

Nama : Irman Prayista

NIM : 1103210094

Kelas : TK-45-02

Chapter 1

1. Introduction to ROS

Apa itu ROS?

Robot Operating System (ROS) adalah sebuah sistem perangkat lunak yang digunakan untuk membuat program-program robot. Sistem ini dirancang agar mudah digunakan dan memungkinkan bagian-bagian robot bekerja bersama.

Apa yang Ditawarkan ROS?

ROS menyediakan berbagai alat dan pustaka yang membantu pengembang membuat, menguji, dan menjalankan robot. Dengan menggunakan ROS, robot bisa melakukan berbagai tugas, seperti bergerak, mengenali benda, atau berinteraksi dengan lingkungan.

Cara Kerja ROS

ROS menggunakan sistem komunikasi yang mempermudah berbagai bagian robot untuk saling berbicara. Komunikasi ini dilakukan lewat mekanisme sederhana seperti pengirim-penerima pesan (*publisher-subscriber*) atau permintaan layanan (*service-client*).

Keunggulan Sumber Terbuka

Karena ROS adalah proyek sumber terbuka, banyak orang di seluruh dunia ikut berkontribusi. Akibatnya, ROS memiliki banyak fitur yang bisa digunakan, seperti alat untuk membuat peta, navigasi, atau mengontrol lengan robot.

Kenapa ROS Penting?

ROS sering digunakan oleh peneliti, mahasiswa, dan pengembang karena membantu mereka membuat robot lebih cepat dan lebih mudah, tanpa harus memulai semuanya dari awal.

Chapter 2

1. ROS Programming

- Buka Sumber Github

<https://github.com/PacktPublishing/Mastering-ROS-for-Robotics-Programming-Third-edition/tree/main>

- Buka folder chapter 2 lalu copy folder `mastering_ros_demo_pkg` dan paste pada linux -> home -> `irmanprayista` -> `catkin_ws` -> `src`
- Buka Ubuntu lalu masuk folder `catkin_ws` dan `catkin_make`
- Buka `roscore`
- Ketik `roslaunch mastering_ros_demo_pkg demo_topic_publisher`.
Pada perintah ini, `mastering_ros_demo_pkg` adalah nama package, sedangkan `demo_topic_publisher` adalah nama node yang akan dijalankan. Publisher bertugas untuk menerbitkan (publish) pesan ke suatu topik (topic) di ROS. Node publisher biasanya digunakan untuk mengirimkan data atau informasi secara periodik atau berdasarkan peristiwa tertentu ke topik yang terdaftar di ROS
- Ketik `roslaunch mastering_ros_demo_pkg demo_topic_subscriber`.
Subscriber untuk subscribe (mendengarkan) topik tertentu yang sudah dipublikasikan oleh node lain (seperti `demo_topic_publisher`). Node ini akan menerima pesan yang dipublikasikan pada topik tersebut dan memprosesnya sesuai dengan implementasinya.

Chapter 3

1. ROS With RVIZ

- Menggunakan RVIZ
`roslaunch mastering_ros_robot_description_pkg view_demo.launch`
Dalam perintah ini, `mastering_ros_robot_description_pkg` adalah nama package yang berisi file `view_demo.launch`, yang biasanya digunakan untuk memuat atau mengonfigurasi robot yang ingin dilihat atau di-interaksikan dengan RViz atau simulator lainnya.
- Menjelaskan model xacro dari lengan tujuh DOF
`roslaunch mastering_ros_robot_description_pkg view_arm.launch`
`view_arm.launch`
file ini kemungkinan besar digunakan untuk memvisualisasikan model lengan robot (robot arm) dalam lingkungan seperti RViz.

- Membuat model robot untuk robot bergerak menggunakan penggerak diferensial

roslaunch mastering_ros_robot_description_pkg

view_mobile_robot.launch

File launch ini digunakan untuk memuat model 3D robot bergerak yang didefinisikan dalam file URDF (Unified Robot Description Format) atau Xacro. Robot ini mungkin memiliki roda atau komponen lain yang memungkinkan pergerakan di lingkungan simulasi.

Chapter 4

1. ROS and Gazebo

- **roslaunch seven_dof_arm_gazebo seven_dof_arm_gazebo_control.launch** untuk menjalankan file launch yang bernama `seven_dof_arm_gazebo_control.launch` dari package ROS bernama `seven_dof_arm_gazebo`.

Fungsinya:

- Meluncurkan Simulasi Gazebo**
- Mengaktifkan Kontroler Robot**
- Menghubungkan dengan ROS**
- Testing dan Debugging**

- **rostopic pub -s /seven_dof_arm/joint4_position_controller/command std_msgs/Float64 "data: 1.0"**

Kode ini digunakan untuk mengirimkan pesan ke sebuah topic di ROS.

Fungsinya:

- rostopic pub**

Perintah untuk menerbitkan (publish) pesan ke suatu topic ROS.

- s**

Menjalankan penerbitan dalam mode synchronous. Artinya, perintah akan menunggu konfirmasi bahwa pesan diterima sebelum melanjutkan.

- /seven_dof_arm/joint4_position_controller/command**

Kode di atas digunakan untuk topik tujuan di mana pesan akan dikirim.

- std_msgs/Float64**

std_msgs/Float64 adalah tipe pesan sederhana untuk angka dengan presisi ganda (double).

e. **"data: 1.0"**

Isi pesan yang dikirimkan. Dalam hal ini, data memiliki nilai **1.0**. Nilai ini biasanya mengacu pada **posisi sudut** (radians) yang harus dicapai oleh **joint 4**.