

A

ROS adalah alat dengan serangkaian pustaka dan fitur untuk membantu pembuatan software bagi robot (dari tahap pengembangan, perancangan, pengujian, penerapan, hingga pemeliharaan). ROS menolong supaya framework pembuatan robot dapat tertata dalam satu software. Ini juga menjamin kemudahan komunikasi/interaksi antar komponen. Kemudahan-kemudahan ini akan meningkatkan efisiensi developer robot.

B

ROS2 adalah versi terbaru ROS yang dirancang untuk mengatasi beberapa keterbatasan. Beberapa perbedaan utamanya terletak di arsitektur komunikasi yang lebih efisien di berbagai jaringan, keamanan yang lebih baik lewat otentikasi dan enkripsi, dukungan multiplatform termasuk Windows yang awalnya tidak didukung penuh, performa yang lebih tinggi, serta pemeliharaan dan perluasan (jika proyek ingin dikembangkan) secara lebih mudah. Tentu saja developer lebih senang memakai ROS2 dengan segala kelebihanannya tersebut.

C

Tujuan utamanya adalah supaya dapat mengidentifikasi dan memetakan kesalahan lebih awal sebelum terlanjur sudah membeli alat dan bahan atau bahkan merakitnya. Mungkin saja ada bug, salah perhitungan, atau lainnya yang dapat menggagalkan suatu proyek robot jika tidak dicek terlebih dahulu.

Contoh kasus pemanfaatan simulasi: tim URO ingin mengembangkan drone pengantar barang, maka dapat lebih dahulu menggunakan simulasi untuk menguji kinerja PID, ketahanan material pilihan, dan efisiensi rute pengantaran guna mengoptimalkan software tanpa harus menghabiskan uang untuk prototipe fisik.

D

Gazebo adalah simulator untuk membuat lingkungan fisik robot. Ini mendukung simulasi fisika yang realistis dari segi sensor dan visualisasi. Langkah menggunakan Gazebo:

1. Instalasi: download ROS dan Gazebo.
2. Konfigurasi workspace: buat workspace ROS dan tambahkan paket Gazebo.
3. Model robot: buat atau unduh model robot dalam format SDF.
4. Peluncuran simulasi: Gunakan file launch ROS untuk memulai Gazebo dengan model robot, serta node ROS yang diperlukan untuk mengontrol robot dalam simulasi.
5. Interaksi: memakai terminal ROS untuk mengirim perintah ke robot dan memantau data dari sensor.

E

Navigasi robot di dunia simulasi memerlukan pemetaan 2d (input) ke ruang 3d (output) menggunakan berbagai proyeksi dan kerangka analisis matematis. Manfaatnya banyak, seperti supaya robot tidak nabrak-nabrak ketika sedang berjalan. Untuk bisa melakukannya, diperlukan dua hal berikut:

1. Mapping: Proses menciptakan peta dari lingkungan. Robot dapat menggunakan sensor (seperti LIDAR) untuk mengumpulkan data tentang lingkungan sekitarnya dan membangun peta yang akurat.
2. Lokalisasi: Proses menentukan posisi robot dalam peta yang sudah dibuat. Algoritma seperti SLAM (Simultaneous Localization and Mapping) dapat digunakan untuk membantu robot memahami lokasinya sambil juga membangun peta.

F

TF memungkinkan robot untuk melacak posisi dan orientasi dari berbagai frame referensi dalam ruang 3d. TF membantu robot memahami hubungan antara berbagai bagian tubuhnya dan posisi di ruang, sehingga pergerakan dapat dilakukan dengan akurat (tidak nabrak dan, lebih dari itu, presisi). Contoh penggunaan: robot memiliki sensor kamera di bagian depan, maka TF digunakan untuk mengkonversi koordinat objek yang terdeteksi oleh kamera ke sistem koordinat robot, sehingga jarak relatifnya dapat dihitung dan robot pun dapat bergerak menuju objek tersebut dengan tepat.