimisasi trajektori dari kedua algoritma ini menunjukkan perbaikan signifikan dalam kualitas jalur, yang mengurangi risiko tabrakan dan memungkinkan pergerakan robot yang lebih stabil di Gazebo.

Melalui analisis ini, terlihat bahwa setiap algoritma memiliki kelebihan dan kekurangan yang dapat dimanfaatkan sesuai kebutuhan spesifik aplikasi. Algoritma Dijkstra sangat berguna dalam situasi yang mengutamakan jalur dengan biaya minimal, walaupun lebih lambat. Algoritma A menawarkan solusi perencanaan jalur yang seimbang antara optimalitas jalur dan efisiensi waktu, sedangkan GBFS cocok untuk situasi yang memerlukan respons cepat dengan jalur yang mungkin tidak optimal. Di sisi lain, penggunaan CHOMP dan STOMP sebagai bagian dari Trajectory Optimization menunjukkan pengaruh yang signifikan dalam menciptakan pergerakan robot yang aman dan efisien. Dengan hasil ini, aplikasi perencanaan jalur dapat disesuaikan tergantung pada kebutuhan praktis di lapangan, baik untuk keperluan navigasi robot dalam lingkungan tertutup yang penuh rintangan maupun aplikasi lainnya yang memerlukan pemrosesan cepat dan pergerakan aman.