

Nama : Fahmi Adhiwangsa

NIM : 1103204142

## ROS And WEBOTS

Publisher dalam ROS2 adalah perangkat yang digunakan untuk mengirimkan pesan (message) ke sistem ROS2. Publisher ini dapat dibagikan dalam berbagai jenis bahasa pemrograman, seperti Python dan C++. Berikut adalah cara menggunakan publisher dalam ROS2 menggunakan Python:

1. Membuat sebuah konsumen (consumer) dengan mengimpor pergantian Python yang diberikan oleh ROS2. Anda dapat menggunakan perintah berikut untuk membuat konsumen:

```
python
from rclpy.node import Node
from std_msgs.msg import String
```

2. Membuat kelas konsumen yang mengimplementasikan logika untuk menangani pesan yang diterima dari publisher:

```
python
class MinimalSubscriber(Node):
    def __init__(self):
        super().__init__('minimal_subscriber')
        self.subscription = self.create_subscription(String,

    def on_data(self, data):
        print(f'Received data: {data}')
```

3. Membuat objek konsumen dan memulai node ROS2:

```
```python
minimal_subscriber = MinimalSubscriber()
minimal_subscriber.start()
input('Press Enter to stop...')
minimal_subscriber.shutdown()
```

4. Untuk menjalankan node konsumen ini, Anda perlu menjalankan skrip Python di atas dalam terminal yang telah diatur.

Selain itu, Anda juga dapat menggunakan contoh kode yang ada di GitHub untuk menggunakan publisher dan subscriber dalam ROS2 Python, seperti contoh kode yang diberikan dalam tutorial ROS2. Dari sini, Anda dapat belajar cara menggunakan kelas-kelas yang disediakan oleh ROS2 untuk mengembangkan perangkat lunak robotika Anda sendiri menggunakan Python.

## ROS Publiiser vs ROS 2 Publiiser

ROS Publisher	VS	ROS 2 Publisher
The duration and time types are defined in the client libraries, they are in C++ and Python.		In ROS2 these types are defined as messages and therefore are consistent across languages.
Nodes cannot survive without roscore.		Nodes can survive without roscore. As it is end to end connection.
ROS uses a centralised discovery.(Single point of failure )		ROS 2 is fully distributed including discovery.
ROS uses message definition and serialization.(DSS-RPC)		ROS2 is DDS vendor independent. (e.g default FastRTPS can be changed)
ROS can not fulfill real-time requirements so ROS is not mostly used in production scenario.		We can get Real-time system because of we can tweak DDS system configurations (QoS) according to the application.
ROS can only use old user-facing APIs (eg 0.4 "Mango Tango") which was released in 2009		ROS2 is build by considering latest API's.

## Publishes with terminal

