

Nama : Ketut Satria Wibisana
NIM : 1103213148
Kelas : TK-45-G09

Laporan Hasil Analisis Simulasi Chapter 1-4

1. Chapter 1 – Introduction to ROS

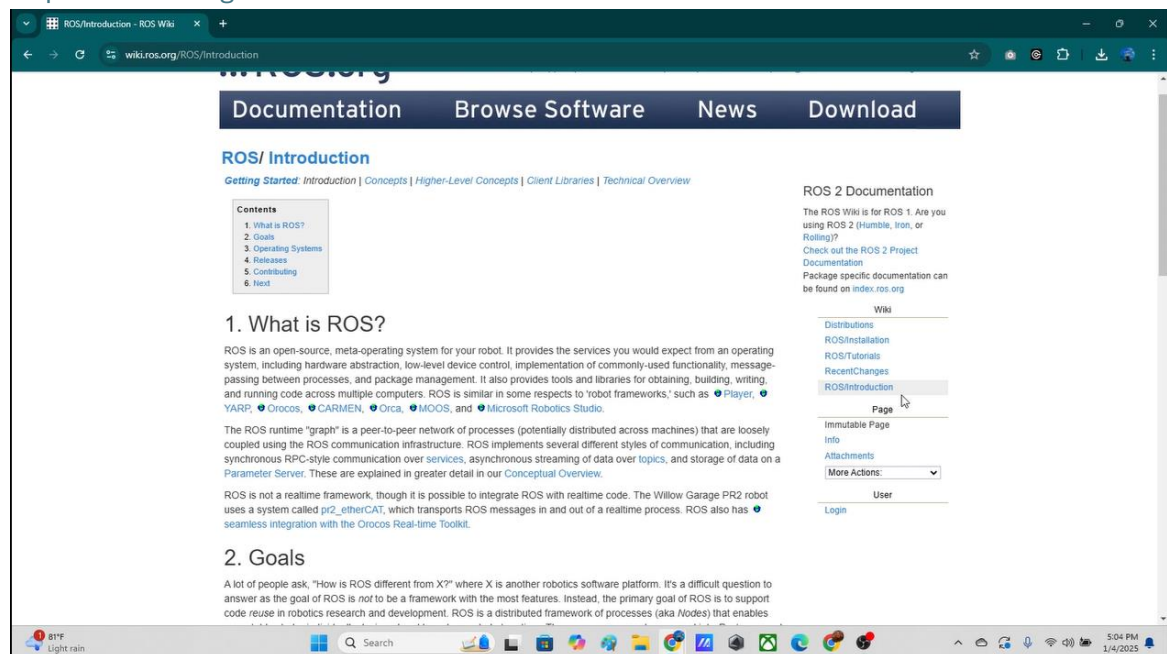
Robot Operating System (ROS) adalah perangkat lunak yang dirancang untuk mempermudah pengembangan dan pemrograman sistem robotika. Platform ini dirancang untuk memberikan kemudahan penggunaan sekaligus memungkinkan integrasi berbagai komponen robot secara efisien. ROS menyediakan berbagai pustaka dan alat bantu yang berguna dalam proses pengembangan, pengujian, hingga pengoperasian robot. Dengan ROS, robot dapat melakukan berbagai fungsi, seperti navigasi, deteksi objek, hingga berinteraksi dengan lingkungannya.

ROS menggunakan sistem komunikasi yang memungkinkan komponen-komponen dalam robot saling berkolaborasi secara efektif. Komunikasi ini dilakukan melalui mekanisme sederhana, seperti model pengiriman dan penerimaan pesan (publisher-subscriber) atau layanan berbasis permintaan (service-client). Salah satu keunggulan utama ROS adalah sifatnya yang open-source, sehingga pengembang dan peneliti dari berbagai belahan dunia dapat memanfaatkannya. Selain itu, ROS dilengkapi dengan fitur-fitur canggih, seperti kemampuan membuat peta, navigasi robot, hingga pengendalian lengan robot.

Dengan berbagai kelebihan tersebut, ROS telah menjadi alat yang sangat berguna bagi kalangan akademisi, mahasiswa, dan pengembang teknologi untuk mempercepat penelitian serta pengembangan robot tanpa harus membuat sistem dari awal.

Informasi lebih lanjut mengenai pengenalan ROS dapat diakses melalui tautan berikut:

<http://wiki.ros.org/ROS/Introduction>



2. Chapter 2 – Getting started with ROS programming

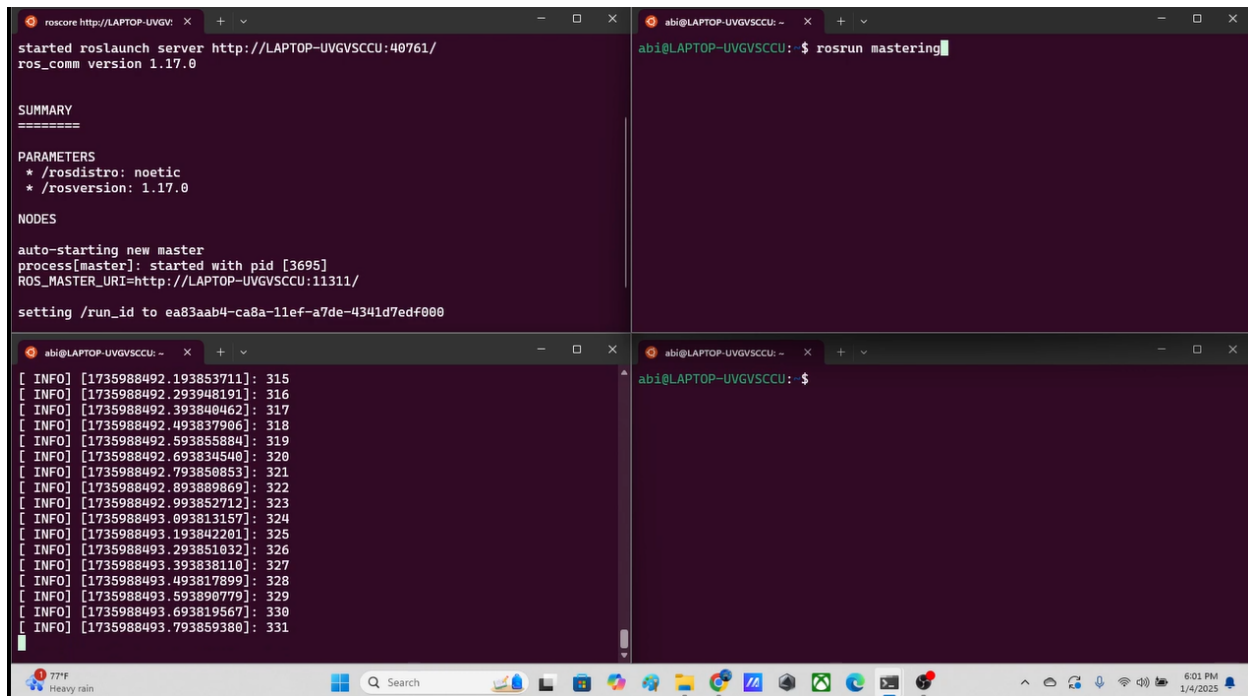
Simulasi yang digunakan dalam penelitian ini memanfaatkan mekanisme komunikasi berbasis publisher-subscriber dengan menggunakan ROS versi Noetic. Sumber referensi utama berasal dari repositori GitHub yang mengulas Mastering ROS for Robotics Programming. Sebelum simulasi dapat dijalankan, diperlukan penginstalan sistem operasi Ubuntu dan ROS Noetic melalui Windows Subsystem for Linux (WSL).

Proses instalasi WSL dimulai dengan menjalankan perintah `wsl --install` melalui Command Prompt. Setelah itu, unduh dan pasang Ubuntu 20.04.6 LTS dari Microsoft Store. Selanjutnya, buka terminal Ubuntu dan lakukan instalasi ROS versi Noetic berdasarkan panduan resmi yang disediakan oleh Ubuntu.

Setelah ROS berhasil terpasang, aktifkan server ROS dengan menjalankan perintah `roscore` pada terminal pertama. Kemudian, pada terminal kedua, jalankan perintah `roslaunch mastering_ros_demo_pkg demo_topic_publisher`. Perintah ini akan mengirimkan pesan ke topik tertentu dalam ROS berdasarkan peristiwa yang terjadi, menghasilkan keluaran berupa angka yang terus bertambah, dimulai dari 0.

Pada terminal ketiga, jalankan perintah `roslaunch mastering_ros_demo_pkg demo_topic_subscriber`. Perintah ini digunakan untuk menerima pesan yang telah dipublikasikan oleh node publisher. Hasil yang diterima akan sesuai dengan data yang dikirimkan oleh publisher, berupa angka yang meningkat secara bertahap dari 0 hingga tak terbatas.

Berikut adalah hasil simulasi yang telah dilakukan:



The screenshot displays four terminal windows from a Windows Subsystem for Linux (WSL) environment. The top-left window shows the output of the `roscore` command, indicating the start of the ROS master server. The top-right window shows the command `roslaunch mastering_ros_demo_pkg demo_topic_publisher` being executed. The bottom-left window shows a list of INFO messages from the publisher node, displaying a sequence of numbers from 315 to 331. The bottom-right window shows the command `roslaunch mastering_ros_demo_pkg demo_topic_subscriber` being executed. The Windows taskbar at the bottom shows the system clock as 6:01 PM on 1/4/2025.

```
roscore http://LAPTOP-UVGV: x + -
started roslaunch server http://LAPTOP-UVGVSCCU:40761/
ros_comm version 1.17.0

SUMMARY
=====

PARAMETERS
* /rostdistro: noetic
* /rosversion: 1.17.0

NODES

auto-starting new master
process[master]: started with pid [3695]
ROS_MASTER_URI=http://LAPTOP-UVGVSCCU:11311/

setting /run_id to ea83aab4-ca8a-11ef-a7de-4341d7edf000

abi@LAPTOP-UVGVSCCU: ~
abi@LAPTOP-UVGVSCCU: $ roslaunch mastering

abi@LAPTOP-UVGVSCCU: ~
abi@LAPTOP-UVGVSCCU: $

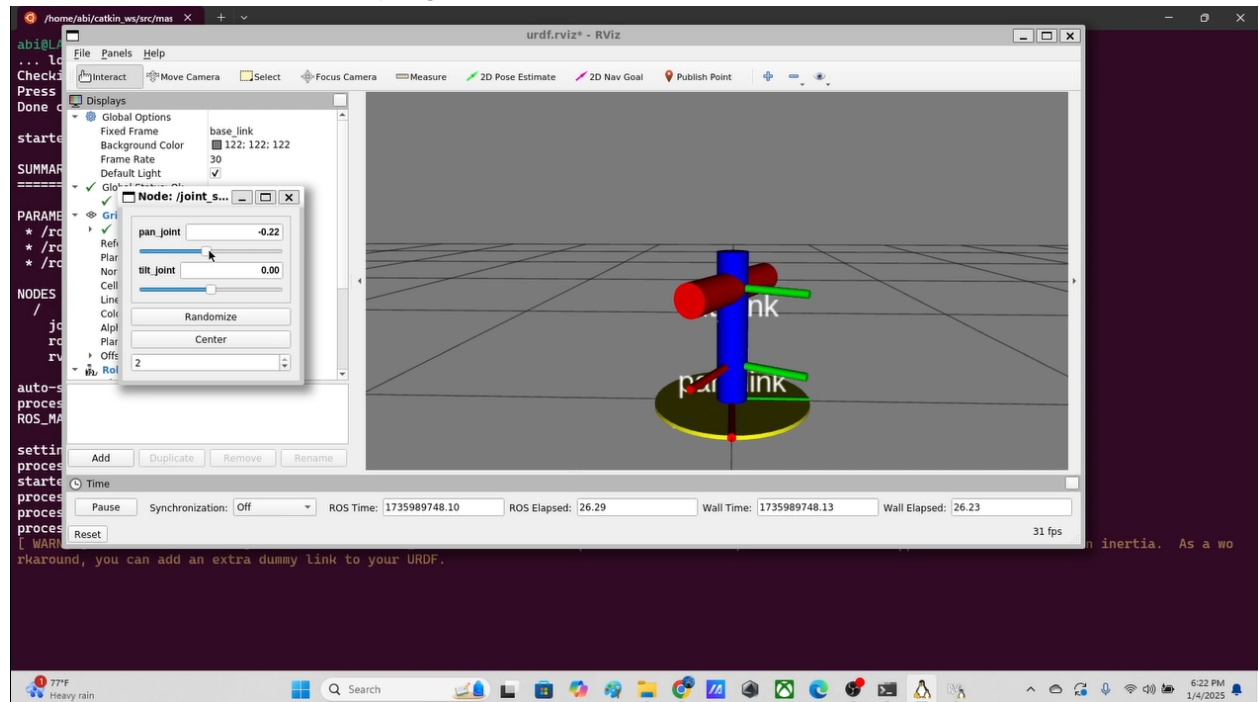
[ INFO] [1735988492.193853711]: 315
[ INFO] [1735988492.293948191]: 316
[ INFO] [1735988492.393840462]: 317
[ INFO] [1735988492.493837906]: 318
[ INFO] [1735988492.593855884]: 319
[ INFO] [1735988492.693834540]: 320
[ INFO] [1735988492.793850853]: 321
[ INFO] [1735988492.893889869]: 322
[ INFO] [1735988492.993852712]: 323
[ INFO] [1735988493.093813157]: 324
[ INFO] [1735988493.193842201]: 325
[ INFO] [1735988493.293851032]: 326
[ INFO] [1735988493.393838110]: 327
[ INFO] [1735988493.493817899]: 328
[ INFO] [1735988493.593890779]: 329
[ INFO] [1735988493.693819567]: 330
[ INFO] [1735988493.793859380]: 331
```

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

a. Penggunaan RViz

Pada tahap ini, simulasi dilakukan dengan menjalankan perintah `roslaunch mastering_ros_robot_description_pkg view_demo.launch`. Perintah ini digunakan untuk memuat dan mengatur konfigurasi robot sehingga dapat divisualisasikan serta dioperasikan dalam RViz.

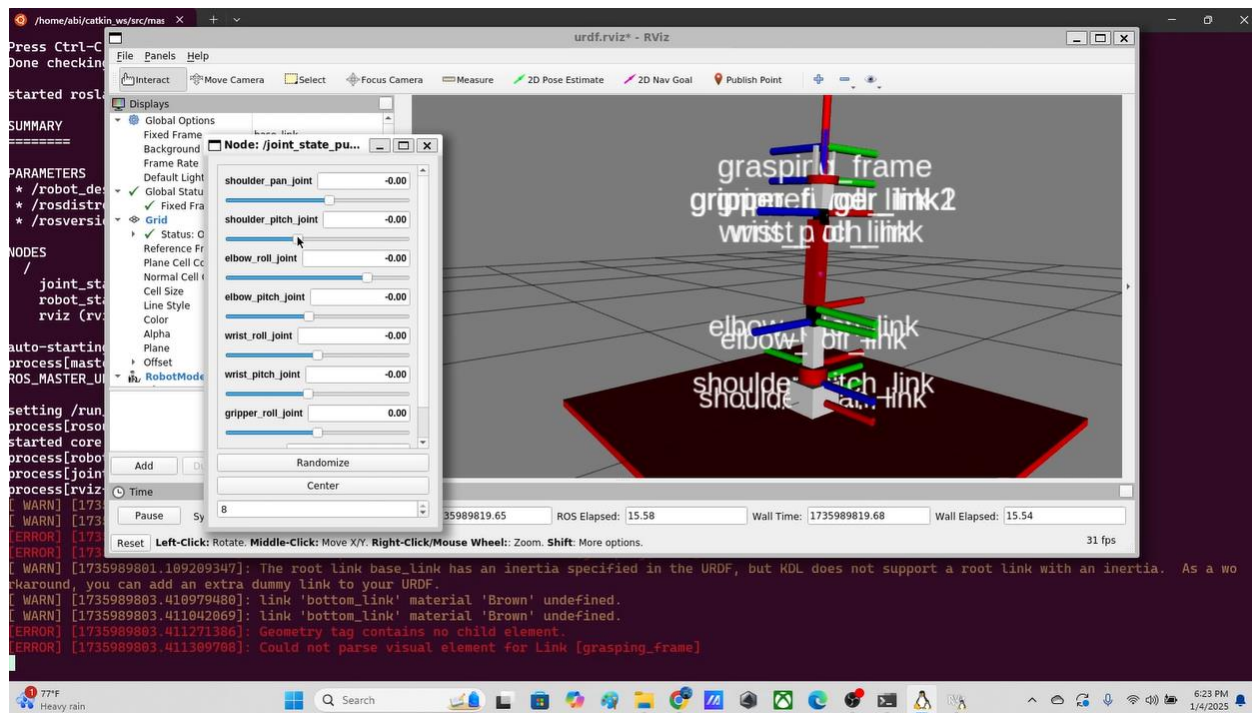
Berikut adalah hasil simulasi yang telah dilaksanakan:



b. Model Xacro untuk Lengan Robot dengan 7 Derajat Kebebasan (DOF)

Pada simulasi ini, perintah `roslaunch mastering_ros_robot_description_pkg view_arm.launch` digunakan untuk memuat dan memvisualisasikan model lengan robot dengan 7 DOF di RViz. Perintah tersebut memuat deskripsi URDF/Xacro dari model lengan robot dan menghubungkannya dengan ROS Master. Dengan menggunakan RViz, pengguna dapat mengamati representasi visual model lengan robot sekaligus melakukan konfigurasi sesuai kebutuhan.

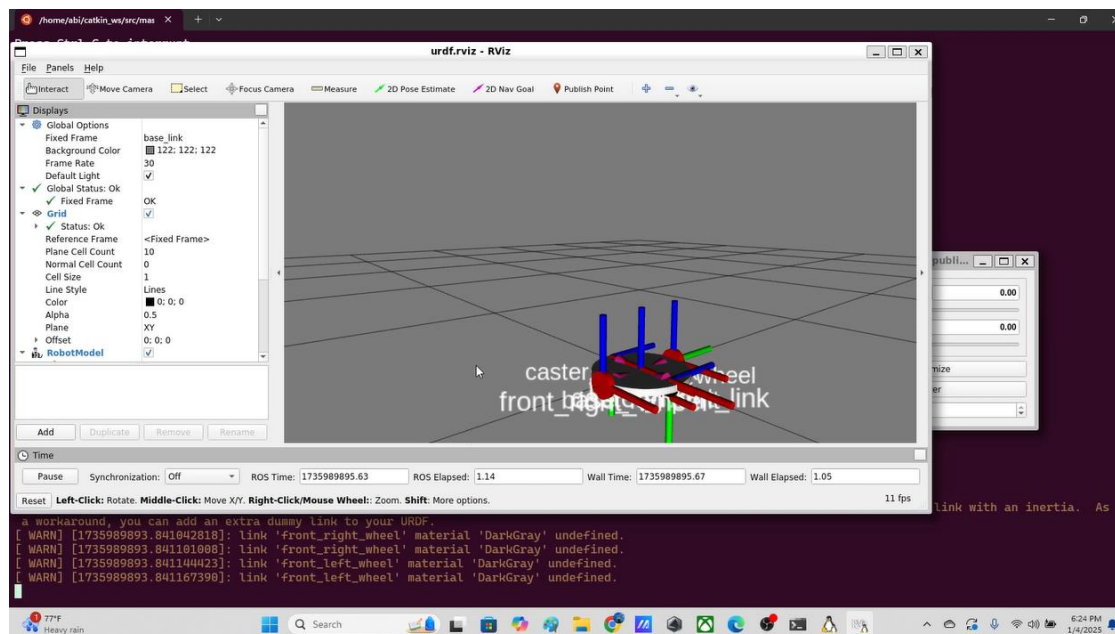
Berikut adalah hasil simulasi yang telah dilaksanakan:



c. Simulasi Model Robot dengan Penggerak Diferensial

Untuk menampilkan simulasi model robot bergerak di RViz, perintah `roslaunch mastering_ros_robot_description_pkg view_mobile_robot.launch` digunakan. Perintah ini bertujuan untuk memuat model robot bergerak dalam format 3D yang telah didefinisikan menggunakan file URDF atau Xacro. Model robot yang dimuat dilengkapi dengan roda, sehingga memungkinkan simulasi pergerakan robot di lingkungan visual RViz.

Berikut adalah hasil simulasi yang telah dilaksanakan:



4. Chapter 4 – Simulating Robots using ROS and Gazebo

Pada tahap ini, simulasi dilakukan dengan menjalankan perintah `roslaunch seven_dof_arm_gazebo seven_dof_arm_gazebo_control.launch`. Perintah ini digunakan untuk mengeksekusi file peluncuran bernama `seven_dof_arm_gazebo_control.launch` yang terdapat dalam paket ROS `seven_dof_arm_gazebo`.

Ketika perintah tersebut dijalankan, beberapa proses utama akan dilakukan, di antaranya:

- Memulai Simulasi di Gazebo: Lingkungan simulasi Gazebo diaktifkan untuk menguji pergerakan robot serta interaksinya dalam ruang virtual.
- Mengaktifkan Kontroler Robot: Kontroler yang dibutuhkan untuk menggerakkan lengan robot akan dikonfigurasi dan diaktifkan selama simulasi berlangsung.
- Pengintegrasian dengan ROS: Semua komponen simulasi, termasuk node fisika Gazebo, node kontroler, dan node komunikasi ROS, akan terhubung untuk memastikan sistem berjalan dengan baik.

Untuk menggerakkan robot, digunakan perintah berikut:

```
rostopic pub -s /seven_dof_arm/joint4_position_controller/command std_msgs/float64  
"data: 1.0"
```

Berikut adalah hasil simulasi yang telah dilaksanakan:

