

# HASIL ANALISIS

## Chapter 1 – Introduction to ROS

ROS atau Robot Operating System adalah kerangka kerja perangkat lunak open-source yang digunakan untuk mengembangkan aplikasi robotika. ROS membantu pengembang mengontrol robot dengan lebih mudah, menyediakan tools dan library untuk berbagai keperluan seperti pemetaan, navigasi, dan manipulasi objek

Kelebihan ROS:

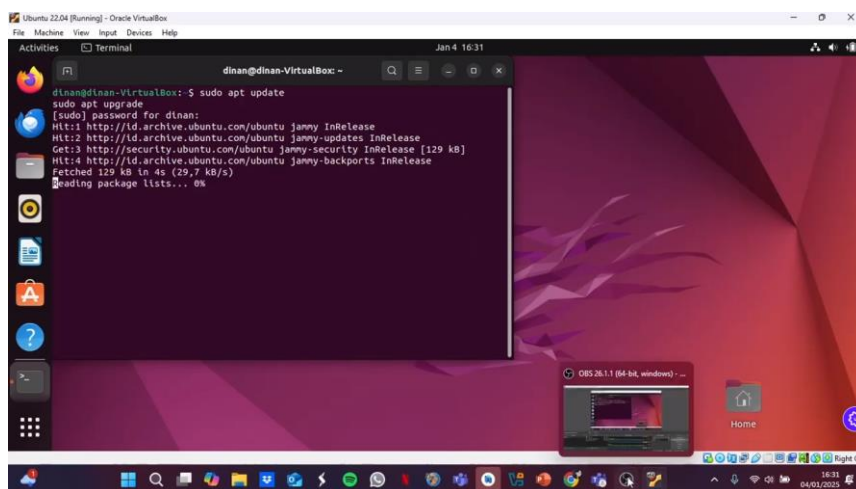
- Mendukung berbagai jenis robot.
- Komunitas global yang besar dan aktif.
- Banyak library siap pakai.
- Mudah diintegrasikan dengan sensor, aktuator, dan algoritma AI.

Distribusi ROS:

- ROS 1: Melodic, Noetic.
- ROS 2: Foxy, Galactic, Humble, dan Iron.

CARA INSTALL ROS :

- Upgrade sistem



- Tambahkan repository ROS2

```

dinan@dinan-VirtualBox: ~
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
dinan@dinan-VirtualBox: ~$ sudo apt install software-properties-common
[sudo] password for dinan:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
software-properties-common is already the newest version (0.99.22.9).
software-properties-common set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Adding component(s) 'universe' to all repositories.
Press [ENTER] to continue or Ctrl-C to cancel.
Hit:1 http://id.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://id.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://id.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:4 http://id.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Me
tadata [103 kB]
Get:5 http://id.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 DEP-11
Metadata [212 B]
Get:6 http://id.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Me
tadata [356 kB]
0% [6 Components:amd64 319 kB/356 kB 90%] [Connecting to security.ubuntu.com]

```

- Tambahkan Kunci GPG ROS2

```

dinan@dinan-VirtualBox: ~
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  curl
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 194 kB of archives.
After this operation, 455 kB of additional disk space will be used.
Get:1 http://id.archive.ubuntu.com/ubuntu jammy-updates/main amd64 curl amd64 7.
81.0-1ubuntu1.20 [194 kB]
Fetched 194 kB in 3s (75.6 kB/s)
Selecting previously unselected package curl.
(Reading database ... 207671 files and directories currently installed.)
Preparing to unpack .../curl_7.81.0-1ubuntu1.20_amd64.deb ...
Unpacking curl (7.81.0-1ubuntu1.20) ...
Setting up curl (7.81.0-1ubuntu1.20) ...
Processing triggers for man-db (2.10.2-1) ...
dinan@dinan-VirtualBox: ~$ sudo curl -sSL https://raw.githubusercontent.com/ros/r
osdistro/master/ros.key -o /usr/share/keyrings/ros-archive-keyring.gpg
dinan@dinan-VirtualBox: ~$

```

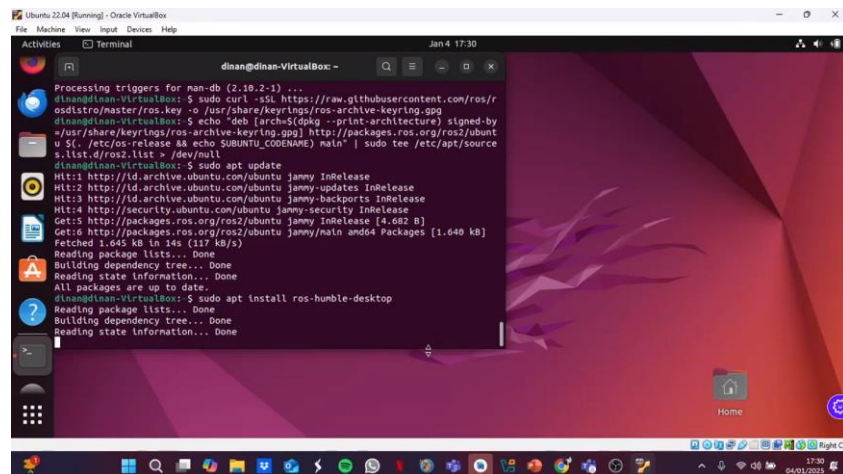
- Tambahkan repository ROS2 ke sourcelist

```

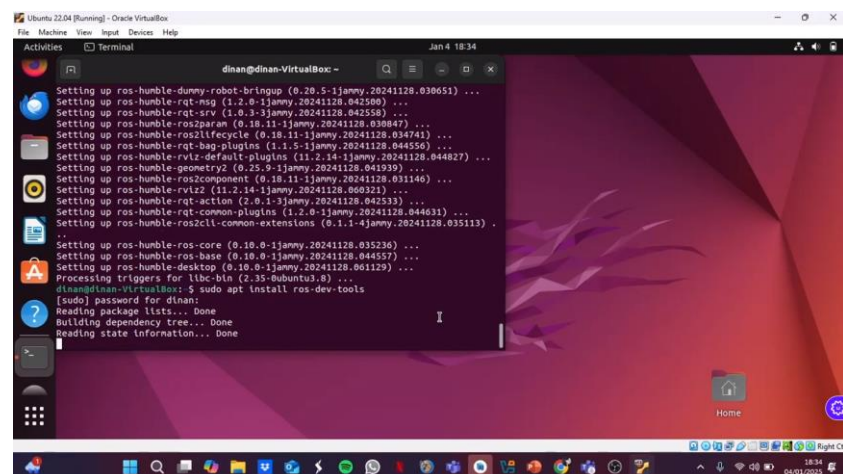
dinan@dinan-VirtualBox: ~
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  curl
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 194 kB of archives.
After this operation, 455 kB of additional disk space will be used.
Get:1 http://id.archive.ubuntu.com/ubuntu jammy-updates/main amd64 curl amd64 7.
81.0-1ubuntu1.20 [194 kB]
Fetched 194 kB in 3s (75.6 kB/s)
Selecting previously unselected package curl.
(Reading database ... 207671 files and directories currently installed.)
Preparing to unpack .../curl_7.81.0-1ubuntu1.20_amd64.deb ...
Unpacking curl (7.81.0-1ubuntu1.20) ...
Setting up curl (7.81.0-1ubuntu1.20) ...
Processing triggers for man-db (2.10.2-1) ...
dinan@dinan-VirtualBox: ~$ sudo curl -sSL https://raw.githubusercontent.com/ros/r
osdistro/master/ros.key -o /usr/share/keyrings/ros-archive-keyring.gpg
dinan@dinan-VirtualBox: ~$ echo "deb [arch=$(dpkg --print-architecture)] signed-by
=/usr/share/keyrings/ros-archive-keyring.gpg http://packages.ros.org/ros2/ubuntu
u $(. /etc/os-release && echo $UBUNTU_CODENAME) main" | sudo tee /etc/apt/source
s.list.d/ros2.list > /dev/null
dinan@dinan-VirtualBox: ~$

```

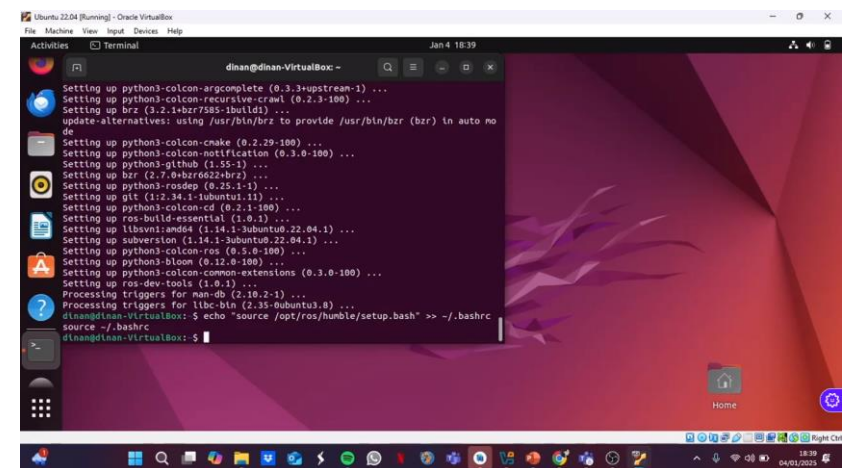
- Install ROS2 Humble



- Install Development Tools



- Tambahkan Setup ROS2 ke ~/.bashrc



## Chapter 2 – Getting Started With ROS Programming

### Simulasi Pertama : Publisher dan Subscriber dengan Topic

- Terminal 1: `ros2 run mastering_ros_demo_pkg demo_topic_publisher`  
Tujuan: Menjalankan node yang mempublikasikan pesan ke sebuah topic tertentu.
- Terminal 2: `ros2 run mastering_ros_demo_pkg demo_topic_subscriber`  
Tujuan: Menjalankan node yang berlangganan ke topic yang sama untuk menerima dan memproses pesan.

Hasil: Subscriber akan mencetak atau memproses pesan yang diterima dari publisher.

### Simulasi Kedua : Publisher dan Subscriber dengan Custom Message

- Terminal 1: `ros2 run mastering_ros_demo_pkg demo_msg_publisher`  
Tujuan: Memublikasikan pesan khusus (custom message) yang didefinisikan dalam package ini ke sebuah topic.
- Terminal 2: `ros2 run mastering_ros_demo_pkg demo_msg_subscriber`  
Tujuan: Berlangganan ke topic yang sama untuk menerima dan memproses pesan khusus tersebut.

Hasil: Anda dapat melihat demonstrasi penggunaan custom message di ROS 2.

### Simulasi Ketiga : Service Server dan Client

- Terminal 1: `ros2 run mastering_ros_demo_pkg demo_service_server`  
Tujuan: Menjalankan node service server yang menyediakan sebuah layanan (service).
- Terminal 2: `ros2 run mastering_ros_demo_pkg demo_service_client`  
Tujuan: Menjalankan node service client untuk memanggil layanan dari server.

Hasil: Client akan mengirim permintaan ke server, dan server akan merespons sesuai dengan logika yang telah didefinisikan.

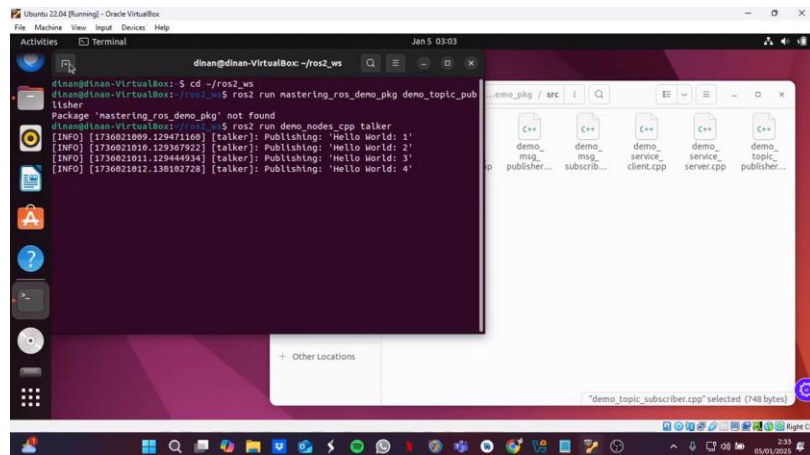
### Simulasi Keempat : Action Client dan Server

- Terminal 1: `ros2 run mastering_ros_demo_pkg demo_action_client 10 1`  
Tujuan: Menjalankan action client yang meminta sebuah aksi dengan parameter (misalnya 10 dan 1 sebagai input).
- Terminal 2: `ros2 run mastering_ros_demo_pkg demo_action_server`  
Tujuan: Menjalankan action server yang menerima permintaan dari client dan mengeksekusi aksi sesuai logika yang telah didefinisikan.

Parameter 10 dan 1 mungkin mengacu pada jumlah tugas dan interval waktu eksekusi. Client mengirimkan permintaan ke server, yang kemudian memproses aksi dan memberikan hasil atau pembaruan status.

Hasil: Action server akan melaporkan status eksekusi atau hasil akhir kepada client.

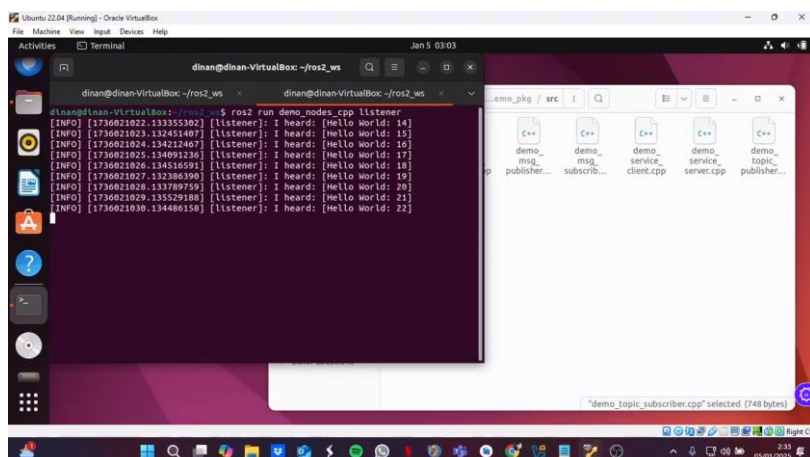
Talker :



The screenshot shows a terminal window titled 'dinan@dinan-VirtualBox: ~/ros2\_ws'. The user has run the command `ros2 run demo_nodes_cpp talker`. The output shows the package 'demo\_nodes\_cpp' being loaded and the 'talker' node publishing 'Hello World' messages at regular intervals. The messages are: 'Hello World: 1', 'Hello World: 2', 'Hello World: 3', and 'Hello World: 4'. A file explorer window is also open, showing the contents of the 'demo\_pkg' directory, including 'demo\_msg\_publisher.cpp', 'demo\_msg\_subscriber.cpp', 'demo\_service\_client.cpp', 'demo\_service\_server.cpp', and 'demo\_topic\_publisher.cpp'.

```
dinan@dinan-VirtualBox: ~/ros2_ws
$ ros2 run demo_nodes_cpp talker
Package 'demo_nodes_cpp' not found
dinan@dinan-VirtualBox: ~/ros2_ws$ ros2 run demo_nodes_cpp talker
[INFO] [1736021009.129471160]: [talker]: Publishing: 'Hello World: 1'
[INFO] [1736021010.129367922]: [talker]: Publishing: 'Hello World: 2'
[INFO] [1736021011.129444934]: [talker]: Publishing: 'Hello World: 3'
[INFO] [1736021012.130102758]: [talker]: Publishing: 'Hello World: 4'
```

Listener :



The screenshot shows a terminal window titled 'dinan@dinan-VirtualBox: ~/ros2\_ws'. The user has run the command `ros2 run demo_nodes_cpp listener`. The output shows the package 'demo\_nodes\_cpp' being loaded and the 'listener' node receiving 'Hello World' messages. The messages are: 'Hello World: 14', 'Hello World: 15', 'Hello World: 16', 'Hello World: 17', 'Hello World: 18', 'Hello World: 19', 'Hello World: 20', 'Hello World: 21', and 'Hello World: 22'. A file explorer window is also open, showing the contents of the 'demo\_pkg' directory, including 'demo\_msg\_publisher.cpp', 'demo\_msg\_subscriber.cpp', 'demo\_service\_client.cpp', 'demo\_service\_server.cpp', and 'demo\_topic\_publisher.cpp'.

```
dinan@dinan-VirtualBox: ~/ros2_ws
$ ros2 run demo_nodes_cpp listener
[INFO] [1736021022.133553892]: [listener]: I heard: 'Hello World: 14'
[INFO] [1736021023.133551407]: [listener]: I heard: 'Hello World: 15'
[INFO] [1736021024.134212467]: [listener]: I heard: 'Hello World: 16'
[INFO] [1736021025.134091236]: [listener]: I heard: 'Hello World: 17'
[INFO] [1736021026.134516591]: [listener]: I heard: 'Hello World: 18'
[INFO] [1736021027.132386390]: [listener]: I heard: 'Hello World: 19'
[INFO] [1736021028.133789759]: [listener]: I heard: 'Hello World: 20'
[INFO] [1736021029.135529188]: [listener]: I heard: 'Hello World: 21'
[INFO] [1736021030.134486158]: [listener]: I heard: 'Hello World: 22'
```

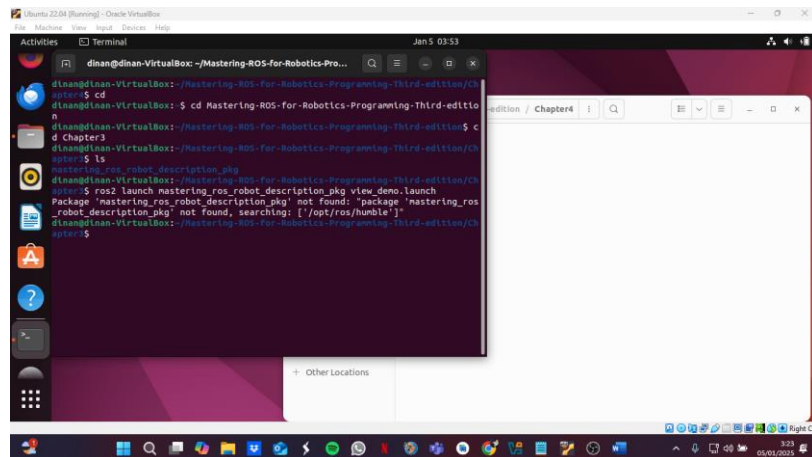
# Chapter 3 – Working with ROS for 3D Modeling

## Simulasi Pertama :

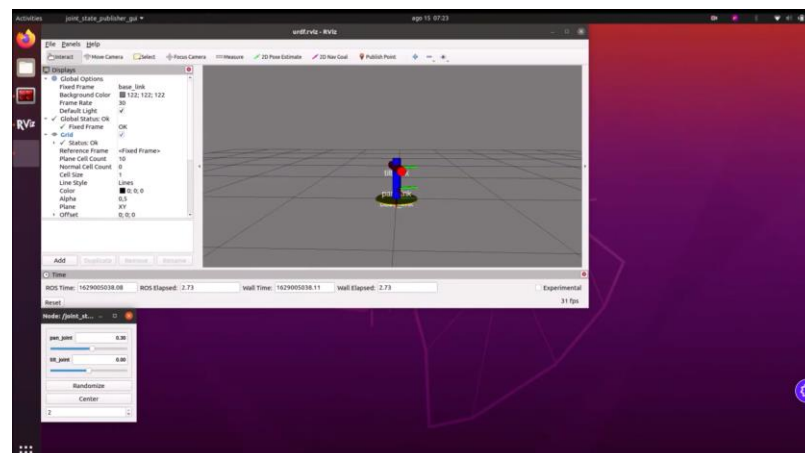
Buka Terminal , Jalankan Perintah :

```
ros2 launch mastering_ros_robot_description_pkg view_demo.launch
```

Output Saya (Belum Berhasil) :



Output Seharusnya :



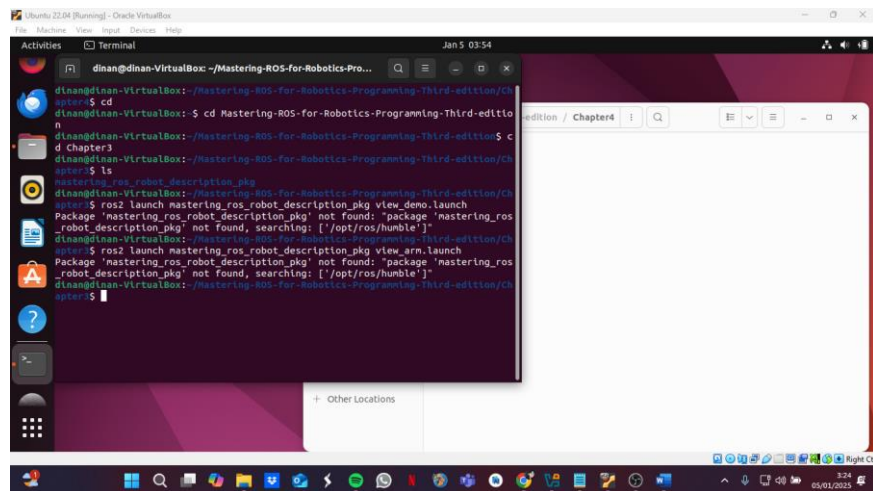


## Simulasi Kedua :

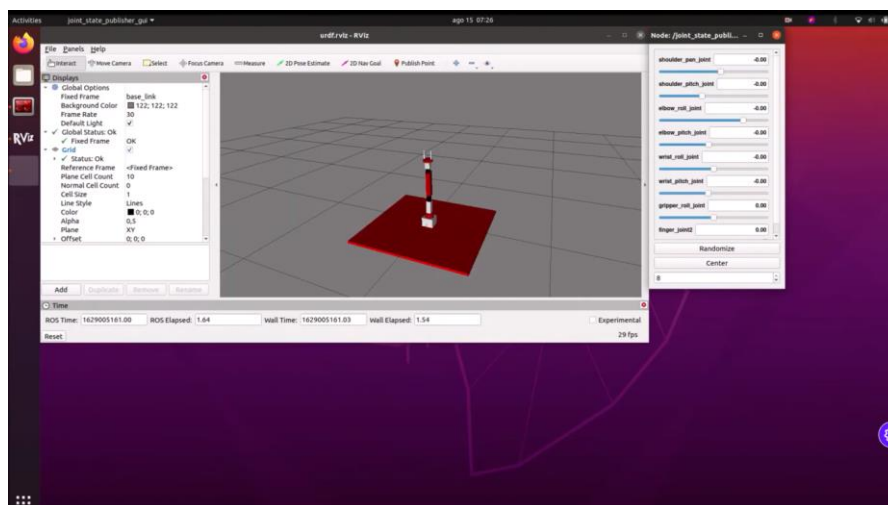
Buka Terminal , Jalankan Perintah :

```
ros2 launch mastering_ros_robot_description_pkg view_arm.launch
```

Output Saya (Belum Berhasil) :



Output Seharusnya :

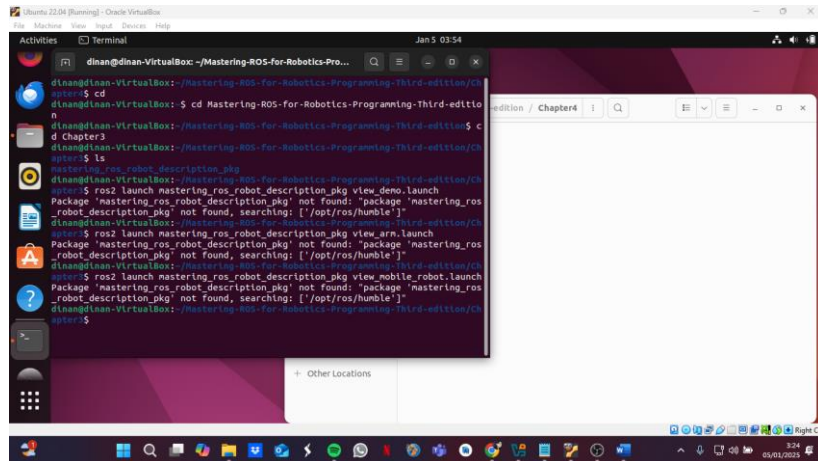


## Simulasi Ketiga :

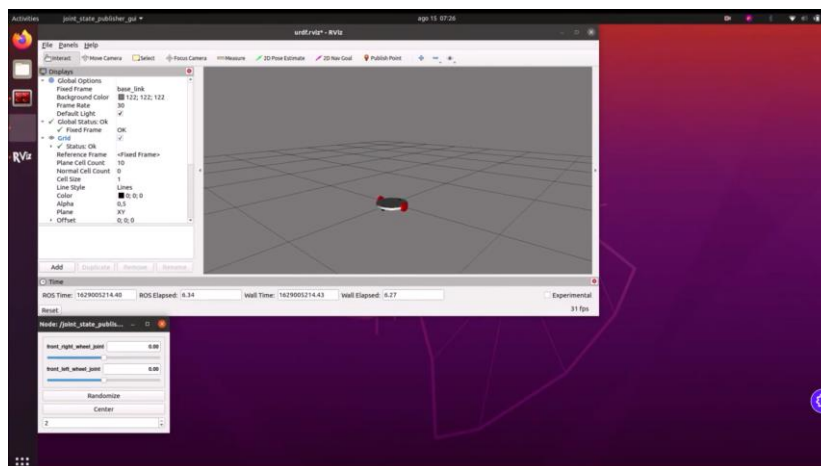
Buka Terminal , Jalankan Perintah :

```
ros2 launch masterling_ros_robot_description_pkg view_mobile_robot.launch
```

Output Saya (Belum Berhasil) :



Output Seharusnya :



Dari ketiga demo tersebut,

- perintah pertama menampilkan visualisasi 3D robot model di Rviz
- perintah kedua menampilkan xacro model dari lengan seven-DOF
- perintah ketiga membuat model robot untuk differential drive mobile robot

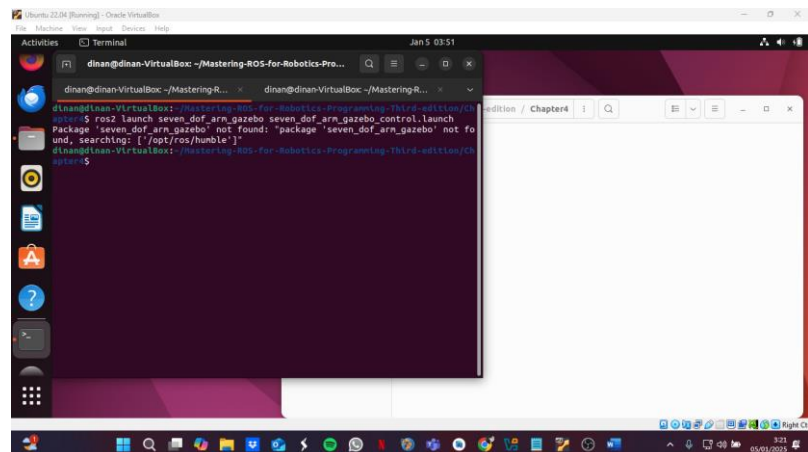


# Chapter 4 – Simulating Robot Using ROS & Gazebo

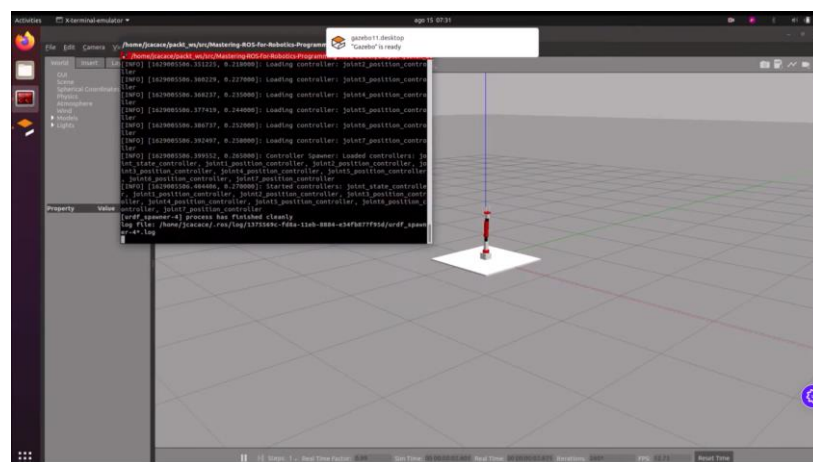
## Simulasi Pertama : Menggerakkan lengan robot

- pada terminal 1 jalankan perintah : `roslaunch seven_dof_arm_gazebo seven_dof_arm_gazebo_control.launch`

Output Saya (Belum Berhasil) :

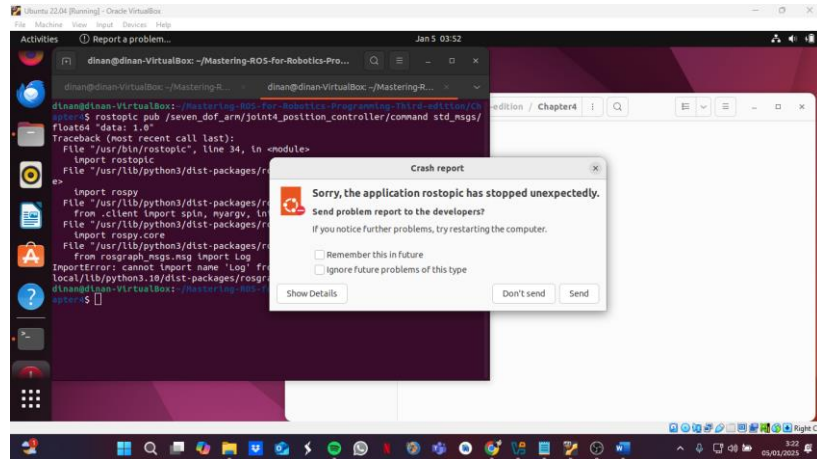


Output Seharusnya :

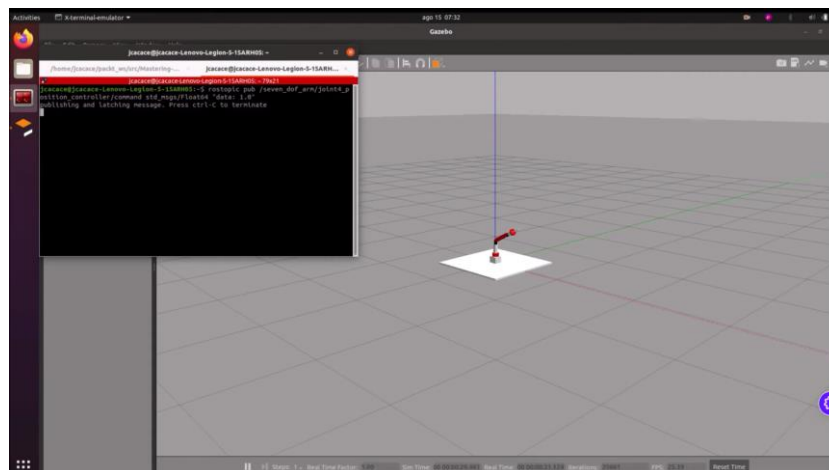


- pada terminal 2 jalankan perintah : `rostopic pub /seven_dof_arm/joint4_position_controller/command std_msgs/float64 "data: 1.0"`

Output Saya (Belum Berhasil) :



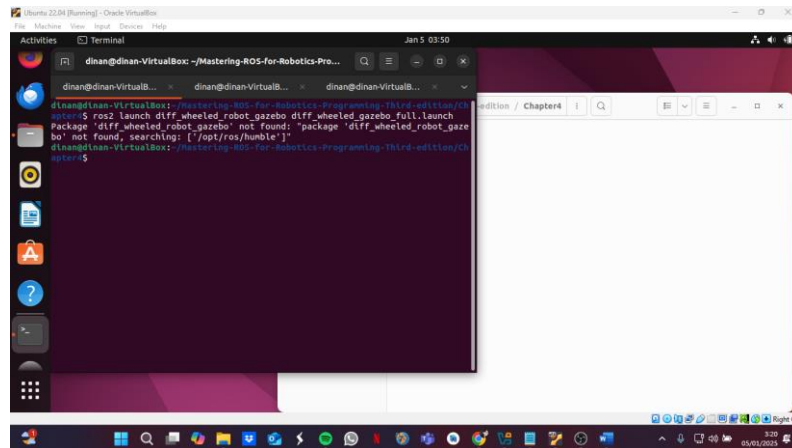
Output Seharusnya :



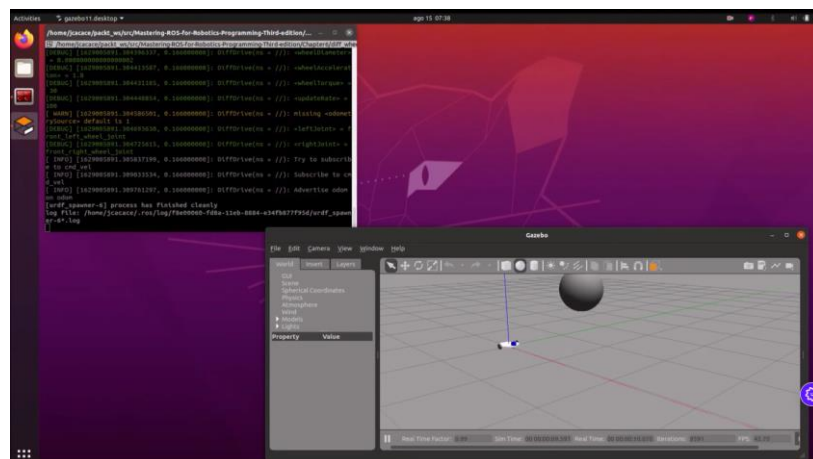
### Simulasi Kedua : Menggerakkan Posisi Robot

- pada terminal 1 jalankan perintah : `roslaunch diff_wheeled_robot_gazebo diff_wheeled_gazebo_full.launch`

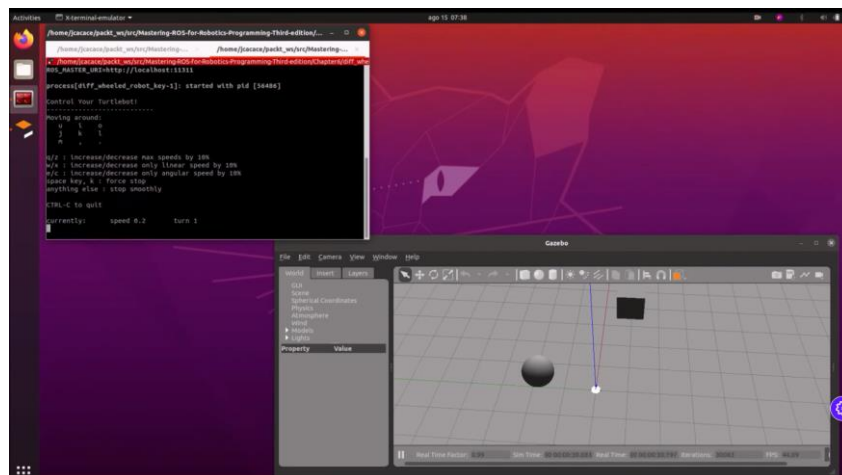
Output Saya (Belum Berhasil) :



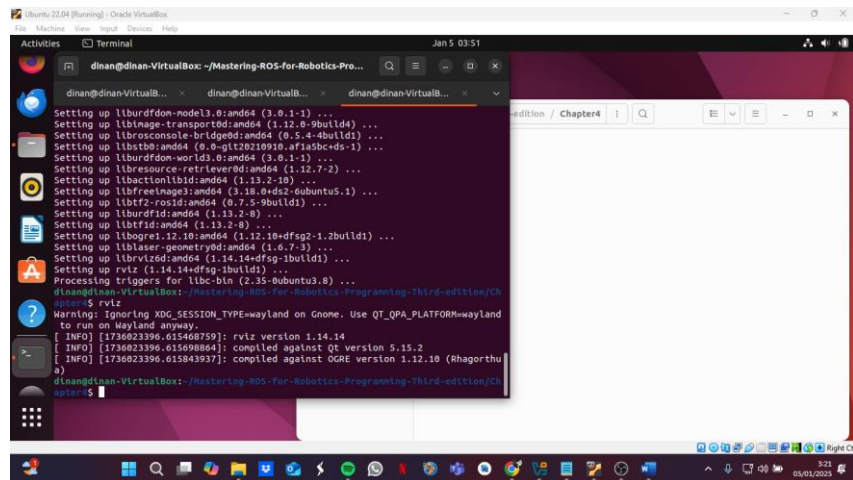
Output Seharusnya :



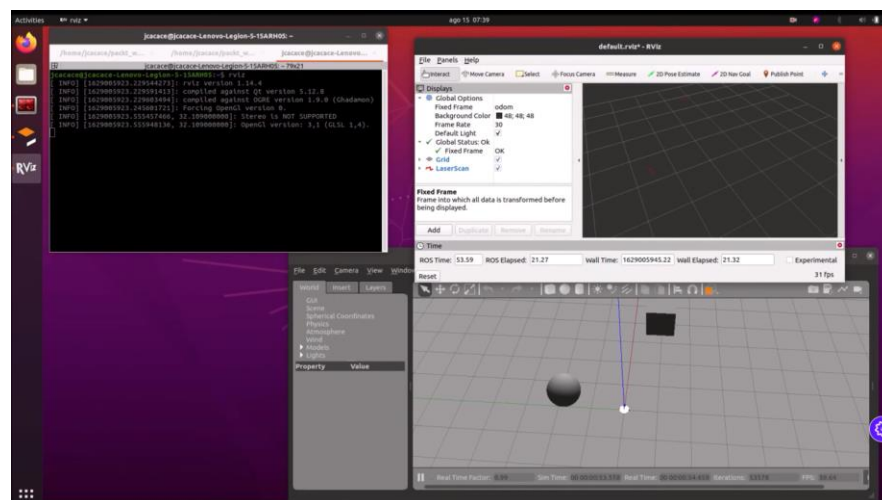
- Output Saya (Belum Berhasil) :



- Output Saya (Belum Berhasil) :



Output Seharusnya :



- pada simulasi pertama, terminal 1 berperan sebagai pembuka program dan terminal 2 berperan sebagai controller gerak robot
- pada simulasi kedua, terminal 1 berperan membuka program gazebo, kemudian terminal 2 berperan sebagai controller berbasis keyboard, dan terminal 3 membuka RViZ