

Nama : Irman Prayista

NIM : 1103210094

Kelas : TK-45-02

## Hasil Analisis Simulasi

### 1. Chapter 1 Introduction to ROS

*Robot Operating System* (ROS) adalah sebuah sistem perangkat lunak yang digunakan untuk membuat program robot. Sistem ini dirancang agar mudah digunakan dan memungkinkan bagian - bagian robot bekerja sama. ROS menyediakan berbagai alat dan pustaka yang membantu pengembangan, pembuatan, pengujian, dan menjalankan robot. Robot bisa melakukan berbagai tugas, seperti bergerak, mengenali benda, atau berinteraksi dengan lingkungan.

ROS menggunakan sistem komunikasi yang mempermudah bagian robot untuk saling terhubung. Komunikasi ini dilakukan melalui mekanisme sederhana, seperti pengirim – penerima pesan (*publisher – subscriber*) atau permintaan layanan (*service – client*). Keunggulan dari ROS ini, yaitu bersifat *open – source* sehingga banyak orang di seluruh dunia menggunakan ROS karena memiliki banyak fitur, seperti alat untuk membuat peta, navigasi, atau mengontrol lengan robot. ROS sering digunakan oleh peneliti, mahasiswa, dan pengembang untuk membantu dalam penelitian robot agar lebih cepat dan mudah, tanpa harus mulai dari awal.

### 2. Chapter 2 ROS Programming

Pada simulasi akan menggunakan mekanisme *publisher – subscriber* dengan ROS versi Noetic. Untuk sumber pengguna simulasi ini berasal dari pengguna github yang membahas *Mastering ROS for Robotics Programming*. Sebelum simulasi, perlu pemasangan Ubuntu dan ROS versi Noetic menggunakan WSL. Pada *Command Prompt*, gunakan perintah `wsl –install` untuk memasang WSL. Lalu pergi ke *Microsoft Store* dan pasang Ubuntu 20.04.6 LTS. Buka Ubuntu 20.04 LTS dan pasang ROS 1 dengan versi Noetic. Untuk sumbernya bisa menggunakan *Ubuntu install of ROS Noetic*. Setelah terpasang, masukkan perintah `roscore` pada tab pertama untuk server dan pada tab kedua masukkan perintah `roslaunch mastering_ros_demo_pkg demo_topic_publisher` yang digunakan untuk mengirim pesan ke suatu topik di ROS berdasarkan peristiwa tertentu. Hasilnya menghasilkan *output* angka dari 0 – tak terbatas. Untuk perintah selanjutnya pada tab ketiga yang berisi `roslaunch mastering_ros_demo_pkg demo_topic_subscriber` yang berfungsi untuk

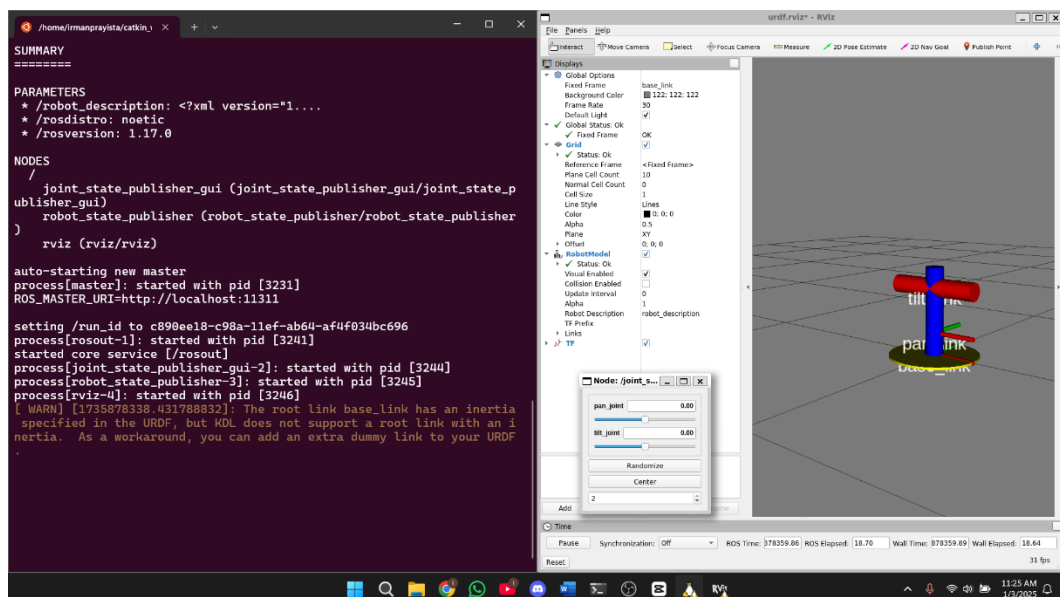
mendengarkan topik tertentu yang sudah dipublikasikan oleh *node* lain atau *publisher*. Hasil yang ditampilkan akan sesuai dengan *publisher* dari angka 0 – tak terbatas.

The screenshot shows two terminal windows. The left window displays the output of `roslaunch` for a ROS master, including parameters like `/rostdistro: noetic` and `/rosversion: 1.17.0`, and the start of the `process[rosout-1]` service. The right window shows a series of `Received [62]` through `Received [77]` messages, indicating data being received from a publisher.

### 3. Chapter 3 ROS with RVIZ

#### a. Menggunakan RVIZ

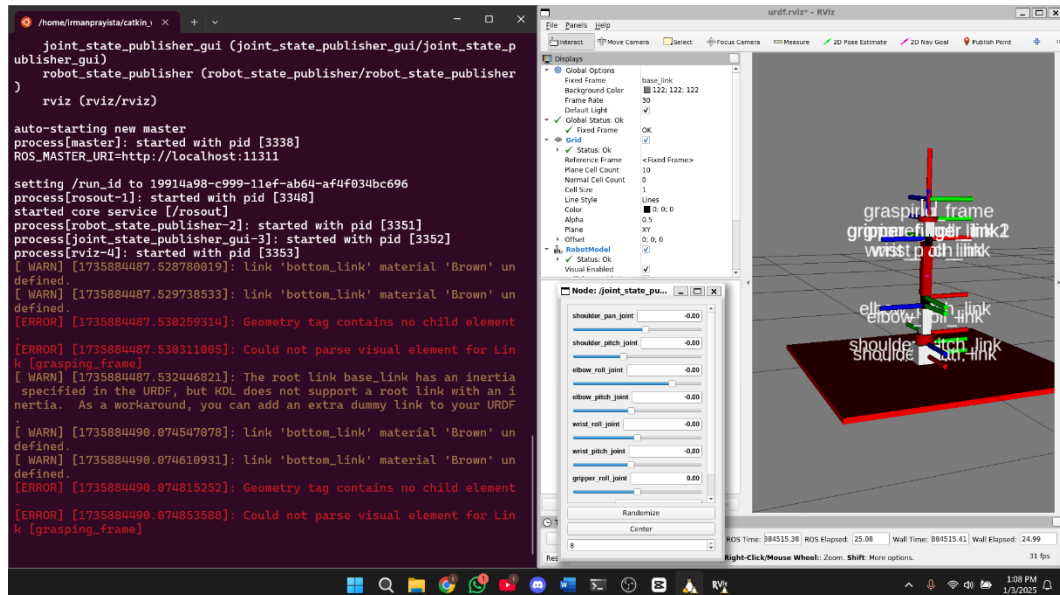
Pada simulasi ini akan menggunakan perintah **`roslaunch mastering_ros_robot_description_pkg view_demo.launch`** yang digunakan untuk memuat atau mengonfigurasi robot yang ingin dilihat atau diinteraksikan dengan RVIZ.



#### b. Menjelaskan Model Xacro dari Lengan Tujuh DOF

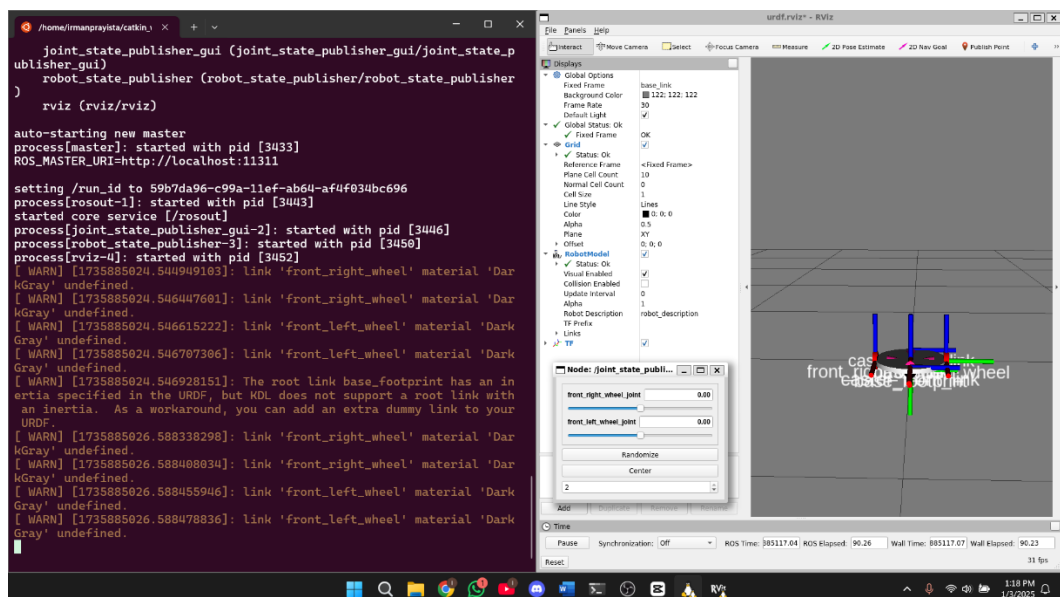
Pada simulasi ini akan menggunakan perintah **`roslaunch mastering_ros_robot_description_pkg view_arm.launch`** yang digunakan

memvisualisasikan model lengan robot dalam RVIZ dan terdapat *node* yang digunakan untuk memuat deskripsi URDF/Xacro dari lengan robot dan menghubungkan model robot dengan ROS Master. RVIZ digunakan sebagai alat untuk memvisualisasikan tampilan model lengan robot.



### c. Membuat Model Robot untuk Pergerakan Robot Menggunakan Penggerak Diferensial

Pada simulasi ini akan menggunakan perintah **roslaunch mastering\_ros\_robot\_description\_pkg view\_mobile\_robot.launch** yang digunakan untuk memuat model 3D robot bergerak yang didefinisikan dalam *file* URDF atau Xacro. Robot ini memiliki roda yang memungkinkan pergerakan di lingkungan RVIZ.



#### 4. Chapter 4 ROS and Gazebo

Pada simulasi ini menggunakan perintah **roslaunch seven\_dof\_arm\_gazebo seven\_dof\_arm\_gazebo\_control.launch** yang digunakan untuk menjalankan *file launch* yang bernama *seven\_dof\_arm\_gazebo\_control.launch* dari *package* ROS bernama *seven\_dof\_arm\_gazebo*. Ketika perintah ini dijalankan, akan meluncurkan simulasi gazebo yang digunakan untuk memulai simulasi robot di lingkungan gazebo, mengaktifkan kontroler robot yang akan mengonfigurasi dan menjalankan kontroler yang diperlukan untuk mengontrol lengan robot dalam simulasi, dan menghubungkan dengan ROS yang digunakan untuk memastikan bahwa mode yang terkait dengan simulasi, seperti *node* untuk fisika Gazebo, *node* kontroler robot, dan *node* komunikasi data antar ROS.

