LAPORAN UJIAN AKHIR SEMESTER ROBOTIKA

Chapter 1- Introduction to ROS

Disusun Oleh :

Rai Barokah Utari-1103200066

PROGRAM STUDI TEKNIK KOMPUTER

FAKULTAS TEKNIK ELEKTRO

UNIVERSITAS TELKOM

2023

A. Creating a ROS package

Paket ROS adalah unit dasar yang dapat dibuat, dikembangkan, dan dirilis ke publik. Distribusi ROS yang digunakan saat ini adalah Noetic Ninjemys. Kami menggunakan sistem pembangunan catkin untuk menghasilkan target dari kode sumber teks, yang digunakan oleh pengguna akhir. Pada distribusi-distribusi ROS sebelumnya seperti Electric dan Fuerte, digunakan sistem pembangun rosbuild. Dengan kekurangan yang dimiliki oleh rosbuild, kemudian muncul catkin, yang mendekatkan sistem kompilasi ROS dengan Cross Platform Make (CMake). Hal ini memungkinkan paket-paket berbasis catkin dapat dipindahkan ke sistem operasi lain, seperti Windows, selama sistem operasi tersebut mendukung CMake dan Python.

1. Working with ROS topics

Topik digunakan sebagai metode komunikasi antara node-node ROS, memungkinkan mereka untuk berbagi aliran informasi yang kontinu yang dapat diterima oleh node-node lainnya. Pada bagian ini, kita akan mempelajari bagaimana topik bekerja. Kita akan membuat dua node ROS untuk memublikasikan sebuah topik dan berlangganan ke dalamnya. Buka folder mastering_ros_demo_pkg dan masuk ke subdirektori /src untuk kode sumber. demo_topic_publisher.cpp dan demo_topic_subscriber.cpp adalah dua set kode yang akan kita bahas.

2. Creating ROS nodes

Node pertama yang akan kita bahas adalah demo_topic_publisher.cpp. Node ini akan menerbitkan nilai integer pada sebuah topik yang disebut /numbers.

3. Building the nodes

Edit file CMakeLists.txt pada paket untuk mengompilasi dan membangun kode sumber. Buka mastering_ros_demo_pkg untuk melihat file CMakeLists.txt yang ada.

B. Adding custom .msg and .srv files

Di bagian ini, kita akan melihat bagaimana cara membuat pesan dan definisi layanan kustom dalam paket saat ini. Definisi pesan disimpan dalam file .msg, sementara definisi layanan disimpan dalam file .srv. Definisi ini memberitahu ROS tentang jenis data dan nama data yang akan dikirimkan dari node ROS. Ketika pesan kustom ditambahkan, ROS

akan mengonversi definisi tersebut menjadi kode C++ yang setara, yang dapat kita masukkan ke dalam node-node kita.

C. Working with ROS services

Di bagian ini, kita akan membuat node-node ROS yang dapat menggunakan definisi layanan yang sudah kita tentukan sebelumnya. Node layanan yang akan kita buat dapat mengirimkan pesan string sebagai permintaan ke server; kemudian, node server akan mengirimkan pesan lain sebagai respons.

D. Working with ROS actionlib

Dalam layanan ROS, pengguna mengatur interaksi permintaan/balasan antara dua node. Jika balasan memakan waktu lama atau server belum selesai, kita harus menunggu, memblokir aplikasi utama. Klien juga bisa memantau proses dari jarak jauh. Actionlib, alternatif dalam ROS, memungkinkan pembatalan dan pengiriman permintaan baru jika tidak selesai tepat waktu. Ini umum digunakan dalam navigasi robot. Spesifikasi aksi dalam actionlib disimpan dalam file .action dalam folder aksi paket ROS, termasuk tujuan, umpan balik, dan hasil yang berkaitan dengan tujuan, kemajuan, dan hasil operasi.