

SOAL 1 URO TEST PROGRAMMING

- a. Robot Operating System (ROS) adalah sebuah framework atau meta operating system yang bersifat open source yang tentunya dapat digunakan untuk robot. Karena bersifat open source, terdapat banyak sekali source code yang tersedia dan tentunya akan memudahkan kita dalam pengembangan robot, selain itu juga tersedia banyak sensor dan perangkat yang akan menambah fungsionalitas pada robot. ROS berperan sangat penting untuk pengembangan robot modern karena ROS sendiri merupakan kerangka kerja utama dalam ekosistem pengembangan robot modern.

Mengapa ROS penting untuk integrasi berbagai komponen robot seperti sensor, aktuator, dan kamera dalam satu sistem yang bekerja harmonis?? ROS sendiri menyediakan kerangka kerja modular yang memungkinkan komunikasi antar komponen yang efisien, abstraksi perangkat keras, sinkronisasi data, serta dukungan simulasi dan pengujian. Dengan ROS, robot modern dapat menggabungkan data dari sensor, mengendalikan aktuator, dan memproses informasi dari kamera dalam satu sistem yang terintegrasi dan bekerja harmonis

- b. Perbedaan utama ROS dan ROS2 terletak pada peningkatan fitur. Beberapa perbedaan utamanya yaitu terletak pada arsitektur komunikasi, real-time capability, dukungan multi-platform, keamanan, manajemen dependensi, dukungan untuk sistem embedded, skalabilitas dan performa di sistem distribusi, lifecycle management, serta dukungan pengembangan berkelanjutan. Hal tersebut juga yang membuat perusahaan lebih memilih menggunakan ROS2 untuk proyek baru.

Untuk performa ROS2 sendiri, ROS2 mendukung aplikasi real-time, yang penting untuk kontrol robotik yang presisi, terutama dalam industri atau robot otonom. Dalam segi keamanan, dengan adanya enkripsi dan autentikasi, ROS2 menawarkan keamanan yang lebih baik untuk aplikasi di lingkungan yang rentan terjadi peretasan dan sebagainya. ROS2 juga Menyediakan fitur lifecycle management yang memungkinkan pengembang mengelola siklus hidup node dengan lebih baik. Ini berguna untuk robot yang harus beroperasi dalam jangka waktu lama atau dalam lingkungan keritis.

- c. Simulasi memberikan lingkungan yang aman dan terkendali untuk menguji berbagai skenario, algoritma, dan interaksi, sehingga

pengembang bisa mengidentifikasi masalah atau meningkatkan performa sistem sebelum investasi besar dilakukan dalam perangkat keras. Keuntungannya, dapat menghemat biaya prototype, mempercepat pengujian, pengujian di lingkungan yang terkendali, mengidentifikasi masalah lebih awal, serta mengoptimasi desain.

Contoh kasus dalam pengembangan robot penjelajah ruang angkasa. Dalam kasus ini, pengembang ingin membuat robot untuk mengeksplorasi permukaan mars. Pengujian robot secara langsung ke planet mars tentu akan memakan biaya yang sangat mahal, dan pengujian di bumi juga terbatas pada lingkungan simulasi mars yang hanya sebagian mirip. Dengan simulasi, pengembang dapat mereplika permukaan mars yang asli, gravitasi rendah, serta kondisi atmosfer. Hal tersebut memungkinkan pengujian robot yang sangat mendekati situasi di dunia nyata. Pengembang dapat menguji algoritma navigasi, kendali daya, dan interaksi dengan medan yang sulit, tanpa memerlukan uji coba mahal di lapangan.

- d. Gazebo adalah perangkat lunak simulasi robot yang biasanya digunakan untuk membuat serta menguji robot dalam lingkungan fisik virtual. Gazebo memungkinkan simulasi yang realistis dari berbagai sensor, aktuator, serta lingkungan yang kompleks dengan dinamika fisika yang akurat. Karena kemampuan simulasi ini, Gazebo sering digunakan bersama dengan Robot Operating System (ROS) untuk memungkinkan pengujian sistem robot sebelum perangkat keras robotik sebenarnya dibuat.

bagaimana Gazebo digunakan untuk mensimulasikan lingkungan fisik bagi robot? Gazebo sendiri memiliki fitur simulasi fisika, simulasi sensor, kendali aktuator, lingkungan 3d, serta integrasi dengan ROS yang tentunya akan sangat berguna untuk mensimulasikan lingkungan fisik bagi robot.

Langkah-langkah mengintegrasikan ROS dengan gazebo:

- a. Instalasi Gazebo dan ROS – sebelum integrasi, tentunya kita harus menginstall terlebih dahulu ROS dan Gazebo. Pada ROS versi Melodic atau Noetic, Gazebo biasanya sudah terinstal bersama paket ROS.
- b. Buat atau Gunakan Model Robot di Gazebo – Dapat menggunakan model robot yang ada atau membuat model robot Anda sendiri dengan format URDF (Unified Robot Description Format) atau SDF (Simulation Description Format).

- c. Integrasi Gazebo dengan ROS (ROS Gazebo plugin) - Untuk menghubungkan Gazebo dengan ROS, perlu menggunakan plugin khusus seperti gazebo_ros yang memungkinkan robot dikontrol melalui ROS. Tambahkan plugin gazebo_ros ke file URDF atau SDF.
 - d. Siapkan ROS Node untuk Mengontrol Robot - Setelah plugin terhubung, kita bisa menulis ROS node untuk mengontrol robot.
 - e. Launch Gazebo Bersama ROS – Buat file launch ROS yang memuat robot ke dalam simulasi Gazebo dan mulai node kontrol.
- e. Navigasi robot di dunia simulasi melibatkan berbagai konsep dan teknik penting yang memungkinkan robot untuk bergerak secara otonom di lingkungan yang belum dikenal atau diketahui. Beberapa konsep dasar yang harus dipahami meliputi mapping, lokalisasi, path planning, dan motion control.

Bagaimana fitur ini dapat diimplementasikan pada robot? Konsep ini bisa diimplementasikan menggunakan Robot Operating System (ROS) dengan paket ROS Navigation Stack yang menyediakan fungsionalitas lengkap untuk navigasi, termasuk pemetaan, lokalisasi, dan perencanaan jalur.

- f. TF (Transform) adalah sebuah library di dalam ROS yang memungkinkan pelacakan posisi dan orientasi dari berbagai frame koordinat yang terkait dengan robot atau objek lain dalam sebuah sistem. TF memungkinkan robot memahami bagaimana berbagai bagian dari tubuhnya atau objek eksternal saling berhubungan dalam ruang tiga dimensi (3D), serta bagaimana objek-objek ini berubah seiring waktu. TF bekerja dengan publikasi transformasi dari satu frame ke frame lain dalam bentuk matriks transformasi homogen (matriks 4x4), yang menyimpan informasi posisi (translasi) dan orientasi (rotasi) antara frame. TF mengelola tree (pohon) dari transformasi ini yang disebut TF tree, yang memungkinkan robot atau komponen lain untuk mengetahui hubungan spasial dari semua frame di sistem.

Misalnya, Saya sedang mengembangkan robot bergerak dengan kamera dan LiDAR di atasnya. Saya ingin memastikan bahwa saat robot bergerak di lingkungan simulasi, kamera dan LiDAR juga memberikan data yang sesuai dengan posisi dan orientasi robot. Untuk itu, TF akan membantu mengelola transformasi antara berbagai komponen robot sehingga data sensor dapat digunakan secara akurat.