

**URO TEST ITB**  
**SOAL NOMOR 3D**

**Muhammad Faiz Alfada Dharma - 19624244**

Untuk bisa melakukan *control* dan *communication* dengan servo Dynamixel AX-18A, kita dapat menggunakan sistem *publish-subscribe* pada Raspberry Pi. Sistem *publish subscribe* pada Raspberry Pi dapat dilakukan menggunakan python dan ROS2. Dalam menerapkan, hal tersebut, dibutuhkan beberapa perangkat, yaitu Raspberry Pi, Dynamixel AX-18A, dan U2D2. U2D2 adalah *converter communication* USB berukuran kecil yang memungkinkan pengendalian dan pengoperasian Dynamixel dengan PC. Berikut adalah beberapa langkah untuk menggunakan sistem *publish and subscribe* dengan menggunakan python dan ROS2:

1. Membuat *package*

- Buka terminal baru dan *source your ROS2 installation* agar perintah ros2 dapat berfungsi
- Jalankan perintah dibawah ini untuk membuat *package*:  
`ros2 pkg create --build-type ament_python --license Apache-2.0 py_pubsub`

2. Tulis publisher node

- Masuk ke dalam `ros2_ws/src/py_pubsub/py_pubsub`
- Unduh *example talker* dengan memasukkan perintah berikut:
  - a) Linux, wget  
[https://raw.githubusercontent.com/ros2/examples/jazzy/rclpy/topics/minimal\\_publisher/examples\\_rclpy\\_minimal\\_publisher/publisher\\_member\\_function.py](https://raw.githubusercontent.com/ros2/examples/jazzy/rclpy/topics/minimal_publisher/examples_rclpy_minimal_publisher/publisher_member_function.py)
  - b) macOS, wget  
[https://raw.githubusercontent.com/ros2/examples/jazzy/rclpy/topics/minimal\\_publisher/examples\\_rclpy\\_minimal\\_publisher/publisher\\_member\\_function.py](https://raw.githubusercontent.com/ros2/examples/jazzy/rclpy/topics/minimal_publisher/examples_rclpy_minimal_publisher/publisher_member_function.py)
  - c) Windows, curl -sk  
`https://raw.githubusercontent.com/ros2/examples/jazzy/rclpy/topics/minimal_publisher/examples_rclpy_minimal_publisher/publisher_member_function.py -o publisher_member_function.py`
- Kemudian akan ada file baru dengan nama *publisher\_member\_function.py*
- Open file dengan menggunakan code dibawah ini

```

import rclpy
from rclpy.node import Node

from std_msgs.msg import String

class MinimalPublisher(Node):

    def __init__(self):
        super().__init__('minimal_publisher')
        self.publisher_ = self.create_publisher(String, 'topic', 10)
        timer_period = 0.5 # seconds
        self.timer = self.create_timer(timer_period, self.timer_callback)
        self.i = 0

    def timer_callback(self):
        msg = String()
        msg.data = 'Hello World: %d' % self.i
        self.publisher_.publish(msg)
        self.get_logger().info('Publishing: "%s"' % msg.data)
        self.i += 1

```

```

def main(args=None):
    rclpy.init(args=args)

    minimal_publisher = MinimalPublisher()

    rclpy.spin(minimal_publisher)

    # Destroy the node explicitly
    # (optional - otherwise it will be done automatically
    # when the garbage collector destroys the node object)
    minimal_publisher.destroy_node()
    rclpy.shutdown()

if __name__ == '__main__':
    main()

```

### 3. Tulis *subscriber node*

- Kembali ke `ros2_ws/src/py_pubsub/py_pubsub` untuk membuat *node* berikutnya. Masukkan kode berikut di terminal,
  - a) Linux, wget  
`https://raw.githubusercontent.com/ros2/examples/jazzy/rclpy/topics/minimal_subscriber/examples_rclpy_minimal_subscriber/subscriber_member_function.py`
  - b) macOS, wget  
`https://raw.githubusercontent.com/ros2/examples/jazzy/rclpy/topics/minimal_subscriber/examples_rclpy_minimal_subscriber/subscriber_member_function.py`
  - c) Windows, curl -sk  
`https://raw.githubusercontent.com/ros2/examples/jazzy/rclpy/topics/minimal_subscriber/examples_rclpy_minimal_subscriber/subscriber_member_function.py -o subscriber_member_function.py`
- Examine the code  
Buka file `subscriber_member_function.py` dengan menggunakan text editor

```
import rclpy
from rclpy.node import Node

from std_msgs.msg import String

class MinimalSubscriber(Node):

    def __init__(self):
        super().__init__('minimal_subscriber')
        self.subscription = self.create_subscription(
            String,
            'topic',
            self.listener_callback,
            10)
        self.subscription # prevent unused variable warning

    def listener_callback(self, msg):
        self.get_logger().info('I heard: "%s"' % msg.data)
```

```
def main(args=None):  
    rclpy.init(args=args)  
  
    minimal_subscriber = MinimalSubscriber()  
  
    rclpy.spin(minimal_subscriber)  
  
    # Destroy the node explicitly  
    # (optional - otherwise it will be done automatically  
    # when the garbage collector destroys the node object)  
    minimal_subscriber.destroy_node()  
    rclpy.shutdown()  
  
if __name__ == '__main__':  
    main()
```

4. *Build and run* dan pastikan bahwa sistem *publish-subscribe* berjalan dengan baik