

Nama : Fariz Rahman Ramadhan

NIM : 1103204046

UTS ROBOTIKA TECHINAL REPORT

GETTING STARTED WITH TURTLESIM | SPAWN TURTLE2 IN TURTLESIM

I. PENJELASAN UMUM

Robot Operating System 2 (ROS2) adalah sebuah kerangka pengembangan perangkat lunak yang dirancang khusus untuk memudahkan pembuatan dan pengoperasian perangkat lunak untuk robot. ROS2 adalah penerus dari ROS yang awalnya dikembangkan oleh Willow Garage. Dengan fokus pada kehandalan, keamanan, dan skalabilitas, ROS2 memberikan platform yang kuat untuk pengembangan robotika modern.

Salah satu cara untuk memahami dasar-dasar ROS2 adalah melalui tutorial yang spesifik, seperti "Getting Started with turtlesim | Spawn turtle2 in turtlesim." Dalam tutorial ini, pengguna diperkenalkan pada simulasi yang disebut turtlesim, lingkungan simulasi yang sederhana namun bermanfaat untuk memahami konsep-konsep dasar ROS2.

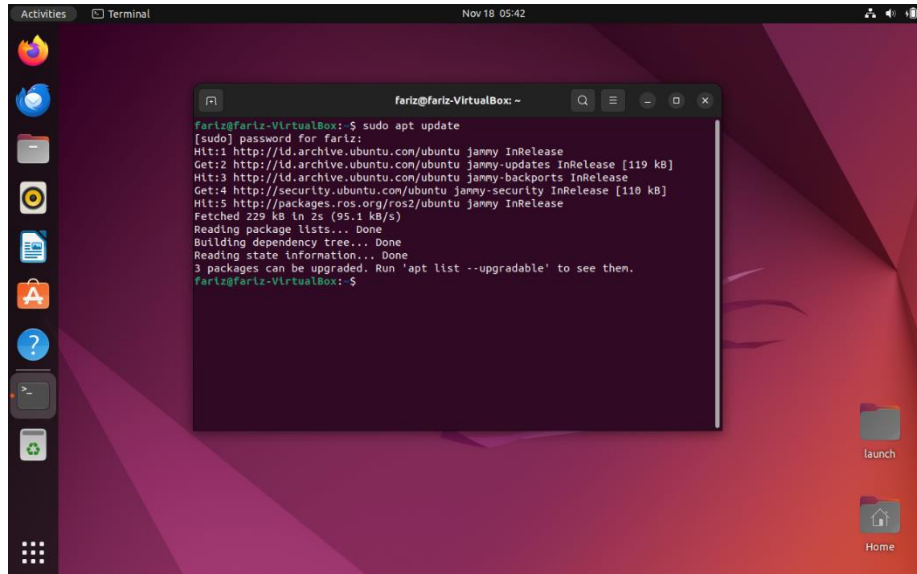
Turtlesim adalah simulator yang memungkinkan pengguna untuk mengontrol seekor "turtle" yang bergerak di dalam lingkungan 2D. Turtle tersebut dapat diperintahkan untuk bergerak, berputar, dan melakukan berbagai tugas lainnya. Simulasi ini menciptakan lingkungan virtual yang ideal untuk memahami konsep-konsep ROS2, seperti node, topic, dan service.

Dalam konteks tutorial ini, langkah-langkahnya mungkin melibatkan pembuatan dan manajemen node ROS2. Node merupakan entitas dasar dalam sistem ROS yang berkomunikasi satu sama lain melalui mekanisme publikasi dan langganan topik (topics). Topik digunakan untuk pertukaran pesan antar node, dan tutorial ini mungkin memandu pengguna untuk menciptakan turtle kedua dengan menggunakan perintah atau skrip khusus, yang dikenal sebagai "spawn."

Melalui implementasi praktis ini, pengguna diharapkan dapat memahami cara mengorganisasi dan mengelola entitas dalam ROS2, serta bagaimana komunikasi antar entitas dilakukan melalui mekanisme ROS2. Selain itu, tutorial ini mungkin membuka pintu bagi konsep-konsep pengembangan lanjutan, memberikan landasan untuk pemahaman yang lebih mendalam tentang robotika dan pengembangan perangkat lunak yang terkait.

II. Pengerjaan

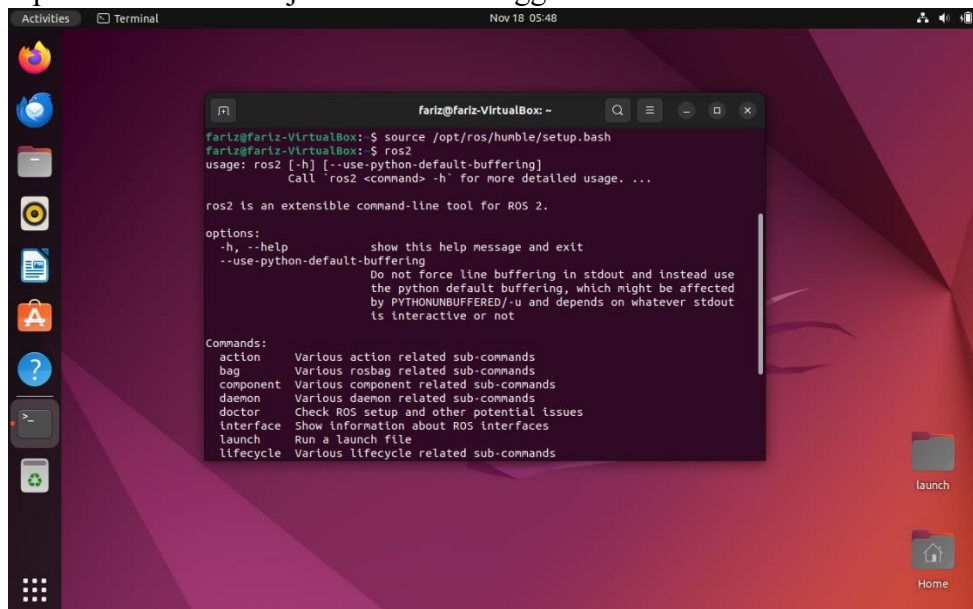
- Langkah pertama yang melibatkan perintah `sudo apt update` pada sistem berbasis Ubuntu atau Debian bertujuan untuk mengupdate informasi paket yang tersedia dari repositori paket yang terpasang di sistem.



```
fariz@fariz-VirtualBox: ~  
fariz@fariz-VirtualBox:~$ sudo apt update  
[sudo] password for fariz:  
Hit:1 http://id.archive.ubuntu.com/ubuntu jammy InRelease  
Get:2 http://id.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]  
Hit:3 http://id.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]  
Hit:5 http://packages.ros.org/ros2/ubuntu jammy InRelease  
Fetched 229 kB in 2s (95.1 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
3 packages can be upgraded. Run 'apt list --upgradable' to see them.  
fariz@fariz-VirtualBox:~$
```

Jadi, ketika menjalankan `sudo apt update`, akan memberitahu sistem untuk memperbarui daftar paket yang tersedia, sehingga dapat memastikan bahwa informasi terkini mengenai paket-paket yang tersedia di repositori dapat diakses dan digunakan. Setelah menjalankan perintah ini,

- Langkah kedua, yaitu menjalankan perintah `source /opt/ros/humble/setup.bash`, bertujuan untuk memasukkan variabel lingkungan (environment variables) yang diperlukan untuk menjalankan dan menggunakan ROS 2 di terminal.



```
fariz@fariz-VirtualBox: ~  
fariz@fariz-VirtualBox:~$ source /opt/ros/humble/setup.bash  
fariz@fariz-VirtualBox:~$ ros2  
usage: ros2 [-h] [--use-python-default-buffering]  
        Call 'ros2 <command> -h' for more detailed usage. ...  
  
ros2 is an extensible command-line tool for ROS 2.  
  
options:  
  -h, --help            show this help message and exit  
  --use-python-default-buffering  
                        Do not force line buffering in stdout and instead use  
                        the python default buffering, which might be affected  
                        by PYTHONUNBUFFERED/-u and depends on whatever stdout  
                        is interactive or not  
  
Commands:  
  action                Various action related sub-commands  
  bag                   Various rosbag related sub-commands  
  component             Various component related sub-commands  
  daemon               Various daemon related sub-commands  
  doctor                Check ROS setup and other potential issues  
  interface             Show information about ROS interfaces  
  launch               Run a launch file  
  lifecycle             Various lifecycle related sub-commands
```

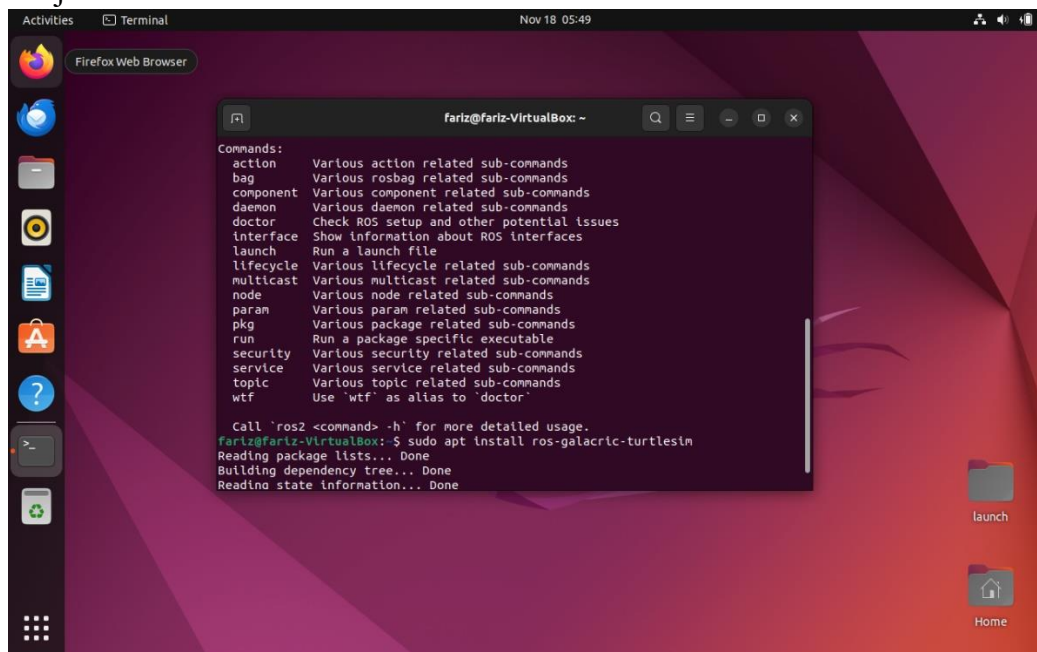
Setelah menjalankan perintah ini, variabel lingkungan yang dibutuhkan oleh ROS 2 akan diatur dengan benar untuk sesi terminal saat itu. Ini mencakup penambahan jalur ke eksekutabel ROS

2, library, dan lainnya ke variabel PATH sehingga sistem dapat menemukan dan menjalankan perintah-perintah ROS 2 dengan benar.

Setelah itu, disarankan untuk menjalankan perintah ROS 2 tertentu, yang mungkin seperti `ros2 run`, `ros2 launch`, atau perintah ROS 2 lainnya. Penggunaan perintah ini bergantung pada tugas atau skenario pengembangan tertentu yang ingin dijalankan.

Dengan menjalankan skrip `setup.bash` dan kemudian menjalankan perintah ROS 2, memastikan bahwa lingkungan diatur dengan benar untuk menggunakan ROS 2 dan bahwa sistem dapat mengakses dan menjalankan perangkat lunak ROS 2 dengan baik. Ini merupakan langkah penting sebelum dapat mulai mengembangkan atau menjalankan aplikasi yang berbasis ROS 2.

- Langkah ketiga, yaitu menjalankan perintah `sudo apt install ros-galactic-turtlesim`, bertujuan untuk menginstal paket turtlesim yang terkait dengan ROS Galactic. Penjelasan lebih rinci:



The screenshot shows a terminal window titled 'fariz@fariz-VirtualBox: ~' with a list of ROS 2 commands and their descriptions. Below the list, the command `fariz@fariz-VirtualBox: ~$ sudo apt install ros-galactic-turtlesim` is entered, followed by the output: `Reading package lists... Done`, `Building dependency tree... Done`, and `Reading state information... Done`.

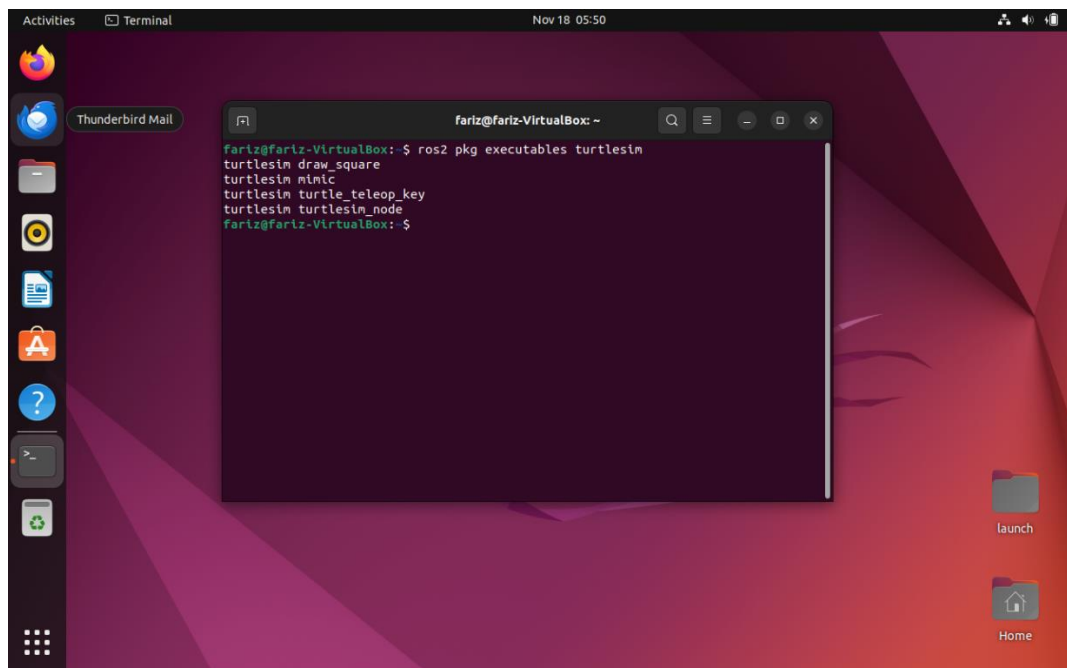
```
Commands:
action      Various action related sub-commands
bag         Various rosbag related sub-commands
component   Various component related sub-commands
daemon      Various daemon related sub-commands
doctor      Check ROS setup and other potential issues
interface   Show information about ROS interfaces
launch      Run a launch file
lifecycle   Various lifecycle related sub-commands
multicast   Various multicast related sub-commands
node        Various node related sub-commands
param       Various param related sub-commands
pkg         Various package related sub-commands
run         Run a package specific executable
security     Various security related sub-commands
service     Various service related sub-commands
topic       Various topic related sub-commands
wtf         Use 'wtf' as alias to 'doctor'

Call 'ros2 <command> -h' for more detailed usage.
fariz@fariz-VirtualBox: ~$ sudo apt install ros-galactic-turtlesim
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

`ros-galactic-turtlesim`: Ini adalah nama paket ROS yang ingin diinstal. Dalam hal ini, paket tersebut disebut turtlesim dan ditargetkan untuk versi ROS Galactic. turtlesim adalah simulator sederhana yang memungkinkan untuk mengontrol turtle di lingkungan simulasi 2D. Versi ROS disebutkan sebagai "Galactic," yang menunjukkan bahwa paket ini sesuai dengan distribusi ROS Galactic.

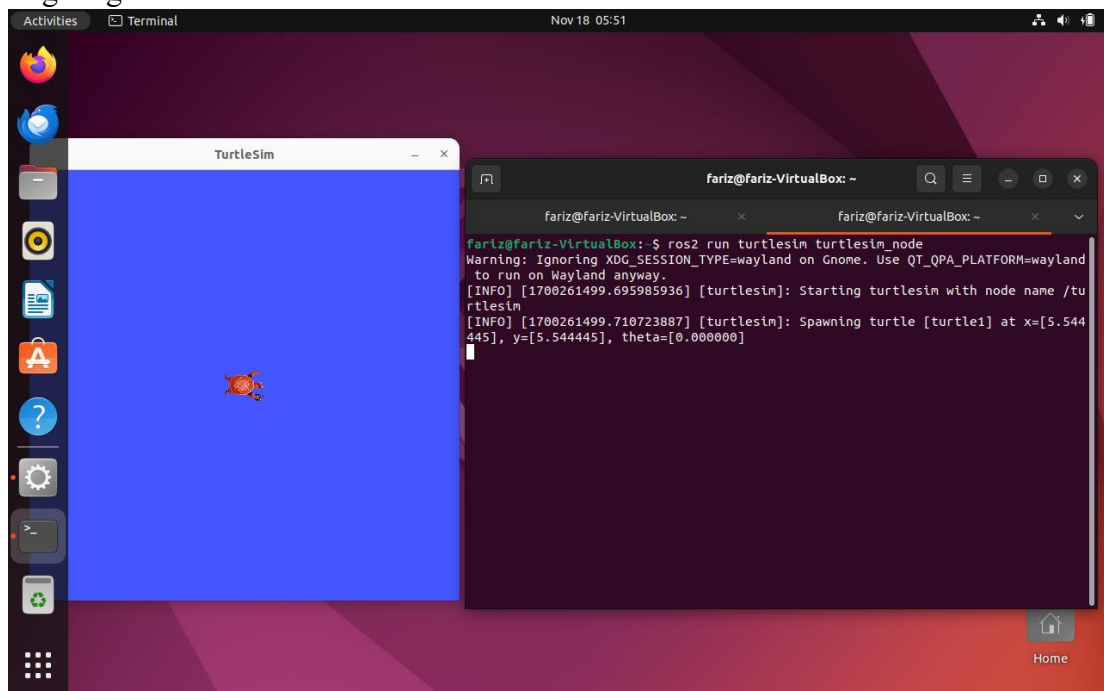
Jadi, ketika menjalankan perintah `sudo apt install ros-galactic-turtlesim`, memberitahu sistem untuk mengunduh dan menginstal paket turtlesim yang sesuai dengan versi ROS Galactic. Setelah proses instalasi selesai, Anda dapat menggunakan simulator turtlesim ini sebagai alat untuk belajar dan menguji konsep-konsep dasar ROS dalam lingkungan simulasi sebelum mengimplementasikannya pada robot fisik.

- Langkah selanjutnya yaitu menginputkan perintah `ros2 pkg executables turtlesim` digunakan untuk menampilkan daftar eksekutabel (executables) yang terkait dengan paket turtlesim dalam lingkungan ROS 2.



executables: Ini adalah sub-perintah yang digunakan untuk menampilkan daftar eksekutabel yang terkait dengan paket ROS tertentu. Hasil dari perintah ini akan menampilkan daftar eksekutabel yang terkait dengan paket turtlesim, sehingga Anda dapat melihat perintah-perintah apa yang dapat Anda jalankan untuk mengakses atau mengontrol simulasi turtlesim.

- Langkah selanjutnya memasukan Perintah `ros2 run turtlesim turtlesim_node` yang digunakan untuk menjalankan node (simpul) `turtlesim_node` dari paket `turtlesim` dalam lingkungan ROS 2.

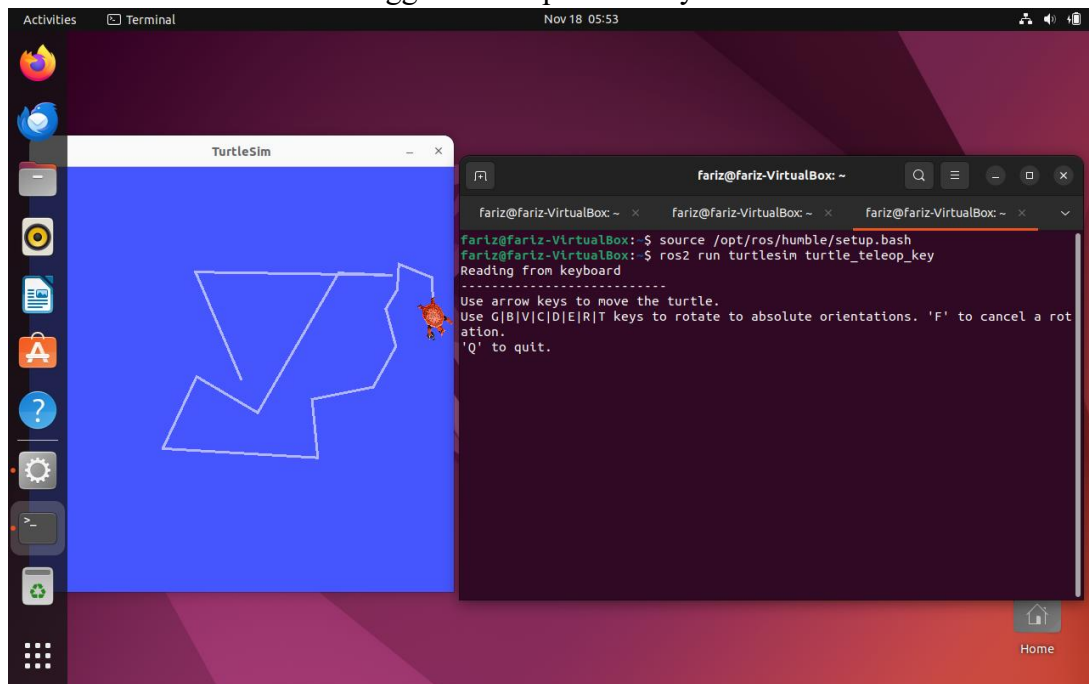


adi, perintah lengkapnya, `ros2 run turtlesim turtlesim_node`, memberi tahu sistem ROS 2 untuk menjalankan node `turtlesim_node` dari paket `turtlesim`. Node ini bertanggung

jawab atas simulasi turtlesim, yang mencakup penggambar turtle di lingkungan 2D dan menerima perintah-perintah untuk menggerakkan turtle.

Ketika Anda menjalankan perintah ini, sistem akan memulai node `turtlesim_node`, dan Anda akan melihat efeknya di lingkungan simulasi. Ini menciptakan antarmuka visual yang memungkinkan Anda untuk mengontrol turtle dan mengamati pergerakannya di lingkungan 2D.

- Terakhir inputkan perintah `ros2 run turtlesim turtle_teleop_key` digunakan untuk menjalankan sebuah node yang memungkinkan Anda mengontrol pergerakan turtle dalam simulasi turtlesim menggunakan input dari keyboard.



Jadi, ketika Anda menjalankan perintah `ros2 run turtlesim turtle_teleop_key`, sistem ROS 2 akan memulai node `turtle_teleop_key` yang memungkinkan Anda mengontrol turtle menggunakan keyboard. Setelah menjalankan perintah ini, Anda dapat membuka terminal dan memastikan bahwa terminal tersebut memiliki fokus (kursor berada di dalamnya). Kemudian, Anda dapat menggunakan tombol-tombol pada keyboard untuk menggerakkan turtle

III. KESIMPULAN

Secara menyeluruh, tutorial ini memberikan gambaran tentang penggunaan ROS 2 dalam konteks simulasi dengan turtlesim. ROS 2, atau Robot Operating System 2, adalah kerangka pengembangan perangkat lunak sumber terbuka yang dirancang untuk mendukung pengembangan aplikasi robotika. Turtlesim, sementara itu, adalah simulator sederhana yang digunakan untuk memahami konsep-konsep dasar ROS 2 melalui simulasi pergerakan turtle di lingkungan 2D.

Langkah-langkah yang telah dijelaskan dalam tutorial mencakup:

- Pembaruan Informasi Paket (`sudo apt update`): Mengupdate daftar paket yang tersedia dari repositori sistem, memastikan informasi paket yang terbaru dapat diakses.
- Pengaturan Variabel Lingkungan (`source /opt/ros/humble/setup.bash`): Memasukkan variabel lingkungan yang diperlukan untuk menggunakan ROS 2 dalam sesi terminal saat ini.
- Instalasi turtlesim (`sudo apt install ros-galactic-turtlesim`): Menginstal paket turtlesim untuk ROS Galactic, memastikan bahwa simulator turtlesim dan dependensinya terpasang.
- Pemeriksaan Eksekutabel Paket (`ros2 pkg executables turtlesim`): Menampilkan daftar eksekutabel yang terkait dengan paket turtlesim, memberikan informasi tentang perintah-perintah yang dapat dijalankan untuk berinteraksi dengan turtlesim.
- Menjalankan turtlesim_node (`ros2 run turtlesim turtlesim_node`): Memulai node untuk mengontrol simulasi turtlesim, menciptakan lingkungan simulasi 2D dengan turtle yang dapat bergerak.
- Menjalankan turtle_teleop_key (`ros2 run turtlesim turtle_teleop_key`): Memulai node untuk mengontrol turtle dalam simulasi menggunakan input keyboard, memberikan pengalaman praktis dalam mengirimkan perintah dan mengamati respons turtle.

Secara keseluruhan, langkah-langkah ini membantu pengguna memahami dasar-dasar ROS 2, konsep-konsep seperti node, paket, dan variabel lingkungan, sambil memberikan pengalaman langsung dalam mengontrol dan mengamati perilaku turtle dalam simulasi turtlesim. Tutorial ini menjadi dasar bagi pengembang untuk terlibat lebih lanjut dalam pengembangan perangkat lunak robotika menggunakan ROS 2.