

## **01 Introduction to ROS**

Di bab ini, kita akan belajar tentang ide dasar ROS dan cara menggunakan sistem manajemen paket ROS untuk memulai pemrograman ROS. Mengapa ROS penting? ROS adalah kerangka kerja yang fleksibel yang memberikan alat dan perpustakaan untuk menulis perangkat lunak robot. ROS dimulai pada tahun 2007 dan dikembangkan di Willow Garage, sebuah lab penelitian robotika. Tujuannya adalah membuat cara standar untuk memprogram robot dengan menyediakan komponen perangkat lunak siap pakai yang mudah diintegrasikan ke dalam aplikasi robot kustom.

ROS mirip dengan sistem operasi karena menyediakan fungsi seperti abstraksi perangkat keras, manajemen paket, dan toolchain pengembang. ROS berfungsi sebagai "meta-OS," membantu pengembangan perangkat lunak robot dan menyediakan layanan mirip sistem operasi dalam mengelola sumber daya dan alur kerja pengembangan.

## **02 Getting Started With ROS Programming**

Pada tahap awal belajar pemrograman ROS, kita akan memahami beberapa konsep dasar dan keterampilan praktis yang diperlukan untuk mengembangkan perangkat lunak robot menggunakan ROS. Ini melibatkan pemahaman tentang ROS Master, ROS Nodes, Manajemen Paket ROS, dan Sistem Komunikasi ROS. Kita juga akan mempelajari konfigurasi dan pemrograman node, serta praktik pengembangan seperti pembuatan node ROS dan penggunaan berkas pesan kustom.

Selain itu, kita akan menerapkan konsep-konsep ini pada kasus nyata untuk menyelesaikan tugas-tugas robotika yang lebih kompleks. Dengan memahami ini, pemrogram dapat membangun dasar yang kuat untuk mengembangkan perangkat lunak robotika dengan ROS, termasuk keterampilan mengorganisir proyek, berkomunikasi antar komponen, dan menggunakan alat dan konsep ROS.

## **03 Working with ROS for 3D Modelling**

Dalam fase awal pembuatan robot, langkah pertama melibatkan desain dan pemodelan. Gunakan alat CAD seperti Autodesk Fusion 360 atau SolidWorks untuk membuat model robot. Pemodelan robot memungkinkan simulasi untuk memeriksa cacat dalam desain sebelum memasuki fase manufaktur. Bab ini membahas proses desain dua robot dengan tujuh Derajat Kebebasan dan robot penggerak diferensial.

## **04 Simulating Robots Using ROS and Gazebo**

Setelah merancang model 3D robot, langkah berikutnya adalah mensimulasikannya. Dalam bab ini, kita akan menggunakan simulator Gazebo untuk mensimulasikan robot dengan tujuh Derajat Kebebasan dan robot penggerak diferensial. Gazebo adalah simulator multi-robot untuk simulasi robotik kompleks di dalam dan di luar ruangan. Bab ini menjelaskan integrasi Gazebo dengan ROS dan cara menggunakan model simulasi yang sudah ada

## **05 Simulating Robots Using ROS, CoppeliSim, and Webots**

Bab ini membahas cara menyiapkan simulator CoppeliSim dan Webots dan menghubungkannya dengan ROS. Kita akan mempelajari pengaturan simulator ini, mensimulasikan manipulator robot, menyiapkan Webots dengan ROS, menulis pengontrol pertama, dan membuat node teleoperasi menggunakan webots\_ros. Semua ini akan membantu kita memahami cara menggunakan simulator ini baik sebagai perangkat lunak mandiri maupun terhubung dengan layanan dan topik ROS.