

# Tugas UAS

## Robotika

Nama : muhammad makhluhi makbullah

Kelas : TK 45 01

NIM : 1103210171

### 1. Publisher dan Subscriber di ROS

Publisher adalah komponen yang bertugas untuk mengirimkan (memublikasikan) pesan ke topik tertentu di ROS. Topik ini merupakan saluran komunikasi tempat pesan dikirim dan diterima. Sebuah Publisher dapat diprogram untuk memublikasikan pesan secara periodik (misalnya, setiap detik) atau berdasarkan event tertentu. Setiap Publisher terkait dengan tipe pesan tertentu (misalnya `std_msgs/String`, `geometry_msgs/Twist`, dll). Tipe pesan ini mendefinisikan struktur data yang dipublikasikan.

Cara kerja Publisher:

- Publisher membuat komunikasi dengan topik tertentu.
- Publisher mengirimkan pesan (data) ke topik tersebut.
- Node lain yang tertarik pada topik tersebut dapat menerima pesan ini melalui Subscriber.

```
Select roscore http://0a9d27a51c73:11311/
Microsoft Windows [Version 10.0.19045.3930]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>docker pull ros:noetic
noetic: Pulling from library/ros
Digest: sha256:456b9a58b63e259c388fb59d8f6c1db6b17967f7b515ebb5aeffffb4eb2fba888
Status: Image is up to date for ros:noetic
docker.io/library/ros:noetic

C:\Users\ASUS>docker images -a
REPOSITORY          TAG          IMAGE ID      CREATED      SIZE
utrarobosoccer/noetic latest       2c24443e0978  4 years ago  1.31GB
ros                  noetic      456b9a58b63e  4 years ago  1.31GB

C:\Users\ASUS>docker run -it --rm ros:noetic
root@0a9d27a51c73:/# roscore
... logging to /root/.ros/log/028278ce-ca41-11ef-9377-0242ac110002/roslaunch-0a9d27a51c73-28.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://0a9d27a51c73:43523/
ros_comm version 1.17.0

SUMMARY
=====

PARAMETERS
* /rostdistro: noetic
* /rosversion: 1.17.0

NODES

auto-starting new master
process[master]: started with pid [36]
ROS_MASTER_URI=http://0a9d27a51c73:11311/

setting /run_id to 028278ce-ca41-11ef-9377-0242ac110002
process[roscout-1]: started with pid [46]
started core service [/roscout]
```

### 1. *docker pull ros:noetic*

- Perintah: `docker pull ros:noetic`
- Fungsi: Perintah ini digunakan untuk mengunduh (download) image Docker ROS versi Noetic dari Docker Hub ke komputer.

Penjelasan lebih lanjut:

`docker pull` adalah perintah untuk mengunduh image Docker dari registry (dalam hal ini Docker Hub).

- `ros:noetic` adalah nama image yang ingin diunduh, di mana `ros` adalah nama repositori dan `noetic` adalah tag untuk versi ROS (versi 1.15 dari ROS). Tag ini menunjukkan versi tertentu dari image ROS yang dibutuhkan.
- Tujuan: Setelah perintah ini dijalankan, image Docker ROS Noetic akan diunduh dan disimpan di sistem, dan siap untuk digunakan untuk menjalankan container ROS.

### 2. *docker images -a*

- Perintah: `docker images -a`
- Fungsi: Perintah ini digunakan untuk menampilkan semua image Docker yang ada di sistem, termasuk image yang tidak lagi digunakan (dangling images).

Penjelasan lebih lanjut:

- `docker images`: Perintah ini menampilkan daftar image Docker yang ada di sistem.
- `-a`: Opsi `-a` (atau `--all`) menunjukkan bahwa perintah ini akan menampilkan semua image Docker yang ada, termasuk image yang tidak terpakai atau dalam status "dangling".
- Tujuan: Perintah ini memungkinkan untuk melihat image yang sudah ada, termasuk ROS Noetic yang baru saja diunduh dengan perintah sebelumnya.

### 3. *docker run -it --rm ros:noetic*

- Perintah: `docker run -it --rm ros:noetic`
- Fungsi: Perintah ini digunakan untuk menjalankan container baru dari image ROS Noetic yang telah diunduh dan masuk ke dalam container tersebut dalam mode interaktif.

Penjelasan lebih lanjut:

- `docker run`: Perintah ini digunakan untuk membuat dan menjalankan container dari image Docker yang telah ada.
- `-it`: Ini adalah dua opsi yang digunakan bersama:
- `-i` (interactive): Menyediakan terminal interaktif untuk menjalankan perintah di dalam container.
- `-t` (tty): Memberikan terminal virtual yang memungkinkan Anda berinteraksi dengan container secara lebih nyaman (seperti terminal biasa).

- `--rm`: Opsi ini memastikan bahwa container akan dihapus secara otomatis setelah Anda keluar atau container dihentikan.
- `ros:noetic`: Ini adalah nama image yang digunakan untuk membuat container, yaitu image ROS Noetic yang telah Anda unduh sebelumnya.

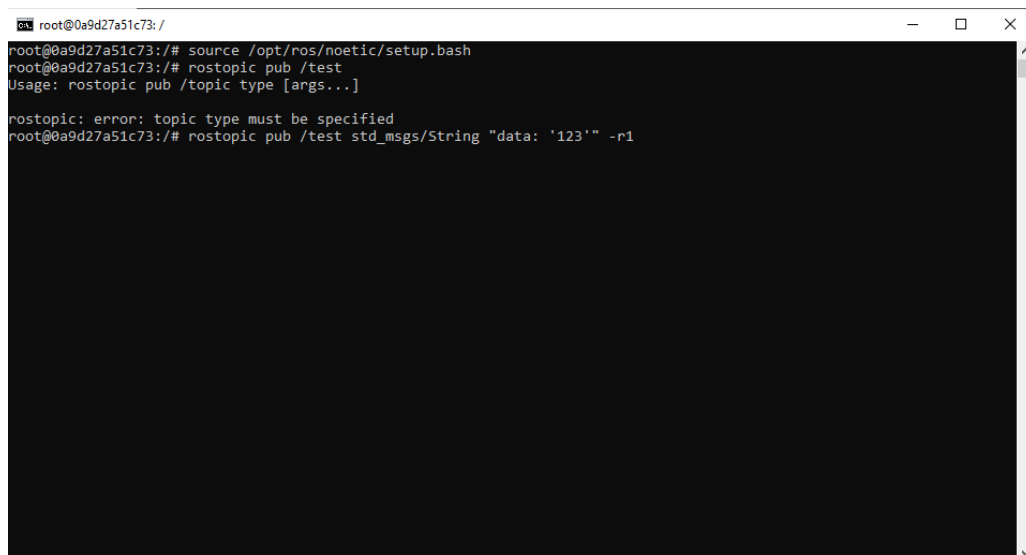
#### 4. roscore

adalah perintah penting dalam ROS (Robot Operating System) yang berfungsi untuk menjalankan ROS Master, yang merupakan komponen pusat dalam sistem ROS. ROS Master memungkinkan komunikasi antar node di dalam sistem ROS dengan mengelola informasi tentang node dan topik.

Penjelasan tentang roscore:

Fungsi Utama:

- roscore bertugas untuk memulai dan menjalankan server utama dalam ROS yang disebut ROS Master.
- ROS Master mengelola komunikasi antara node, topik, layanan, dan parameter dalam sistem ROS.
- Tanpa menjalankan roscore, node-node di ROS tidak dapat berkomunikasi satu sama lain, karena ROS Master yang mengatur bagaimana node terhubung dan berkomunikasi melalui topik dan layanan.



```

root@0a9d27a51c73: /
root@0a9d27a51c73: /# source /opt/ros/noetic/setup.bash
root@0a9d27a51c73: /# rostopic pub /test
Usage: rostopic pub /topic type [args...]

rostopic: error: topic type must be specified
root@0a9d27a51c73: /# rostopic pub /test std_msgs/String "data: '123'" -r1

```

*source /opt/ros/ro/noetic/setup.bash:*

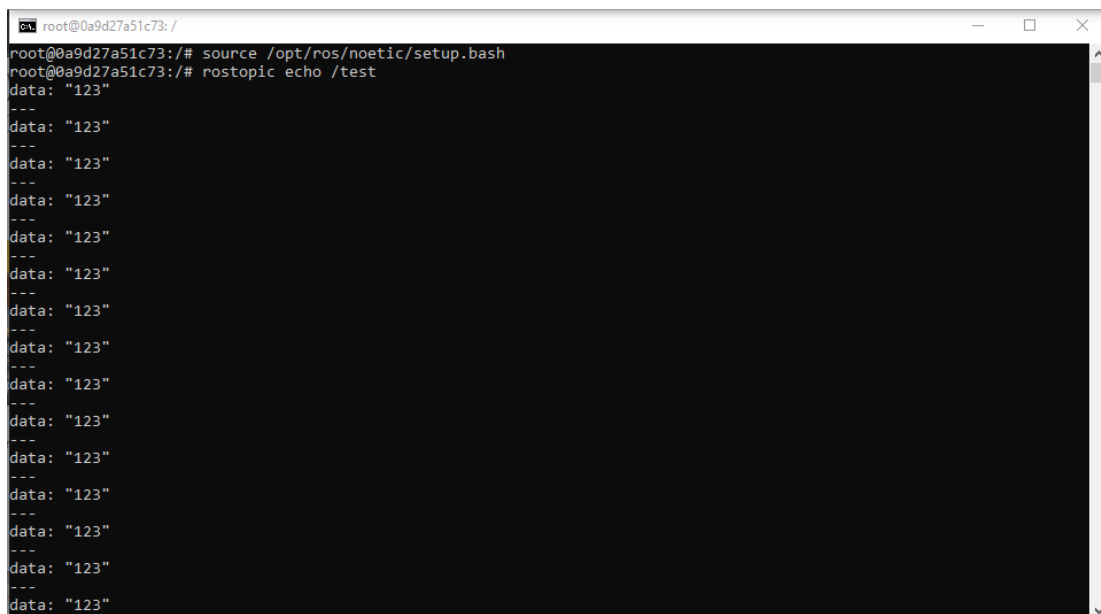
- Perintah ini digunakan untuk menginisialisasi lingkungan ROS (Robot Operating System) agar siap digunakan. ROS memiliki berbagai variabel lingkungan dan pustaka yang perlu diatur sebelum menjalankan perintah ROS lainnya. Perintah ini mengarah ke file `setup.bash` yang ada di instalasi ROS Noetic yang digunakan untuk menyiapkan lingkungan kerja.

### rostopic pub /test:

- Ini adalah perintah untuk menerbitkan (publish) pesan ke suatu topik di ROS. /test adalah nama topik yang digunakan untuk komunikasi antara node di ROS. Biasanya, rostopic pub digunakan untuk mengirim pesan ke suatu topik dalam sistem ROS, dan perintah ini harus diikuti dengan tipe pesan yang akan dipublikasikan.

### rostopic pub /test std\_msgs/String "data: '123'" -r1:

- Perintah lengkap ini juga digunakan untuk mem-publish pesan ke topik /test, namun kali ini lebih spesifik.
- std\_msgs/String adalah jenis pesan yang akan dipublikasikan, yang berarti pesan yang dikirim adalah tipe string.
- "data: '123'" adalah nilai yang akan dikirim dalam pesan. Dalam hal ini, pesan berisi string dengan nilai '123'.
- -r1 adalah parameter untuk mengatur rate atau laju publikasi pesan. Dalam hal ini, laju pengiriman pesan adalah 1 Hz, yang berarti pesan akan dipublikasikan setiap detik.



```
root@0a9d27a51c73: /
root@0a9d27a51c73: /# source /opt/ros/noetic/setup.bash
root@0a9d27a51c73: /# rostopic echo /test
data: "123"
---
data: "123"
---
data: "123"
---
data: "123"
---
data: "123"
---
data: "123"
---
data: "123"
---
data: "123"
---
data: "123"
---
data: "123"
---
data: "123"
---
data: "123"
---
data: "123"
```

### source /opt/ros/ros/noetic/setup.bash:

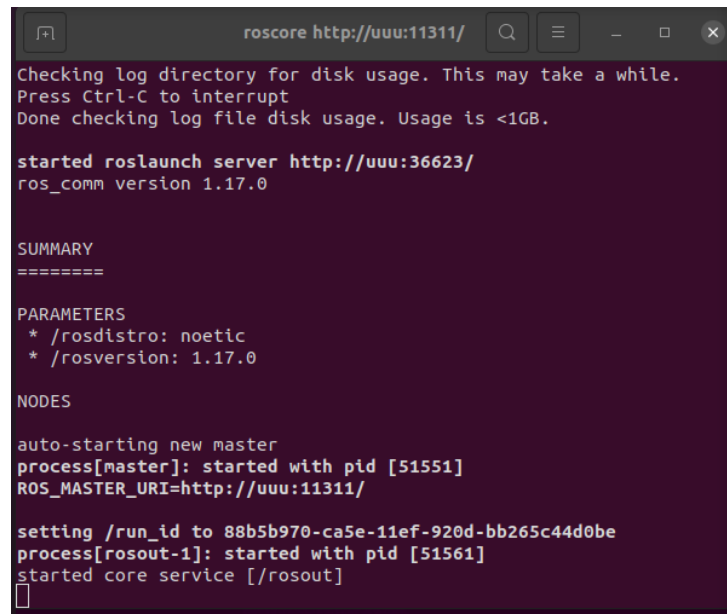
- perintah ini digunakan untuk menginisialisasi lingkungan ROS (Robot Operating System) dengan memuat konfigurasi yang ada di file setup.bash. Ini memastikan bahwa ROS dapat berjalan dengan baik dan dapat mengenali perintah-perintah yang digunakan dalam ROS, seperti rostopic dan lainnya.

### rostopic echo /test:

- Perintah ini digunakan untuk menjadi subscriber pada topik /test. rostopic echo akan menampilkan pesan yang diterima oleh topik tersebut ke terminal.

- Dalam hal ini, subscriber ini akan mendengarkan (subscribe) pada topik /test dan menampilkan pesan yang diterima di layar terminal.
- Jika ada pesan yang dipublikasikan ke topik /test (misalnya menggunakan perintah `rostopic pub /test std_msgs/String "data: '123'" -r1` seperti dalam kode sebelumnya), maka subscriber ini akan menampilkan pesan tersebut pada terminal.

## 2. Turtlesim



```
roscore http://uuu:11311/
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://uuu:36623/
ros_comm version 1.17.0

SUMMARY
=====

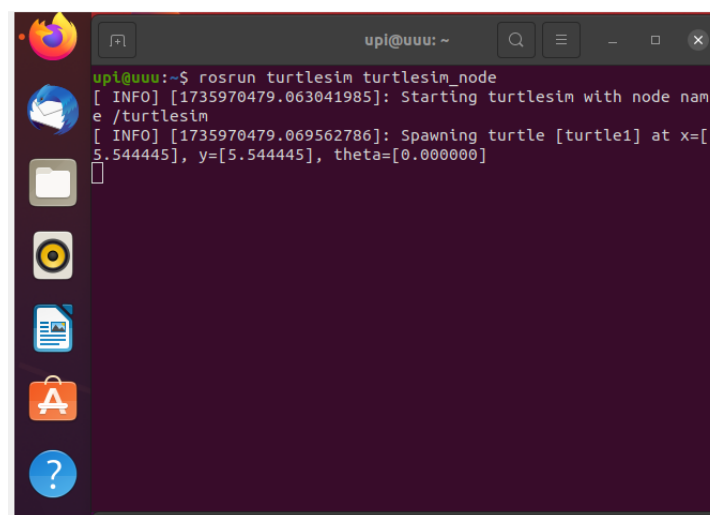
PARAMETERS
* /rosdistro: noetic
* /rosversion: 1.17.0

NODES

auto-starting new master
process[roscout-1]: started with pid [51551]
ROS_MASTER_URI=http://uuu:11311/

setting /run_id to 88b5b970-ca5e-11ef-920d-bb265c44d0be
process[roscout-1]: started with pid [51561]
started core service [/roscout]
```

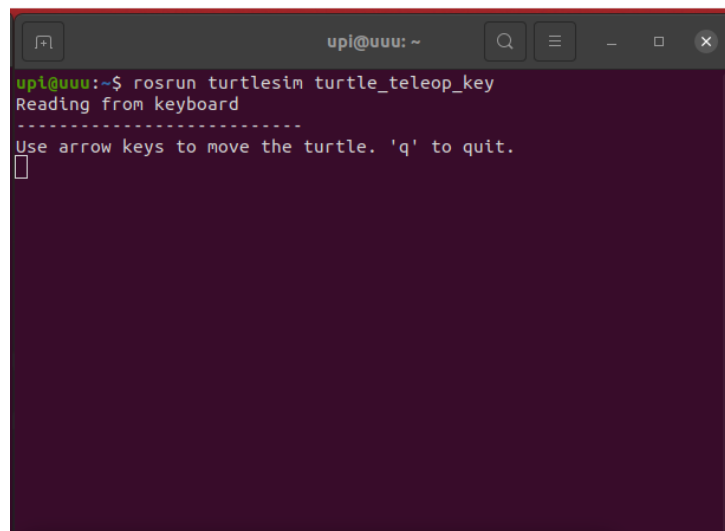
roscore adalah perintah yang digunakan untuk memulai sistem inti dari Robot Operating System (ROS). Ini adalah proses utama yang harus dijalankan sebelum menggunakan ROS, karena semua node ROS memerlukan layanan inti ini untuk berkomunikasi satu sama lain.



```
upi@uuu:~$ roslaunch turtlesim turtlesim_node
[ INFO] [1735970479.063041985]: Starting turtlesim with node name /turtlesim
[ INFO] [1735970479.069562786]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]
```

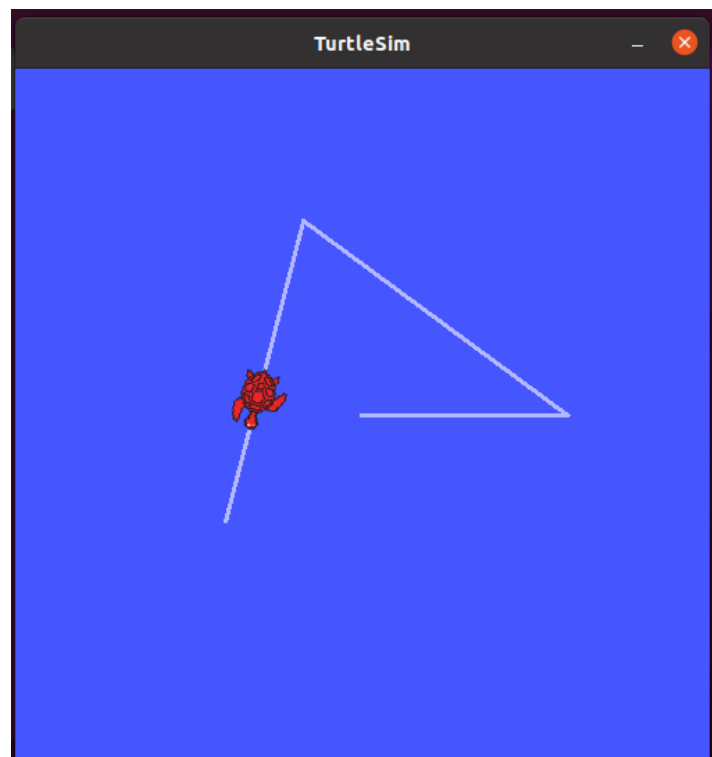
turtlesim\_node adalah node utama dalam package turtlesim. Fungsinya adalah: Membuka jendela simulasi GUI dengan latar biru (mirip kolam). Menampilkan seekor kura-kura yang dapat

digerakkan dengan perintah dari node lain. Node ini menyediakan topik (topics), layanan (services), dan parameter (parameters) untuk interaksi dengan kura-kura.



```
upi@uuu: ~  
upi@uuu:~$ roslaunch turtlesim turtle_teleop_key  
Reading from keyboard  
-----  
Use arrow keys to move the turtle. 'q' to quit.  
█
```

roslaunch turtlesim turtle\_teleop\_key untuk Mengontrol gerakan kura-kura dengan perintah



Tampilan dari kura-kura yang kita perintahkan melalui terminal halaman roslaunch turtlesim turtle\_teleop\_key



- Membuat direktori workspace untuk ROS:
- catkin\_ws adalah workspace utama tempat semua kode ROS Anda akan disimpan.
- src adalah subdirektori tempat Anda akan menambahkan paket-paket ROS.

#### 4. Masuk ke Workspace dan Membangun

cd ~/catkin\_ws

catkin\_make

- cd ~/catkin\_ws: Masuk ke direktori workspace.
- catkin\_make: Membuat workspace dengan menggunakan tool build Catkin. Ini menghasilkan direktori devel/ dan build/ di dalam workspace.

#### 5. Sumberkan Workspace

source devel/setup.bash

- Mengaktifkan workspace sehingga terminal Anda dapat mengenali paket-paket yang ada di dalam workspace ini.

#### 6. Menambahkan Workspace ke File .bashrc

echo "source ~/catkin\_ws/devel/setup.bash" >> ~/.bashrc

source ~/.bashrc

- echo ... >> ~/.bashrc: Menambahkan perintah untuk menyertakan workspace ke file konfigurasi terminal Anda, sehingga setiap kali terminal dibuka, workspace akan otomatis dikenali.
- source ~/.bashrc: Menerapkan perubahan tanpa perlu menutup dan membuka terminal.

#### 7. Menyalin Repositori Gazebo ROS

cd ~/catkin\_ws/src

git clone https://github.com/ros-simulation/gazebo\_ros\_pkgs.git

- cd ~/catkin\_ws/src: Masuk ke direktori sumber dalam workspace.
- git clone ...: Mendownload repositori gazebo\_ros\_pkgs dari GitHub. Repositori ini berisi plugin dan alat tambahan untuk mengintegrasikan Gazebo dengan ROS.

#### 8. Menjalankan Simulasi Dunia Kosong di Gazebo

roslaunch gazebo\_ros empty\_world.launch

- Perintah ini menjalankan Gazebo dengan dunia simulasi kosong menggunakan ROS launch file dari paket gazebo\_ros.
- gazebo\_ros: Paket utama untuk integrasi Gazebo dengan ROS.
- empty\_world.launch: File konfigurasi yang memulai Gazebo dengan dunia kosong (tanpa robot atau objek).



## 4. Introduction

### 1. Tambahkan Repository ROS ke Daftar Sumber Sistem

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

- Menambahkan alamat repository ROS ke dalam file sistem yang digunakan oleh manajer paket Ubuntu untuk menemukan dan mengunduh perangkat lunak.
- `sh -c` memungkinkan Anda menjalankan perintah shell.
- `lsb_release -sc` menghasilkan nama kode rilis Ubuntu (misalnya, "focal" untuk Ubuntu 20.04).
- File yang diperbarui (`ros-latest.list`) digunakan untuk mempermudah akses ke paket ROS.

### 2. Instal curl jika belum terinstal

`sudo apt install curl`

- Menginstal curl, alat command-line untuk mentransfer data dari atau ke server.
- curl dibutuhkan untuk mengunduh kunci GPG di langkah berikutnya.

### 3. Tambahkan Kunci GPG untuk Repository ROS

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

- Menambahkan kunci GPG dari repository ROS ke sistem untuk memastikan bahwa paket berasal dari sumber yang terpercaya.
- curl mengambil file kunci dari URL.
- `sudo apt-key add -` menambahkan kunci ke sistem.

### 4. Perbarui Daftar Paket

`sudo apt update`

- Memperbarui daftar paket yang tersedia agar mencakup paket dari repository ROS.

### 5. Instal ROS Noetic

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

- Menginstal versi "desktop-full" dari ROS Noetic, yang mencakup pustaka ROS, alat pengembangan, dan simulasi Gazebo.

### 6. Tambahkan ROS ke Variabel Lingkungan

`echo "source /opt/ros/noetic/setup.bash" >> ~/.bashrc`

`source ~/.bashrc`

- Menambahkan perintah untuk memuat pengaturan ROS setiap kali terminal baru dibuka.

- ~/.bashrc adalah file konfigurasi shell untuk bash.

#### 7. Tambahkan Konfigurasi untuk Shell Zsh (Opsional)

`echo "source /opt/ros/noetic/setup.zsh" >> ~/.zshrc`

`source ~/.zshrc`

- Sama seperti langkah sebelumnya, tetapi untuk shell Zsh (jika digunakan).

#### 8. Instal Dependensi untuk Workspace ROS

`sudo apt install python3-rosdep python3-rosinstall python3-rosinstall-generator python3-wstool build-essential`

- Menginstal alat tambahan untuk pengembangan ROS, termasuk:
- rosdep: Mengelola dependensi paket.
- rosinstall: Mengelola paket dan workspace ROS.
- build-essential: Alat pengembangan dasar.

#### 9. Instal rosdep (Jika Belum Terinstal)

`sudo apt install python3-rosdep`

- Memastikan bahwa rosdep sudah terinstal.

#### 10. Inisialisasi dan Perbarui rosdep

`sudo rosdep init`

`rosdep update`

- rosdep init: Menyiapkan file konfigurasi awal untuk rosdep.
- rosdep update: Mengunduh dan memperbarui daftar dependensi yang diketahui.