

Nama : Ketut Satria Wibisana

NIM : 1103213148

Kelas : TK-45-G09

Script / Naskah ROS Chapter 1 – 4

1. Chapter 1 – Introduction to ROS

Apa itu ROS?

ROS, atau Robot Operating System, adalah sebuah meta-operating system yang dikhususkan untuk robot. ROS tidak hanya sekadar perangkat lunak biasa, melainkan sebuah framework yang menyediakan berbagai layanan penting, seperti:

- a. Abstraksi perangkat keras,
- b. Pengendalian perangkat,
- c. Manajemen paket,
- d. Komunikasi antar proses, dan
- e. Alat pengembangan perangkat lunak untuk robotika.

Secara sederhana, ROS mempermudah pengembangan dan pengoperasian kode untuk robot. Sebagai contoh, jika Anda ingin mengendalikan lengan robot, mengenali objek dengan kamera, atau bahkan menciptakan robot yang dapat beroperasi secara mandiri, ROS menawarkan berbagai pustaka dan alat yang dapat langsung digunakan.

Bagaimana ROS Bekerja?

ROS memiliki sesuatu yang disebut ROS Runtime Graph, yaitu jaringan komunikasi antar komponen dalam sistem robot yang dikenal dengan istilah nodes.

Setiap node merupakan program kecil yang bertugas untuk menjalankan fungsi tertentu.

Contohnya:

- Ada node untuk mengendalikan roda robot,
- Ada node untuk membaca data sensor,
- Ada node untuk memproses gambar dari kamera, dan sebagainya.

Komunikasi antar node dalam ROS dilakukan menggunakan tiga mekanisme utama:

- Topics → Digunakan untuk komunikasi asinkron, seperti pengiriman data sensor secara kontinu ke sistem.
- Services → Digunakan untuk komunikasi sinkron, misalnya memberikan perintah agar robot berpindah ke lokasi tertentu.
- Parameter Server → Berfungsi untuk menyimpan pengaturan global dalam sistem.

Sistem ini bersifat terdistribusi, sehingga beberapa node dapat dijalankan pada komputer yang berbeda dalam jaringan.

Keunggulan dan Tujuan ROS

Mengapa ROS begitu diminati dalam dunia robotika?

Tujuan utama ROS adalah menyederhanakan pengembangan robot dengan memungkinkan penggunaan kembali kode yang telah ada. Dengan demikian, ROS memfasilitasi kolaborasi antara peneliti dan pengembang melalui berbagi pustaka dan algoritma.

Beberapa keunggulan ROS meliputi:

- Ringan dan fleksibel, memungkinkan integrasi dengan framework lain seperti OpenRAVE dan Orocos.
- Mendukung berbagai bahasa pemrograman, termasuk Python, C++, dan Java.
- Memiliki framework bawaan untuk pengujian, yaitu rostest, yang mempermudah proses pengujian.
- Mendukung proyek skala besar dengan kemampuan sistem terdistribusi yang berjalan di beberapa komputer.

Karena itu, ROS sangat cocok digunakan dalam penelitian dan aplikasi industri robotika.

Perbandingan ROS dengan Framework Lain

Dibandingkan dengan framework lain seperti Player, ROS lebih sesuai untuk robot dengan kompleksitas tinggi yang memiliki banyak sensor dan aktuator. Player lebih cocok untuk robot dasar dengan fungsi mobile. Sementara itu, jika dibandingkan dengan Microsoft Robotics Studio, ROS memiliki keunggulan berupa komunitas open-source yang lebih besar dan dukungan multi-platform.

Integrasi ROS dengan Teknologi Lain

Salah satu keunggulan utama ROS adalah kemampuannya untuk berintegrasi dengan teknologi lain, seperti:

- OpenCV untuk pemrosesan gambar dan visi komputer.
- Simulator seperti Gazebo untuk menguji robot dalam lingkungan virtual sebelum diterapkan di dunia nyata.
- Orocos untuk mendukung aplikasi robotika waktu nyata dalam kebutuhan industri berperforma tinggi.

Dengan fleksibilitas ini, ROS dapat digunakan pada berbagai jenis robot, mulai dari drone, kendaraan otonom, hingga robot manipulator untuk aplikasi industri.

Sistem Operasi yang Mendukung ROS

Saat ini, ROS hanya kompatibel dengan sistem berbasis UNIX, seperti Ubuntu dan macOS.

Versi dan Distribusi ROS

Saya menggunakan distribusi ROS Noetic.

2. Chapter 2 – Getting started with ROS programming

- Akses referensi GitHub untuk bab 2 melalui tautan berikut:

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

- Unduh file dalam format ZIP, ekstrak isi file tersebut, kemudian buka folder chapter 2. Selanjutnya, pindahkan folder `mastering_ros_demo_pkg` ke direktori **Linux -> home -> nama -> catkin_ws -> src**.
- Masuk ke direktori **catkin_ws**, jalankan perintah **catkin_make**, lalu aktifkan pengaturan dengan menjalankan source **devel/setup.bash**.
- Di terminal pertama, jalankan perintah **roscore** untuk memulai layanan utama ROS.
- Di terminal kedua, gunakan perintah **roslaunch mastering_ros_demo_pkg demo_topic_publisher** untuk menjalankan topik penerbit.
- Di terminal ketiga, jalankan perintah **roslaunch mastering_ros_demo_pkg demo_topic_subscriber** untuk menjalankan topik pelanggan.

3. Chapter 3 – Working with ROS for 3d Modelling

- Akses referensi GitHub untuk bab 3 melalui tautan berikut:

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

- Unduh file ZIP dari repositori tersebut, kemudian ekstrak isinya. Setelah itu, buka folder chapter 3 dan pindahkan direktori `mastering_ros_robot_description_pkg` ke lokasi berikut: **Linux -> home -> nama -> catkin_ws -> src**.
- Akses direktori **catkin_ws** melalui terminal, lalu jalankan perintah **catkin_make** untuk membangun workspace. Setelah selesai, aktifkan lingkungan kerja dengan menggunakan perintah **source devel/setup.bash**.
- Untuk simulasi pertama, jalankan perintah berikut di terminal: **roslaunch mastering_ros_robot_description_pkg view_demo.launch**.
- Simulasi kedua dapat dijalankan dengan mengetikkan perintah **roslaunch mastering_ros_robot_description_pkg view_arm.launch** di terminal.
- Simulasi ketiga dilakukan dengan menjalankan perintah berikut: **roslaunch mastering_ros_robot_description_pkg view_mobile_robot.launch**.

4. Chapter 4 – Simulating Robots using ROS and Gazebo

- Buka referensi GitHub untuk bab 4 melalui tautan berikut:

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

- Unduh repositori dalam format ZIP, kemudian ekstrak isinya. Setelah file diekstrak, temukan folder chapter 4 dan pindahkan direktori `seven_dof_arm_gazebo` ke lokasi berikut: **Linux -> home -> nama -> catkin_ws -> src**.
- Masuk ke direktori **catkin_ws** melalui terminal, jalankan perintah **catkin_make** untuk membangun workspace. Setelah itu, aktifkan konfigurasi workspace dengan perintah **source devel/setup.bash**.
- Untuk menjalankan simulasi dof arm, gunakan perintah berikut di terminal:
roslaunch seven_dof_arm_gazebo seven_dof_arm_gazebo_control.launch
- Untuk menggerakkan lengan robot, jalankan perintah ini:

```
rostopic pub -s /seven_dof_arm/joint4_position_controller/command  
std_msgs/float64 "data
```