

Langkah-langkah pada chapter ini memberikan pemahaman mendalam tentang konsep-konsep komunikasi antar node di ROS. Hal ini menjadi fondasi dalam pengembangan aplikasi robotika yang kompleks.

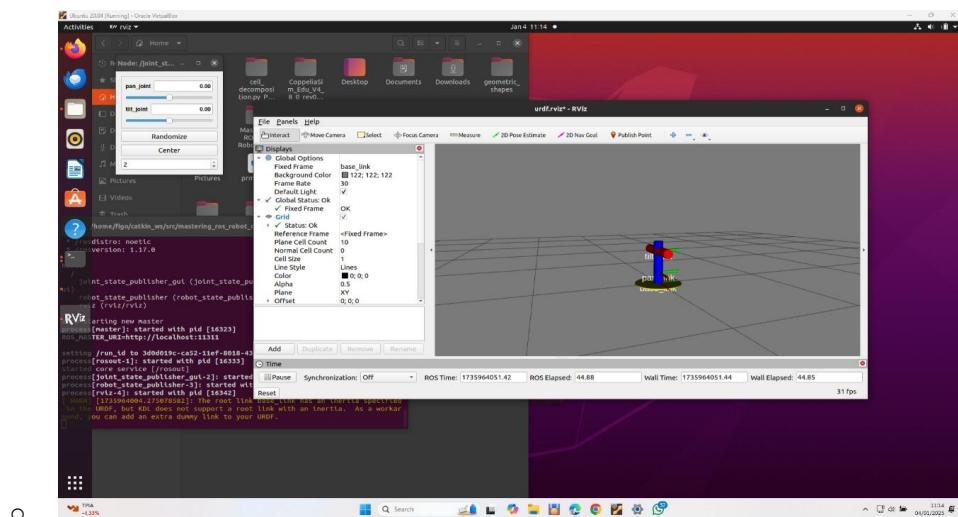
### Chapter 3: Representasi Robot

Chapter ini mendemonstrasikan cara memvisualisasikan model robot menggunakan deskripsi URDF di Rviz.

Langkah-langkah:

#### 1. Visualisasi Model Robot

- Lihat model dasar robot menggunakan `roslaunch mastering_ros_robot_description_pkg view_demo.launch`.
- Model lengan robot divisualisasikan dengan `roslaunch mastering_ros_robot_description_pkg view_arm.launch`.
- Model robot mobile divisualisasikan dengan `roslaunch mastering_ros_robot_description_pkg view_mobile_robot.launch`.



Representasi visual robot membantu memverifikasi desain dan struktur robot sebelum implementasi di dunia nyata. Pemahaman ini penting dalam desain robot modular.

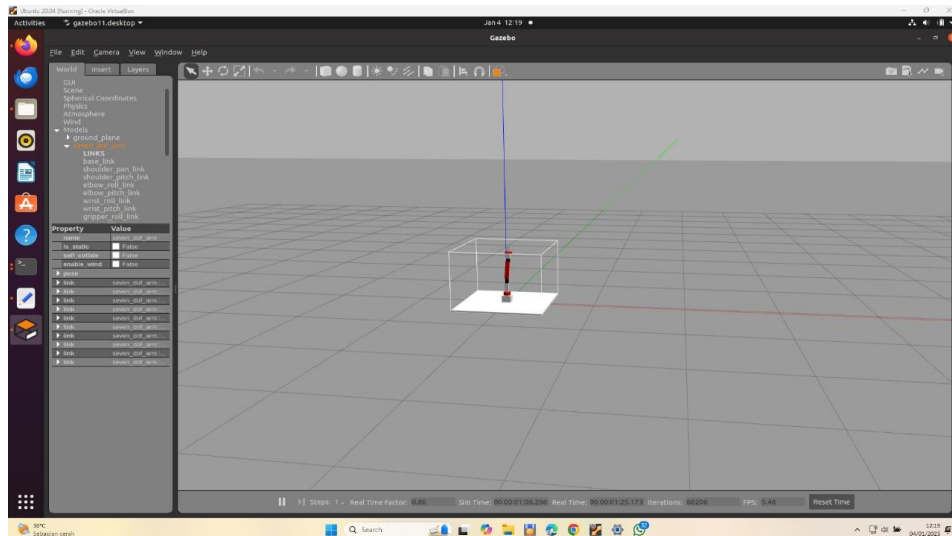
### Chapter 4: Simulasi Gazebo

Chapter ini membahas simulasi lengan robotik menggunakan Gazebo dan mengontrolnya melalui ROS.

Langkah-langkah:

1. Jalankan simulasi dengan `roslaunch seven_dof_arm_gazebo seven_dof_arm_world.launch`.
2. Kirim perintah untuk menggerakkan sendi robot dengan:

```
rostopic pub /seven_dof_arm/joint4_position_controller/command std_msgs/Float64 "data: 1.0"
```



Simulasi ini memungkinkan pengujian pergerakan robot secara virtual sehingga kesalahan dapat diidentifikasi sebelum implementasi nyata.

## Chapter 5: Integrasi CoppeliaSim dan Webots

Chapter ini mendemonstrasikan integrasi CoppeliaSim dan Webots dengan ROS untuk simulasi robotik yang lebih kompleks.

Langkah-langkah:

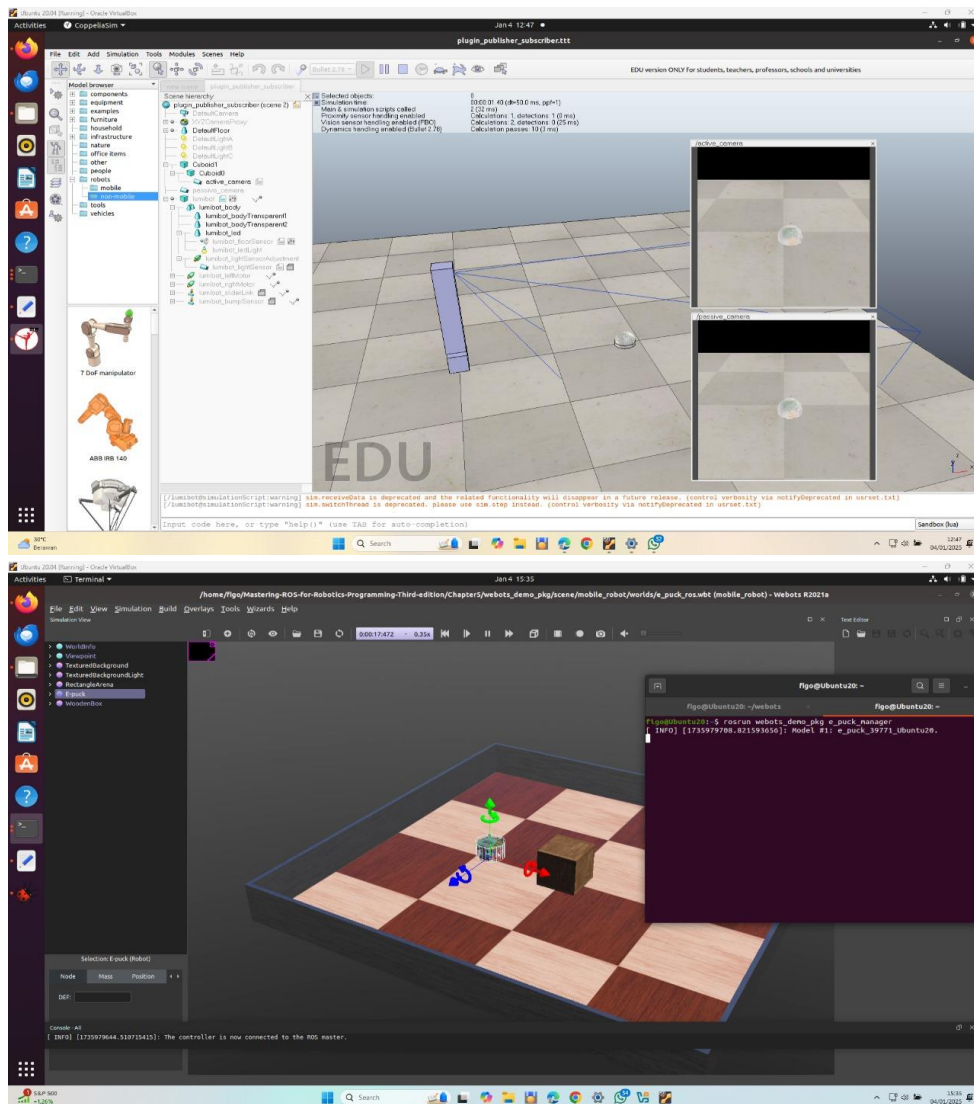
1. Publikasi data di CoppeliaSim dengan:

```
rostopic pub /csim_demo/seven_dof_arm/elbow_pitch/cmd std_msgs/Float32 "data: 1.0"
```

2. Lihat data yang diterima menggunakan:

```
rostopic echo /csim_demo/seven_dof_arm/elbow_pitch/state
```

3. Jalankan Webots dengan ./webots.



Integrasi ini memungkinkan penggunaan berbagai simulator robot untuk skenario yang lebih realistis.

## Chapter 6: Navigasi dan SLAM

Chapter ini berfokus pada navigasi robot mobile menggunakan SLAM (Simultaneous Localization and Mapping).

Langkah-langkah:

1. Jalankan simulasi di Gazebo dengan:

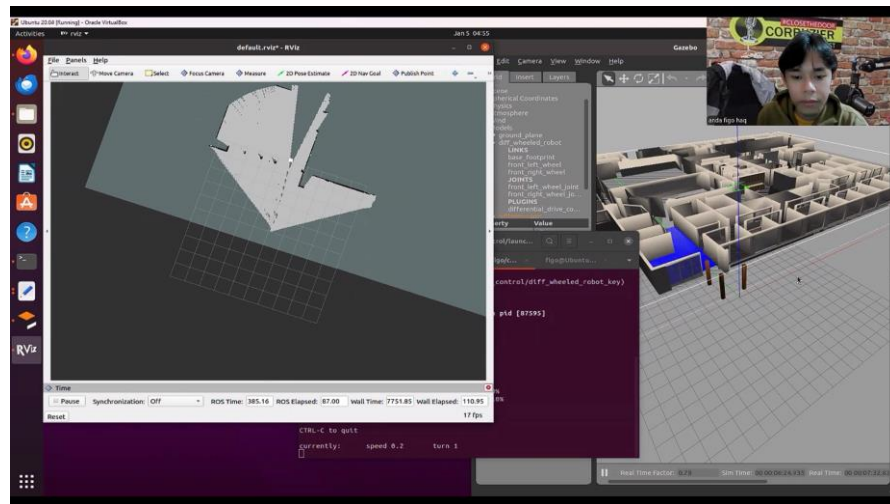
```
roslaunch diff_wheeled_robot_gazebo diff_wheeled_gazebo_full.launch
```

2. Implementasi SLAM dengan:

```
roslaunch diff_wheeled_robot_gazebo gmapping.launch
```

3. Simpan peta dengan `roslaunch map_server map_saver -f willow`.

4. Gunakan AMCL untuk lokalisasi dengan `roslaunch diff_wheeled_robot_gazebo amcl.launch`.



SLAM adalah komponen kunci dalam robotika mobile untuk membangun peta lingkungan secara otomatis sambil melokalisasi dirinya sendiri.

## Chapter 7: Manipulasi Objek

Chapter ini menjelaskan manipulasi objek menggunakan robot dengan tujuh derajat kebebasan.

Langkah-langkah:

1. Jalankan demo manipulasi dengan:

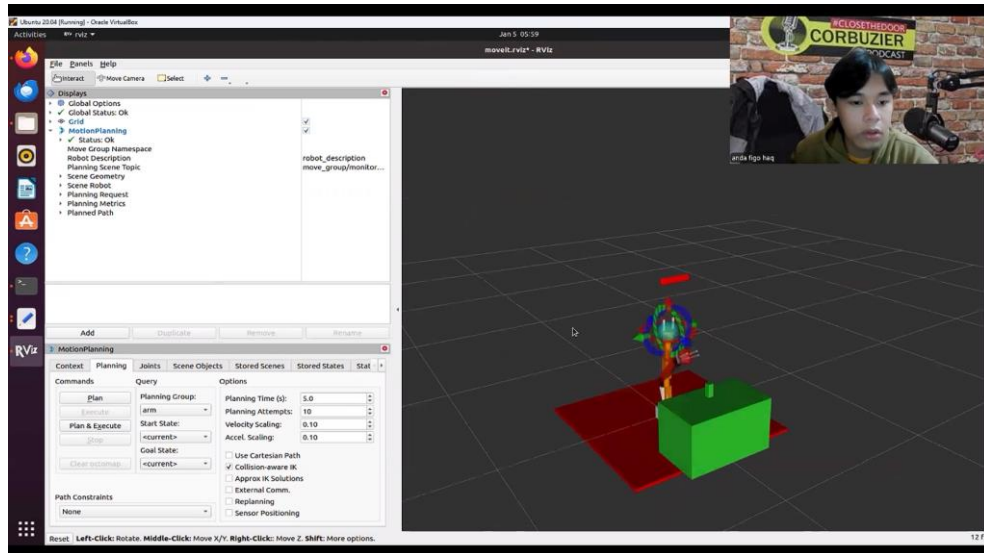
`roslaunch seven_dof_arm_config demo.launch`

2. Tambahkan objek kolisi menggunakan:

`roslaunch seven_dof_arm_test add_collision_object`

3. Lakukan pick and place dengan:

`roslaunch seven_dof_arm_test pick_place`



Manipulasi objek adalah langkah penting dalam robotika untuk mendukung aplikasi seperti perakitan otomatis dan penanganan material.

### Catatan Tambahan

Setiap chapter membangun pemahaman bertahap dalam pengembangan sistem robotika menggunakan ROS. Dokumentasi ini juga mendukung pembelajaran kolaboratif dengan integrasi berbagai simulator dan paket ROS.