

TECHNICAL REPORT ROBOTICS

Mastering ROS2 Book With Webot



Disusun oleh:

Ario Syawal Muhammad / 1103201243

PROGRAM STUDI TEKNIK KOMPUTER

FAKULTAS TEKNIK ELEKTRO

TELKOM UNIVERSITY

2024

ROBOT OPERATING SYSTEM

Robot Operating System (ROS) adalah kerangka kerja robotika open-source yang digunakan untuk mengembangkan aplikasi robotik. ROS menyediakan infrastruktur untuk memprogram, menjalankan, dan mendebug robot.

ROS terdiri dari sejumlah komponen, termasuk:

- **Node:** Node adalah unit dasar pemrograman ROS. Node dapat berkomunikasi satu sama lain menggunakan **topik**.

❖ Node ROS

- **Topik:** Topik adalah saluran komunikasi antara node. Topik dapat digunakan untuk mengirim berbagai jenis data, seperti data sensor, data aktuator, atau data kontrol.
- **Message:** Message adalah format data yang digunakan untuk mengirim data melalui topik.

❖ Message ROS

- **Service:** Service adalah cara untuk melakukan permintaan dan mendapatkan respons antara node.

❖ Service ROS

- **Parameter:** Parameter adalah nilai yang dapat dikonfigurasi oleh pengguna. Parameter dapat digunakan untuk menyesuaikan perilaku robot.

❖ Parameter ROS

ROS dapat digunakan untuk mengembangkan berbagai jenis aplikasi robotik, termasuk:

- **Robotika otonom:** ROS dapat digunakan untuk mengembangkan robot yang dapat beroperasi secara mandiri tanpa kendali manusia.
- **Robotika kolaboratif:** ROS dapat digunakan untuk mengembangkan robot yang dapat bekerja sama dengan manusia.
- **Robotika industri:** ROS dapat digunakan untuk mengembangkan robot untuk produksi industri.
- **Robotika penelitian:** ROS dapat digunakan untuk mengembangkan robot untuk penelitian ilmiah.

ROS adalah kerangka kerja robotika yang kuat dan fleksibel yang dapat digunakan untuk mengembangkan berbagai jenis aplikasi robotik. ROS telah diadopsi oleh perusahaan, pusat penelitian, dan universitas di seluruh dunia.

Berikut adalah beberapa manfaat menggunakan ROS:

- **Komunitas yang besar:** ROS memiliki komunitas pengguna yang besar dan aktif yang menyediakan dukungan dan sumber daya.

- **Dukungan lintas platform:** ROS mendukung berbagai platform, termasuk Linux, Windows, dan macOS.
- **Kemudahan penggunaan:** ROS relatif mudah digunakan, bahkan untuk pengembang pemula.
- **Kecepatan pengembangan:** ROS dapat mempercepat pengembangan aplikasi robotik.

Jika Anda tertarik untuk mempelajari lebih lanjut tentang ROS, ada banyak sumber daya yang tersedia online dan di perpustakaan.

MASTERING ROS BOOK

Mastering ROS for Robotics Programming adalah buku yang membahas ROS dan bagaimana menggunakannya untuk memprogram robot. Buku ini diterbitkan oleh Packt Publishing dan ditulis oleh Lentin Joseph. Buku ini ditujukan untuk pengguna ROS menengah, tetapi menyediakan juga pengenalan singkat terhadap konsep dasar. Buku ini menyediakan contoh kode dan latihan untuk memahami konsep dengan lebih baik. Kode sumber juga tersedia pada repositori GitHub untuk eksperimen langsung.

Buku ini dimulai dengan memperkenalkan konsep dasar ROS dan kemudian mengulas langkah-langkah memodelkan serta mensimulasikan robot. Pada bagian ini, pembahasannya mencakup penggunaan simulator seperti Gazebo, CoppeliaSim, dan Webots untuk mengontrol dan berinteraksi dengan robot yang telah dimodelkan. Para pembaca akan mempelajari cara terhubung ke robot menggunakan MoveIt! dan paket navigasi ROS melalui simulator tersebut. Selanjutnya, buku ini membahas plugin ROS, pengontrol, dan nodelet. Pada bagian terakhir, pembahasan melibatkan cara mengintegrasikan MATLAB dan Simulink dengan ROS.

Berikut adalah chapter yang akan dibahas dalam *Technical Report* ini:

- Chapter 1. Introduction to ROS
- Chapter 2. Getting Started with ROS Programming
- Chapter 3. Working with ROS for 3D Modeling
- Chapter 4. Simulating Robots Using ROS and Gazebo
- Chapter 5. Simulating Robots Using ROS, CoppeliaSim and Webots

CHAPTER 1: INTRODUCTION TO ROS

Dua bab pertama buku ini akan memperkenalkan konsep dasar ROS dan sistem manajemen paket ROS untuk mendekati pemrograman ROS. Pada bab pertama ini, kita akan membahas konsep ROS seperti master ROS, node ROS, server parameter ROS, serta pesan dan layanan ROS, sambil mendiskusikan apa yang kita perlukan untuk menginstal ROS dan cara memulai dengan master ROS .

Berikut adalah ROS Packages:

- **config:** Semua file konfigurasi yang digunakan dalam paket ROS ini disimpan di folder ini. Folder ini dibuat oleh pengguna dan merupakan praktik umum untuk memberi nama folder config karena ini adalah tempat kami menyimpan file konfigurasi.
- **include/package_name:** Folder ini terdiri dari header dan pustaka yang perlu kita gunakan di dalam paket.
- **script:** Folder ini berisi skrip Python yang dapat dieksekusi. Dalam diagram blok, kita dapat melihat dua contoh skrip.
- **src:** Folder ini menyimpan kode sumber C++.
- **launch:** Folder ini berisi file peluncuran yang digunakan untuk meluncurkan satu atau lebih node ROS.
- **msg:** Folder ini berisi definisi pesan khusus.
- **srv:** Folder ini berisi definisi layanan. **tindakan:** Folder ini berisi file tindakan. Kita akan mempelajari lebih lanjut tentang jenis file ini di bab berikutnya.
- **package.xml:** Ini adalah file manifes paket dari paket ini.
- **CMakeLists.txt:** File ini berisi arahan untuk mengkompilasi paket.
- **catkin_create_pkg:** Perintah ini digunakan untuk membuat paket baru.
- **rospack:** Perintah ini digunakan untuk mendapatkan informasi tentang paket yang ada di sistem file.
- **catkin_make:** Perintah ini digunakan untuk membangun paket-paket di ruang kerja.
- **rosdep:** Perintah ini akan menginstal dependensi sistem yang diperlukan untuk paket ini.
- **roscd:** Perintah ini digunakan untuk mengubah direktori saat ini menggunakan nama paket, nama tumpukan, atau lokasi khusus. Jika kita memberi argumen nama paket, argumen tersebut akan beralih ke folder paket tersebut.
- **roscp:** Perintah ini digunakan untuk menyalin file dari suatu paket.
- **roscd:** Perintah ini digunakan untuk mengedit file menggunakan editor vim.
- **rosvun:** Perintah ini digunakan untuk menjalankan executable di dalam sebuah paket.

Ros Noetic Install

ROS Noetic Ninjemys adalah versi ke-12 dari Robot Operating System (ROS). Diberi nama "Noetic" yang berasal dari "Noesis," yang berarti pemahaman atau wawasan. ROS Noetic dirilis pada bulan Mei 2020. Sebagai sebuah platform open-source, ROS Noetic menyediakan kerangka kerja untuk pengembangan perangkat lunak robotika.

ROS Noetic menyediakan perbaikan bug, peningkatan performa, dan fitur baru yang dapat membantu pengembang dalam membangun dan mengelola aplikasi robotika. Ini juga menunjukkan langkah-langkah menuju penyatuan lebih lanjut antara ROS 1 dan ROS 2, dua versi utama ROS.

```
ariosyawal@ariosyawal-VirtualBox: ~  
ariosyawal@ariosyawal-VirtualBox:~$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'  
[sudo] password for ariosyawal:  
ariosyawal@ariosyawal-VirtualBox:~$ sudo apt install curl # if you haven't already installed curl  
curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
curl is already the newest version (7.81.0-1ubuntu1.14).  
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.  
W: Target Packages (main/binary-amd64/Packages) is configured multiple times in /etc/apt/sources.list.d/ros2-latest.list:1 and /etc/apt/sources.list.d/ros2.list:1  
W: Target Packages (main/binary-all/Packages) is configured multiple times in /etc/apt/sources.list.d/ros2-latest.list:1 and /etc/apt/sources.list.d/ros2.list:1  
W: Target Translations (main/i18n/Translation-en_US) is configured multiple times in /etc/apt/sources.list.d/ros2-latest.list:1 and /etc/apt/sources.list.d/ros2.list:1  
W: Target Translations (main/i18n/Translation-en) is configured multiple times in /etc/apt/sources.list.d/ros2-latest.list:1 and /etc/apt/sources.list.d/ros2.list:1  
W: Target DEP-11 (main/dep11/Components-amd64.yml) is configured multiple times in /etc/apt/sources.list.d/ros2-latest.list:1 and /etc/apt/sources.list.d/ros2.list:1
```

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
```

Perintah ini digunakan untuk menambahkan repositori ROS ke daftar sumber paket apt. Repositori ini diperlukan untuk menginstal ROS di sistem Ubuntu.

```
sudo apt install curl
```

Perintah ini digunakan untuk menginstal paket curl. Paket curl diperlukan untuk mengunduh kunci publik ROS.

```
curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add
```

Perintah ini digunakan untuk menambahkan kunci publik ROS ke sistem. Kunci publik ini diperlukan untuk mengonfirmasi bahwa paket ROS telah diunduh dari sumber yang sah.

```
ariosyawal@ariosyawal-VirtualBox:~$ sudo apt install ros-noetic-desktop-full  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done
```

```
sudo apt install ros-noetic-desktop-full
```

program diatas mengunduh dan menginstal pengaturan desktop ROS Noetic yang komprehensif, termasuk alat penting, simulator, dan paket persepsi.

```
ariosyawal@ariosyawal-VirtualBox: ~$ sudo apt update
Hit:1 http://id.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://id.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://id.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Ign:5 http://packages.ros.org/ros/ubuntu jammy InRelease
Get:6 http://id.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [41.7 kB]
Get:7 http://packages.ros.org/ros2/ubuntu jammy InRelease [4,682 B]
Get:8 http://id.archive.ubuntu.com/ubuntu jammy-backports/main i386 Packages [33.9 kB]
Get:9 http://id.archive.ubuntu.com/ubuntu jammy-backports/universe i386 Packages [13.4 kB]
Get:10 http://id.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [24.3 kB]
Get:11 http://id.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.5 kB]
Err:12 http://packages.ros.org/ros/ubuntu jammy Release
```

sudo apt update

digunakan untuk mengunduh atau memperbarui daftar paket yang tersedia dari repositori. Perintah ini akan mengambil informasi terbaru tentang versi paket yang tersedia, serta dependensinya. Informasi ini kemudian dapat digunakan untuk menginstal atau memperbarui paket. Setelah tahap ini seharusnya ROS Noetic sudah terinstall dah sudah dapat digunakan.

CHAPTER 2: GETTING STARTED WITH ROS PROGRAMMING

Pada bagian ini kita akan mencoba untuk menginstall **Catkin**. Catkin adalah sistem pembangunan (build system) yang digunakan dalam Robot Operating System (ROS) untuk mengelola dan membangun kode sumber dari package ROS. Catkin dirancang khusus untuk mendukung ekosistem ROS dan menyediakan alat untuk mempermudah pengembangan, penerapan, dan distribusi perangkat lunak robotika.

Dengan menggunakan Catkin, pengembang ROS dapat dengan mudah mengelola proyek robotika mereka, mengelola dependensi, dan membangun perangkat lunak dengan efisien. Sistem ini membantu menjadikan proses pengembangan robotika di ROS lebih terstruktur dan mudah dielola.

```
Processing triggers for man-db (2.10.2-1) ..
ariosyawal@ariosyawal-VirtualBox: ~$ roscd
```

Perintah roscd adalah salah satu perintah di ROS (Robot Operating System) yang digunakan untuk berpindah (change directory) ke direktori sebuah package ROS. Perintah ini sangat berguna ketika bekerja dengan banyak package ROS dan ingin beralih ke direktori dari package tertentu tanpa harus mengetik jalur lengkapnya secara manual.

```

ariosyawal@ariosyawal-VirtualBox:~/ros_ws$ catkin_make
Command 'catkin_make' not found, but can be installed with:
sudo apt install catkin
ariosyawal@ariosyawal-VirtualBox:~/ros_ws$ sudo apt install catkin
Reading package lists... Done

```

Catkin_make

catkin_make adalah perintah dalam ROS yang digunakan untuk membangun (compile) package yang terdapat dalam workspace Catkin. Workspace Catkin adalah struktur direktori yang berisi package-package ROS yang sedang dikembangkan. Perintah ini biasanya dijalankan di dalam direktori workspace Catkin untuk menghasilkan executables, libraries, dan artifacts lainnya dari source code yang terdapat dalam package.

sudo apt install catkin

Perintah sudo apt install catkin adalah perintah yang digunakan untuk menginstal paket catkin menggunakan manajer paket APT (Advanced Package Tool) pada sistem operasi berbasis Debian, seperti Ubuntu.

```

ariosyawal@ariosyawal-VirtualBox:~/ros_ws$ cd ..
ariosyawal@ariosyawal-VirtualBox:~$ sudo chmod 777 ros_ws -R
ariosyawal@ariosyawal-VirtualBox:~$ cd ros_ws

```

sudo chmod 777 ros_ws -R

Perintah sudo chmod 777 ros_ws -R digunakan untuk mengubah izin (permissions) dari direktori ros_ws dan seluruh isianannya secara rekursif di dalamnya. Ini dapat memiliki dampak besar pada keamanan dan disarankan untuk digunakan dengan hati-hati.

```

ariosyawal@ariosyawal-VirtualBox:~/ros_ws$ cd src
ariosyawal@ariosyawal-VirtualBox:~/ros_ws/src$ catkin_create_pkg min_code roscpp
std_msgs
Created file min_code/package.xml
Created file min_code/CMakeLists.txt
Created folder min_code/include/min_code
Created folder min_code/src
Successfully created files in /home/ariosyawal/ros_ws/src/min_code. Please adjust
the values in package.xml

```

catkin_create_pkg min_code roscpp

Perintah catkin_create_pkg digunakan dalam lingkungan ROS untuk membuat struktur dasar dari sebuah package. Roscpp adalah daftar dependensi (ketergantungan) yang dimiliki oleh package. Dalam hal ini, package yang akan dibuat memiliki dependensi terhadap roscpp, yang merupakan library ROS untuk C++.

```

ariosyawal@ariosyawal-VirtualBox:~/ros_ws/src$ cd min_code
ariosyawal@ariosyawal-VirtualBox:~/ros_ws/src/min_code$ ls
CMakeLists.txt include package.xml src
ariosyawal@ariosyawal-VirtualBox:~/ros_ws/src/min_code$

```

min_code

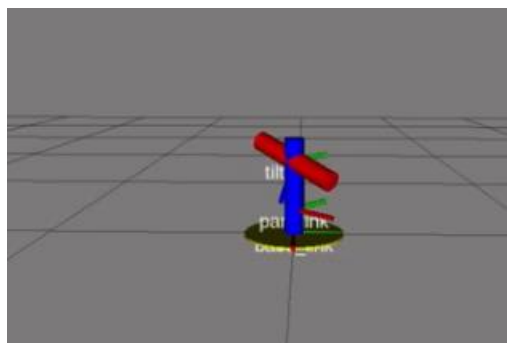
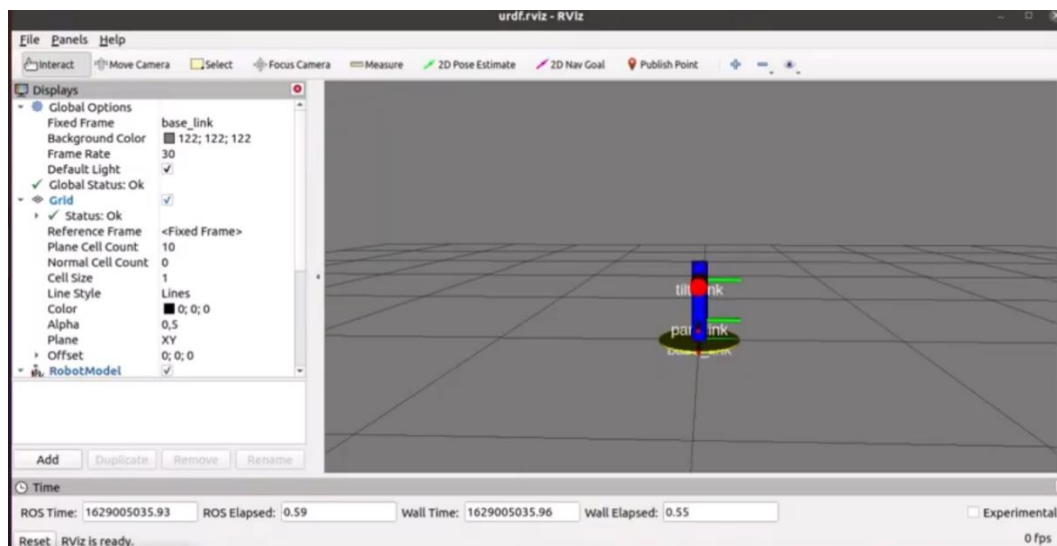
Ini adalah nama direktori yang akan menjadi direktori kerja baru. Dalam konteks perintah ini, direktori "min_code" akan menjadi direktori kerja. Perintah `cd min_code` digunakan untuk berpindah ke direktori yang disebut "min_code."

CHAPTER 3: WORKING WITH ROS FOR 3D MODELING

Tahap awal dalam pengembangan robot melibatkan proses perancangan dan pemodelan. Alat-alat CAD seperti Autodesk Fusion 360, SolidWorks, Blender, dan sebagainya dapat digunakan untuk mendesain dan memodelkan robot. Simulasi merupakan salah satu aspek kunci dari pemodelan robot, memungkinkan evaluasi dan pengujian virtual sebelum implementasi fisiknya.

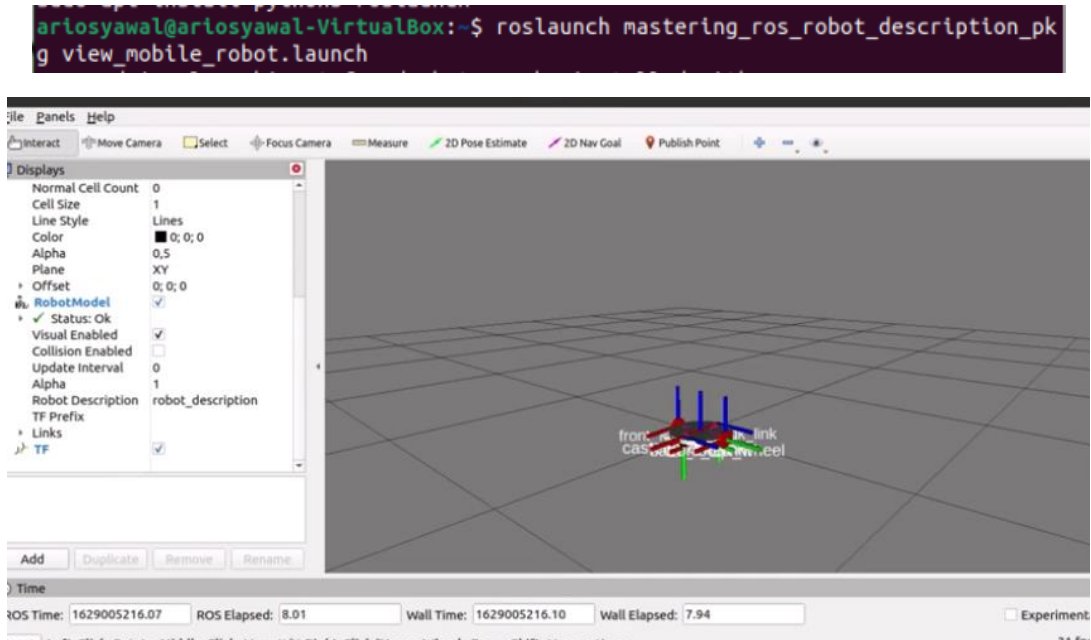
Dalam bagian ini, kita akan mengeksplorasi proses perancangan dua jenis robot. Salah satunya adalah manipulator dengan tujuh Derajat Kebebasan (DOF), sementara yang lainnya adalah robot dengan penggerak diferensial. Pada bagian berikutnya, fokus akan dialihkan ke simulasi, dengan pembahasan tentang pembuatan perangkat keras fisik dan interaksi dengan ROS.

```
artiosyawal@artiosyawal-VirtualBox:~$ roslaunch mastering_ros_robot_description_pkg view_demo.launch
```



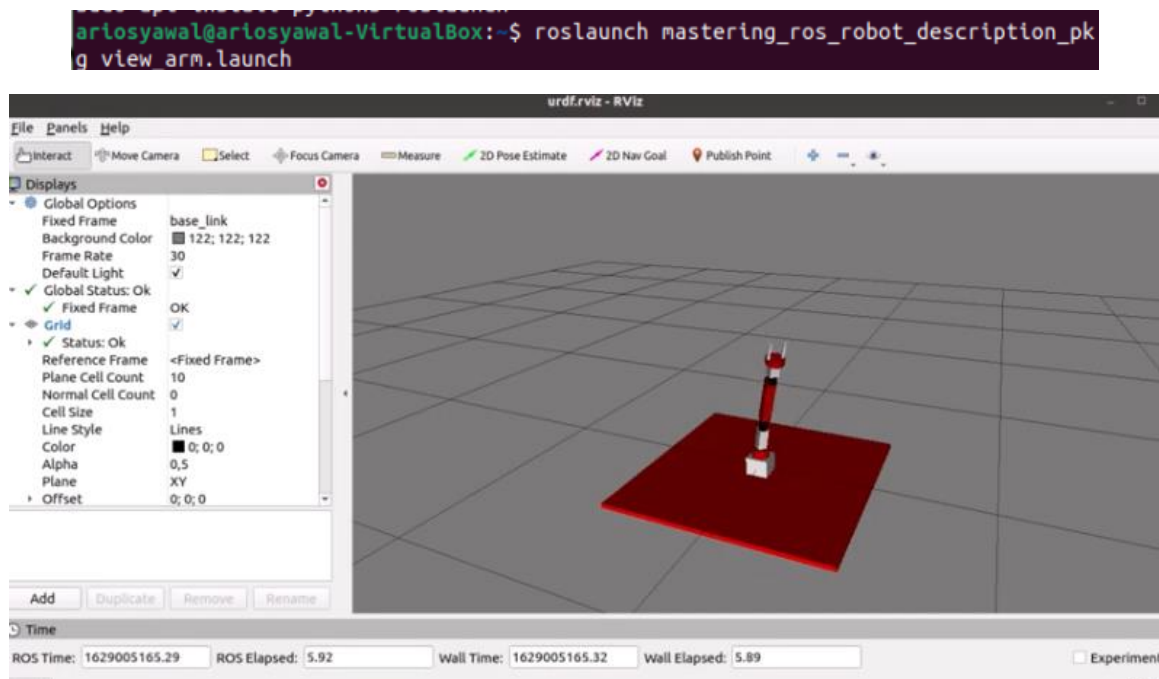
```
roslaunch mastering_ros_robot_description_pkg view_demo.launch
```


Perintah `roslaunch mastering_ros_robot_description_pkg view_demo.launch` adalah perintah di lingkungan ROS (Robot Operating System) yang digunakan untuk meluncurkan (launch) suatu file konfigurasi launch (.launch). Dalam hal ini, file konfigurasi yang diluncurkan adalah `view_demo.launch`.



roslaunch mastering_ros_robot_description_pkg view_mobile_robot.launch

Perintah `roslaunch mastering_ros_robot_description_pkg view_mobile_robot.launch` adalah perintah di lingkungan ROS (Robot Operating System) yang digunakan untuk meluncurkan (launch) file konfigurasi launch dengan nama `view_mobile_robot.launch`.



roslaunch mastering_ros_robot_description_pkg view_arm.launch

Perintah `roslaunch mastering_ros_robot_description_pkg view_arm.launch` adalah perintah di lingkungan ROS (Robot Operating System) yang digunakan untuk meluncurkan (launch) file konfigurasi launch dengan nama `view_arm.launch`.

CHAPTER 4: SIMULATING ROBOTS USING ROS AND GAZEBO

Gazebo adalah simulator multi-robot yang dirancang untuk mensimulasikan robot dalam dan di luar ruangan dengan lingkungan yang kompleks. Dengan Gazebo, kita dapat melakukan simulasi robot yang rumit, sensor robot, serta berbagai objek dalam bentuk objek 3D. Keunggulan Gazebo terletak pada integrasinya yang mulus dengan ROS melalui antarmuka ROS yang tepat, memungkinkan kontrol penuh atas Gazebo di dalam lingkungan ROS.

```
ariosyawal@ariosyawal-VirtualBox:~$ roslaunch seven_dof_arm_gazebo seven_dof_arm_gazebo_control.launch
```

roslaunch seven_dof_arm_gazebo seven_dof_arm_gazebo_control.launch

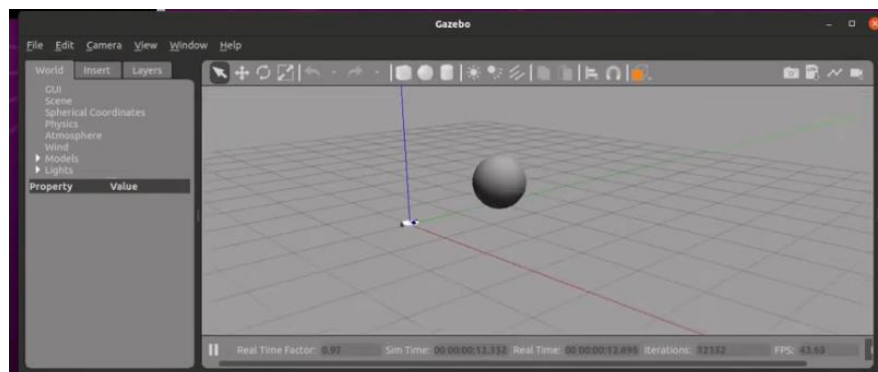
ika perintah di atas dijalankan, ROS akan mencari file `seven_dof_arm_gazebo_control.launch` di dalam package `seven_dof_arm_gazebo` dan meluncurkan simulasi Gazebo sesuai dengan konfigurasi yang didefinisikan di dalamnya. Ini dapat mencakup peluncuran model simulasi robot, lingkungan simulasi, serta node-node yang mengontrol perilakunya.

```
ariosyawal@ariosyawal-VirtualBox:~$ rostopic pub /seven_dof_arm/joint4_position_controller/command std_msgs/Float64 "data: 1.0"
```

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

perintah ini secara efektif mengirimkan pesan dengan nilai 1.0 ke topik yang terkait dengan kontrol posisi joint keempat dari robot dengan tujuh derajat kebebasan. Tujuan dari penggunaan perintah seperti ini dapat beragam, seperti memberikan perintah gerakan atau posisi tertentu pada suatu joint dalam simulasi atau sistem robot yang sedang berjalan.

```
ariosyawal@ariosyawal-VirtualBox:~$ roslaunch diff_wheeled_robot_gazebo diff_wheeled_gazebo_full.launch
```



roslaunch diff_wheeled_robot_gazebo diff_wheeled_gazebo_full.launch

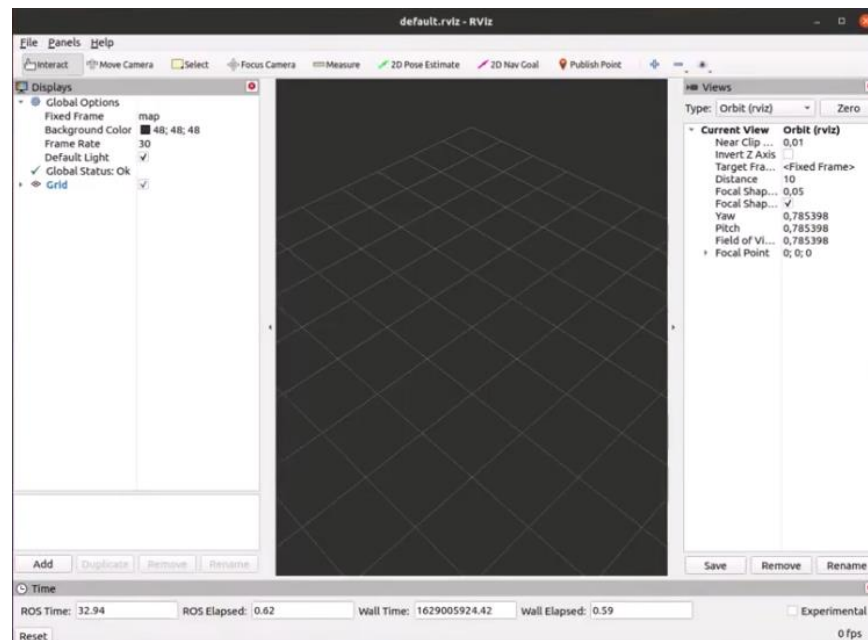
Perintah `roslaunch diff_wheeled_robot_gazebo diff_wheeled_gazebo_full.launch` digunakan untuk meluncurkan (launch) simulasi robot penggerak diferensial pada Gazebo. Jika perintah di atas dijalankan, ROS akan mencari file `diff_wheeled_gazebo_full.launch` di dalam package `diff_wheeled_robot_gazebo` dan meluncurkan simulasi Gazebo sesuai dengan konfigurasi yang didefinisikan di dalamnya. Ini dapat mencakup peluncuran model simulasi robot penggerak diferensial, lingkungan simulasi, serta node-node yang mengontrol perilakunya.

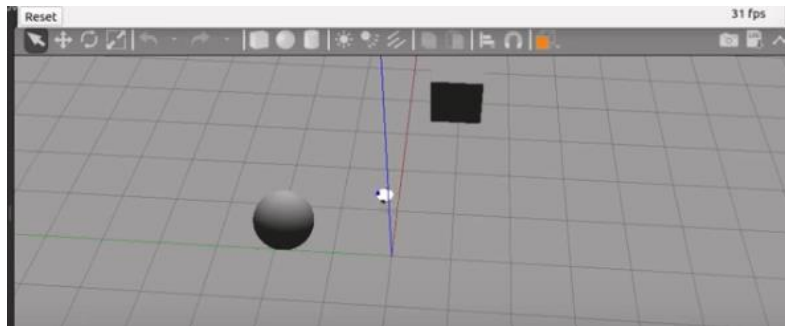
```
ariosyawal@ariosyawal-VirtualBox:~$ roslaunch diff_wheeled_robot_control keyboard_teleop.launch
```

roslaunch diff_wheeled_robot_control keyboard_teleop.launch

Perintah `roslaunch diff_wheeled_robot_control keyboard_teleop.launch` adalah perintah di lingkungan ROS (Robot Operating System) yang digunakan untuk meluncurkan (launch) node atau sekumpulan node yang mengaktifkan kontrol teleoperasi keyboard untuk robot penggerak diferensial.

```
ariosyawal@ariosyawal-VirtualBox:~$ rviz
```





Perintah rviz digunakan untuk meluncurkan aplikasi visualisasi ROS yang disebut RViz (ROS Visualization). RViz adalah alat yang memungkinkan pengguna untuk memvisualisasikan data dari berbagai sensor, model robot, dan informasi lainnya yang dihasilkan oleh sistem ROS. Setelah RViz dijalankan, pengguna dapat mengonfigurasi tampilan visualisasi, menambahkan model robot, menentukan frame koordinat, dan melakukan berbagai tindakan lainnya untuk memantau dan menganalisis data robotika secara interaktif.

CHAPTER 5: SIMULATING ROBOTS USING ROS, COPPELIASIM AND WEBOTS

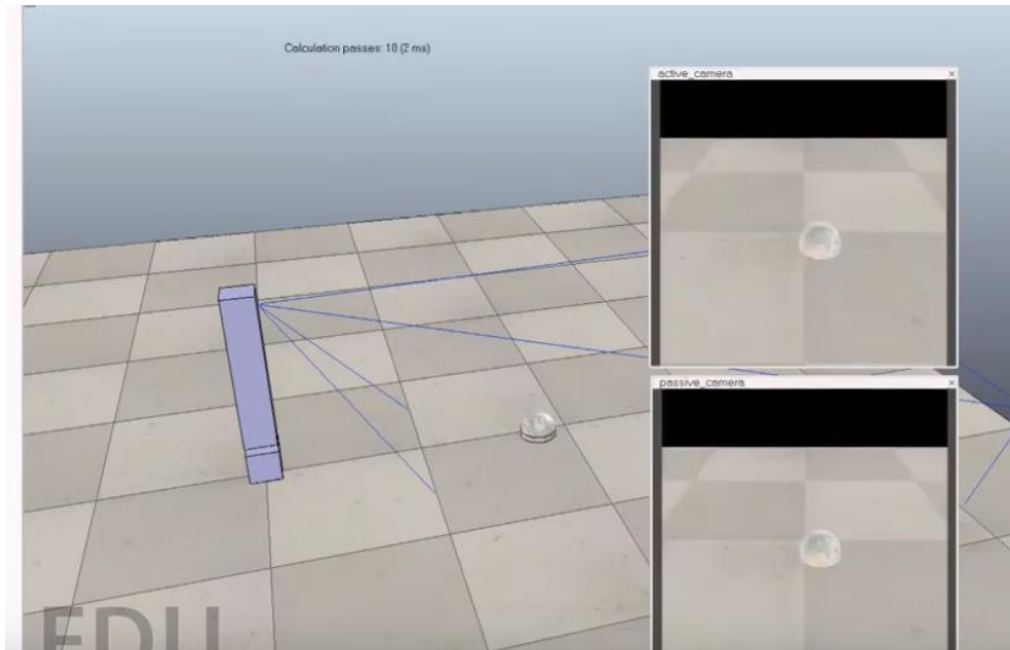
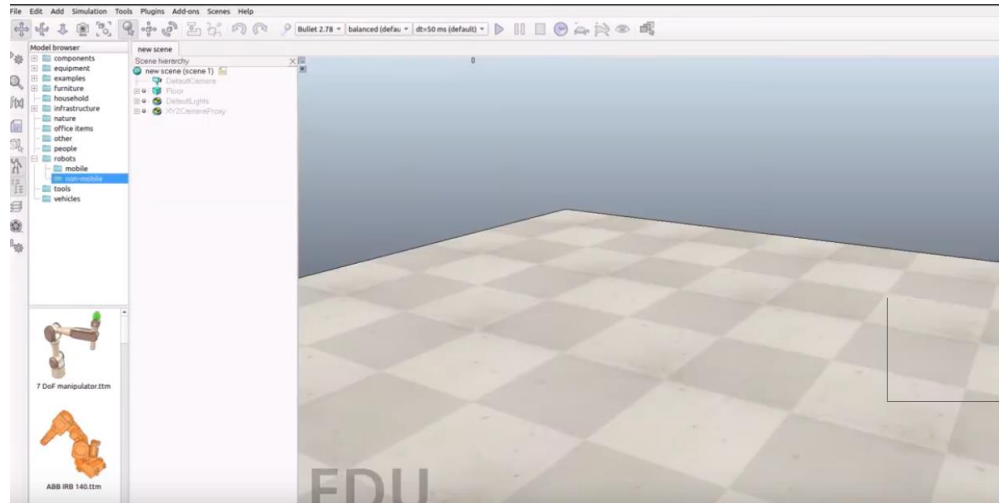
Sebelum memulai penggunaan CoppeliaSim, langkah pertama adalah melakukan instalasi pada sistem dan mengonfigurasi lingkungan agar dapat membentuk jembatan komunikasi antara ROS dan simulasi adegan. CoppeliaSim adalah perangkat lunak lintas platform yang dapat diakses pada berbagai sistem operasi, termasuk Windows, macOS, dan Linux. Pengembangan dilakukan oleh Coppelia Robotics GmbH, dan perangkat lunak ini tersedia dengan lisensi pendidikan dan versi komersial yang dapat diakses secara gratis.

```
ariosyawal@ariosyawal-VirtualBox:~$ cd dev/coppelasim/
```

cd dev/coppelasim

Perintah `cd dev/coppelasim` digunakan untuk mengubah direktori kerja saat ini ke direktori "coppelasim" yang terletak di dalam direktori "dev". Namun, perlu dicatat bahwa perintah ini mungkin tidak benar jika Anda tidak berada di direktori yang sesuai.

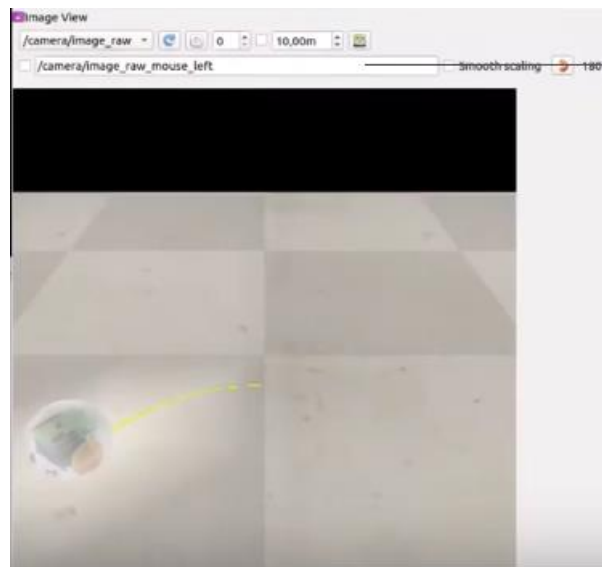
```
ariosyawal@ariosyawal-VirtualBox:~$ ./coppelasim.sh
```



/coppelasim.sh

Perintah `/coppelasim.sh` adalah perintah untuk menjalankan skrip shell (sh) bernama "coppelasim.sh". Namun, perlu dicatat bahwa jalannya skrip ini tergantung pada lokasi dan hak akses file tersebut.

```
ariosityawal@ariosityawal-VirtualBox:~$ rqt_image_view
```

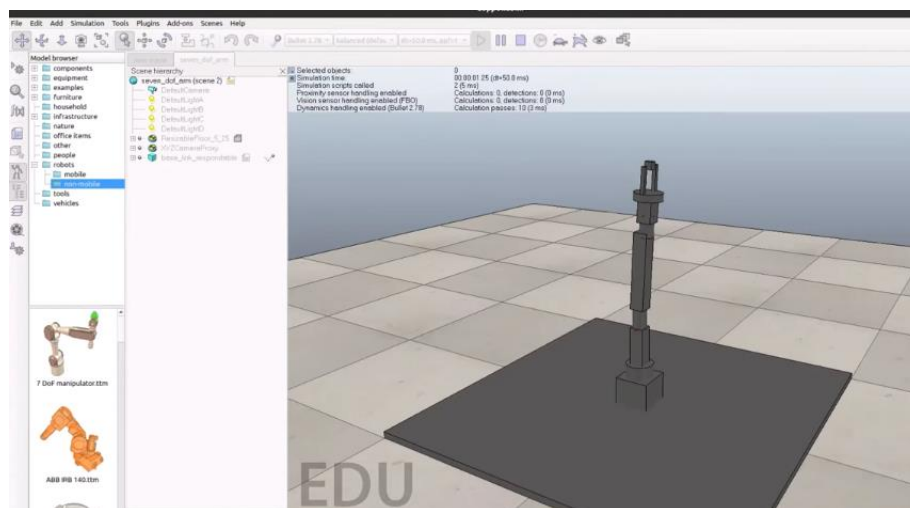


rqt_image_view

Perintah `rqt_image_view` digunakan untuk meluncurkan alat visualisasi gambar dalam lingkungan ROS, yang disebut "`rqt_image_view`". Alat ini memungkinkan pengguna untuk memonitor dan memvisualisasikan data citra atau gambar yang dihasilkan oleh kamera atau sensor visual dalam sistem ROS.

Alat `rqt_image_view` sangat bermanfaat dalam pengembangan dan debugging aplikasi robotika yang menggunakan sensor penglihatan. Ini memungkinkan pengguna untuk secara langsung melihat gambar yang dihasilkan oleh kamera atau sensor visual, membantu dalam pemahaman dan evaluasi performa sistem.

```
ariosityawal@ariosityawal-VirtualBox:~$ roscore
```



roscore

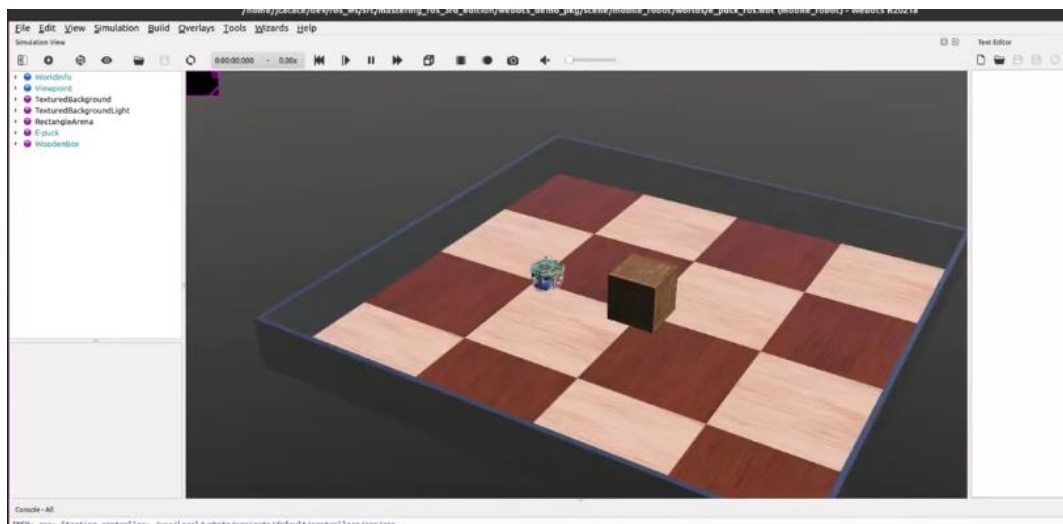
Perintah roscore digunakan untuk memulai inti dari sistem ROS, yang disebut "ROS Master." ROS Master adalah bagian dari infrastruktur komunikasi ROS dan berfungsi sebagai layanan pusat untuk mengoordinasikan komunikasi antara node-node di dalam sistem ROS.

```
ariosityawal@ariosityawal-VirtualBox:~$ rostopic pub /csim_demo/seven_dof_arm/elbow_pitch/cmd std_msgs/Float32 "data: 1.0"
```

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

Perintah ini digunakan untuk mengirimkan perintah atau perubahan pada sendi siku dari manipulator dengan tujuh derajat kebebasan ke dalam sistem ROS. Detail spesifik dari bagaimana pesan ini akan diinterpretasi dan dijalankan oleh sistem tergantung pada konfigurasi dan logika kontrol yang terkait dengan manipulator tersebut.

```
ariosityawal@ariosityawal-VirtualBox:~$ rostopic echo /csim_demo/seven_dof_arm/elbow_pitch/state
```



rostopic echo /csim_demo/seven_dof_arm/elbow_pitch/state

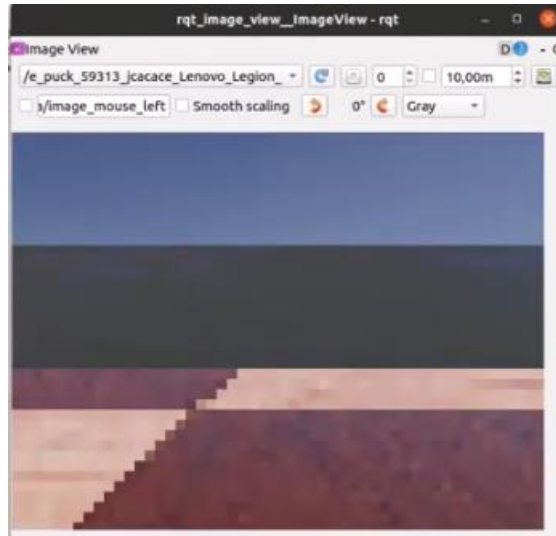
Dengan menjalankan perintah ini, terminal akan menampilkan data yang dikirimkan ke topik tersebut oleh node atau sistem lainnya. Data ini mungkin mencakup informasi tentang posisi, kecepatan, atau atribut lainnya yang terkait dengan sendi siku dari manipulator tersebut.

```
ariosityawal@ariosityawal-VirtualBox:~$ rostopic list
```

rostopic list

Perintah rostopic list digunakan untuk menampilkan daftar topik yang sedang aktif di sistem ROS. Ketika Anda menjalankan perintah ini, terminal akan menampilkan daftar nama-nama topik yang dapat digunakan atau diakses oleh node-node di dalam lingkungan ROS saat itu

```
ariosityawal@ariosityawal-VirtualBox:~$ rostopic echo /model_name
```



rostopic echo /model_name

Dengan menjalankan perintah ini, terminal akan menampilkan data yang dikirimkan ke topik tersebut oleh node atau sistem lainnya. Data ini mungkin mencakup informasi mengenai nama suatu model yang sedang dioperasikan dalam konteks sistem ROS yang sedang berjalan.