Chapter 2

Pertama tama kita harus menginstall ros-noetic dan rosdep jika sudah bisa kita lanjutkan ke langkah berikutnya

source /opt/ros/noetic/setup.bash

Sudo apt update

Jika terjadi error ketika sudo apt update kita lakukan ini

sudo apt clean

sudo rm -rf /var/lib/apt/lists/*

sudo apt update

Lalu pasang dependensi ini

sudo apt install git python3-pip python3-rosdep python3-catkin-tools

Lalu kita update rosdep:

sudo rosdep init

rosdep update

Jika terjadi error saat sudo rosdep init, lakukan langkah ini:

sudo rm /etc/ros/rosdep/sources.list.d/20-default.list

sudo rosdep init

rosdep update

Lalu kita mengclone repository dari github

cd ~/Downloads

git clone https://github.com/PacktPublishing/Mastering-ROS-for-Robotics-Programming-Third-edition.git repository

Pindahkan Chapter 2 ke dalam folder src:

cd ~/Downloads/repository

ls # Pastikan folder Chapter2 ada

mkdir src

mv Chapter2 src/

Kita install rosdepnya

rosdep install --from-paths src --ignore-src -r -y

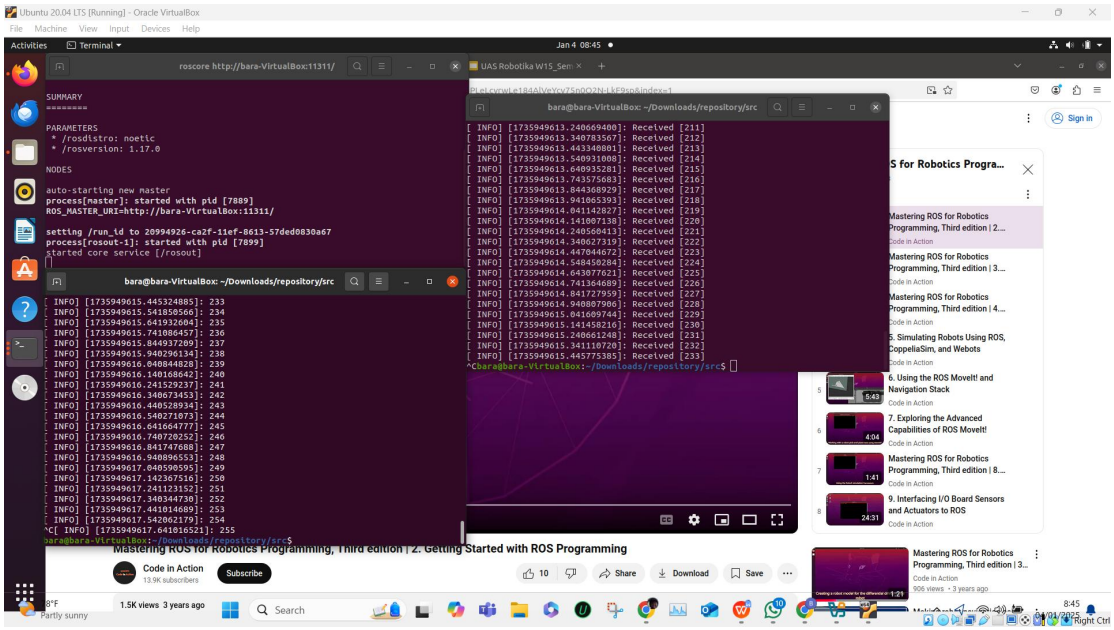
Build workspace menggunakan catkin_make jika belum ada bisa di install dulu

catkin_make

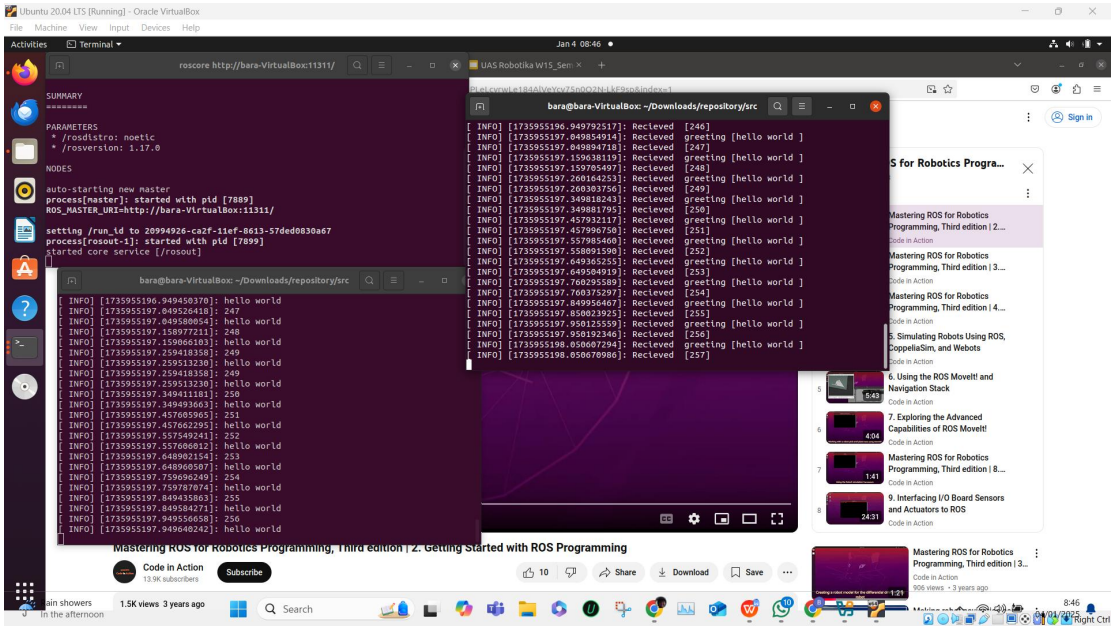
Aktifkan setup workspace:

source devel/setup.bash

1. Jalankan roscore (di terminal baru):
    a) Roscore
2. Jalankan publisher (di terminal baru):
    a) source ~/Downloads/repository/devel/setup.bash
    b) rosrun mastering_ros_demo_pkg demo_topic_publisher

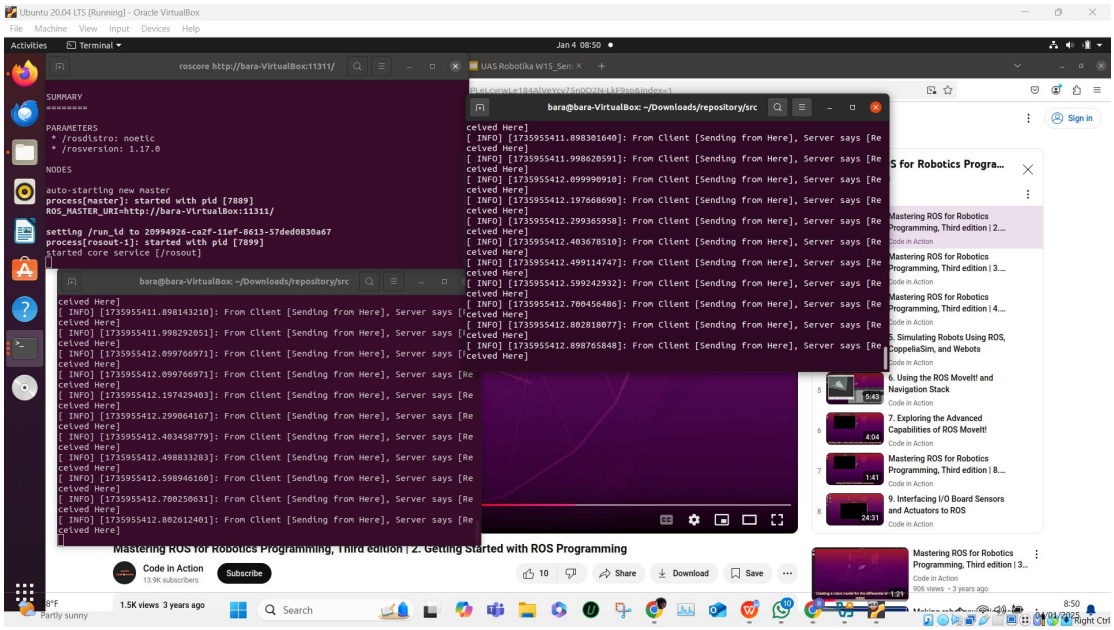


3. Jalankan subscriber (di terminal baru):
    a) source ~/Downloads/repository/devel/setup.bash
    b) rosrun mastering_ros_demo_pkg demo_topic_subscriber



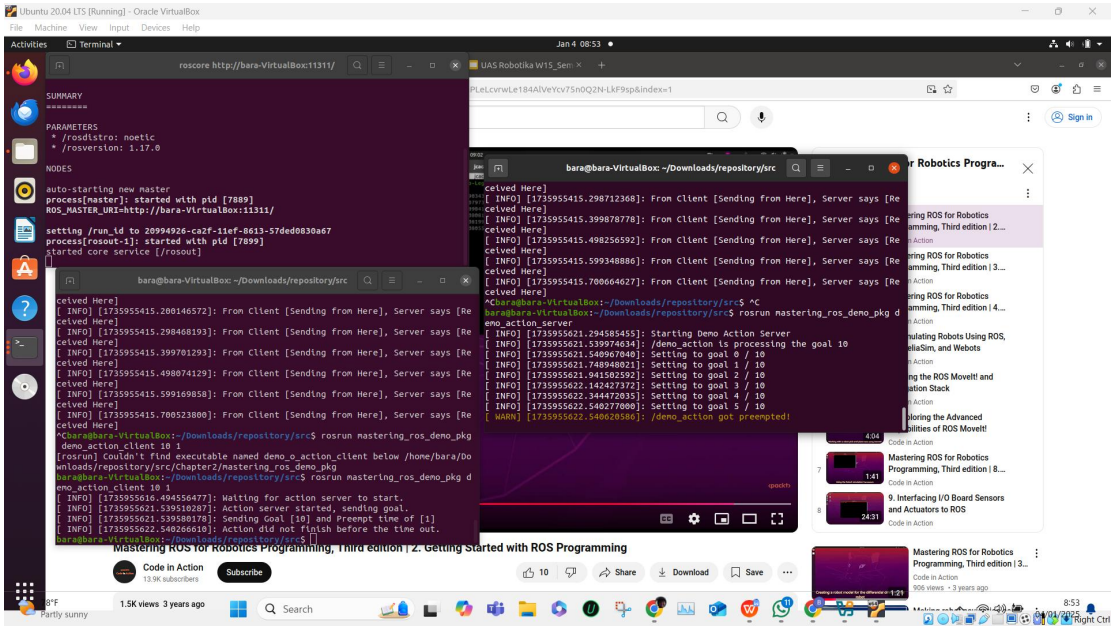
Lalu kita coba tambahkan .msg dan srv files dalam roscorenya

4. Jalankan rosrun
5. Lalu rosrun mastering_ros_demo_pkg demo_msg_publisher
6. rosrun mastering_ros_demo_pkg demo_msg_subscriber



Lalu kita coba tambahkan ROS services

7. rosrun mastering_ros_demo_pkg demo_service_server
8. rosrun mastering_ros_demo_pkg demservice_client



Rosrun action

9. Rosrun mastering_ros_demo_pkg demo_action_server
10. Rosrun mastering_ros_demo_pkg demo_action_client

Analisis Chapter 2

Dari kode instalasi dan penggunaan ROS (Robot Operating System) di atas, dapat dianalisis bahwa sistem ini menggunakan arsitektur publisher-subscriber yang merupakan salah satu pola komunikasi fundamental dalam ROS. Proses instalasi dimulai dengan pemasangan ROS Noetic (versi ROS yang dioptimalkan untuk Ubuntu 20.04) dan dilanjutkan dengan konfigurasi rosdep

yang berfungsi sebagai sistem manajemen dependensi. Struktur proyek menggunakan catkin sebagai build system, yang merupakan standar dalam ekosistem ROS untuk mengelola dan mengompilasi package. Penggunaan repository dari "Mastering ROS for Robotics Programming" menunjukkan bahwa ini adalah tutorial yang terstruktur untuk mempelajari konsep dasar ROS seperti nodes, topics, messages, dan services.

Implementasi yang ditunjukkan mencakup tiga komponen utama: publisher, subscriber, dan service. Publisher (demo_topic_publisher) bertindak sebagai pengirim data, sementara subscriber (demo_topic_subscriber) berperan sebagai penerima data, keduanya berkomunikasi melalui topic ROS. Penambahan custom message (.msg) dan service (.srv) files mendemonstrasikan fleksibilitas ROS dalam mendefinisikan format data yang dipertukarkan antar nodes. Service server dan client (demo_service_server dan demservice_client) mengilustrasikan pola komunikasi request-response, yang berbeda dengan pola publish-subscribe karena bersifat synchronous dan bi-directional. Keseluruhan struktur ini menunjukkan bagaimana ROS memfasilitasi pengembangan sistem robotika yang modular dan dapat diperluas.