

Laporan Analisis Week 14 Robotika

Kavilla Zota Qurzian

1103210190

1. Perencanaan Jalur Sederhana

Pada perintah pertama, robot diminta untuk merencanakan jalur dari titik awal (0, 0) menuju titik tujuan (4, 4) dalam sebuah matriks 2D, dengan rintangan yang ditandai oleh angka 1 dan jalan bebas yang ditandai dengan angka 0. Pendekatan yang digunakan adalah algoritma pencarian jalur, seperti algoritma A* atau BFS, yang mempertimbangkan rintangan untuk menentukan jalur terpendek yang dapat dilalui oleh robot.

2. Gerakan Robot dengan Input Pengguna

Perintah kedua melibatkan interaksi pengguna, dimana robot akan bergerak ke posisi baru berdasarkan input yang diberikan. Dengan menggunakan input pengguna (misalnya arah gerak seperti atas, bawah, kiri, kanan), robot akan mencetak posisi terbarunya di setiap langkahnya. Kode ini membutuhkan pengolahan input secara real-time dan pembaruan posisi robot setelah setiap pergerakan.

3. Simulasi Robot Menghindari Rintangan

Pada perintah ketiga, robot tidak hanya bergerak menuju tujuan, tetapi juga harus menghindari rintangan yang ada di peta 2D. Pendekatan yang digunakan untuk mensimulasikan hal ini adalah dengan memanfaatkan algoritma pencarian jalur dinamis yang dapat menghindari rintangan selama perjalanan, seperti A* dengan penanganan rintangan atau model pencarian jalur berbasis simulasi. Setiap langkah yang diambil oleh robot harus dicatat untuk memberi gambaran jelas mengenai jalur yang diambil.

4. Penjadwalan Robot dengan Prioritas

Perintah keempat melibatkan penjadwalan tugas-tugas yang dimiliki robot berdasarkan prioritas. Setiap tugas memiliki tingkat prioritas yang berbeda, dan robot harus menyelesaikan tugas dengan prioritas tertinggi terlebih dahulu. Sistem antrean prioritas seperti heap atau queue prioritas digunakan untuk memastikan bahwa tugas dengan prioritas tertinggi diproses terlebih dahulu. Kode ini memerlukan pengelolaan sumber daya yang efisien agar robot dapat bekerja secara optimal.

5. Robotik dengan Sistem Event-Driven

Pada perintah kelima, robot harus beroperasi dalam sistem event-driven, yang berarti robot hanya bergerak jika ada perubahan di lingkungannya, seperti munculnya rintangan baru atau perubahan tujuan. Dalam sistem ini, robot mendengarkan dan merespon peristiwa atau perubahan dalam lingkungan secara otomatis tanpa perlu menjalankan operasi berulang, yang membuatnya lebih efisien dalam merespon perubahan real-time.

6. **Robot dengan Model Probabilistik**

Perintah terakhir melibatkan penggunaan model probabilistik untuk menentukan jalur terbaik menuju tujuan. Mengingat ketidakpastian dalam data sensor, robot menggunakan metode probabilistik, seperti Bayesian atau algoritma filter Kalman, untuk mengurangi kesalahan yang mungkin terjadi akibat data sensor yang tidak akurat. Pendekatan ini memungkinkan robot untuk memilih jalur yang lebih aman atau optimal meskipun terdapat ketidakpastian dalam pengukuran.

Secara keseluruhan, setiap perintah mengarah pada penggunaan metode yang berbeda dalam robotika yang bergantung pada pemrograman berbasis Rust untuk memecahkan masalah navigasi, interaksi pengguna, penjadwalan tugas, serta respons terhadap lingkungan yang dinamis. Pemilihan teknik yang tepat sangat bergantung pada kondisi dan kebutuhan spesifik dari sistem robot yang sedang dikembangkan.