Nama : Cathleen Gracia
NRP : 5025231018

Laporan Penugasan 2

1. Langkah pertama dalam pengerjaan penugasan ini adalah membuat workspace dengan command mkdir -p (nama-workspace)/src
    - mkdir -p PenugasanBayucaraka/src
2. Setelah itu navigasikan ke workspace tersebut dengan command cd (nama-workspace)/src
    - cd PenugasanBayucaraka/src
3. Clone repository dengan command git clone (link)
   git clone https://github.com/cthleen/MagangBayu24-ROS2.git
4. Buat package dengan command ros2 pkg create --build-type ament_python {nama-package} untuk python
    - ros2 pkg create --build-type ament_python tugas2
5. Tambahkan publisher node pada direktori tugas2/tugas2
    - cd tugas2/tugas2
    - wget https://raw.githubusercontent.com/ros2/examples/humble/rclpy/topics/minimal_publisher/examples_rclpy_minimal_publisher/publisher_member_function.py
6. Tambahkan dependencies pada package.xml
    - code . (untuk membuka di vscode)
    - tambahkan ini di package.xml
      <exec_depend>rclpy</exec_depend>
      <exec_depend>std_msgs</exec_depend>
7. Tambahkan entry point pada setup.py
    - entry_points={
        'console_scripts': [
            'talker = tugas2.publisher_member_function:main',
        ],
      },
8. Tambahkan subsciber node
    - wget https://raw.githubusercontent.com/ros2/examples/humble/rclpy/topics/minimal_subscriber/examples_rclpy_minimal_subscriber/subscriber_member_function.py
9. Tambakan lagi entry point
    - entry_points={
        'console_scripts': [
            'talker = tugas2.publisher_member_function:main',
            'listener = tugas2.subscriber_member_function:main',
        ],
      },
10. Buat codingan pada publisher_member_function.py dan subscriber_member_function.py
11. Lakukan command colcon build untuk build package tersebut
12. Lakukan command source install/setup.bash

13. Untuk menjalankan talker node menggunakan command ros2 run py_pubsub talker
14. Untuk menjalankan listener node menggunakan command ros2 run py_pubsub listener

Penjelasan codingan

Pada publisher_member_function.py bagian *__init__*, saya menambahkan publisher sehingga menjadi seperti berikut.

```
def __init__(self):
        super().__init__('minimal_publisher')
        self.publisher1 = self.create_publisher(String, 'topic1', 10)
        self.publisher2 = self.create_publisher(String, 'topic2', 10)
        timer_period = 1 # seconds
        self.timer = self.create_timer(timer_period, self.timer_callback)
        self.i = 0
```

Dapat dilihat terdapat 2 publisher dengan 2 topic berbeda.

Berikut adalah sintaks pada *timer_callback*.

```
def timer_callback(self):
        self.i += 1

        istrue1 = self.i % 2 == 0
        istrue2 = self.i % 3 == 0

        msg1 = String()
        msg1.data = str(istrue1)
        self.publisher1.publish(msg1)
        self.get_logger().info('Publisher - 1 - (%d sec) -> "%s"' % (self.i, msg1.data))

        msg2 = String()
        msg2.data = str(istrue2)
        self.publisher2.publish(msg2)
        self.get_logger().info('Publisher - 2 - (%d sec) -> "%s"' % (self.i, msg2.data))
```

Sintaks *self.i += 1* berguna untuk menambahkan nilai setiap kali *timer_callback* dipanggil.
Sintaks *istrue1* dan *istrue2* berguna untuk mengecek apakah detik tersebut kelipatan 2 dan 3. Apabila merupakan kelipatan 2 dan 3, *istrue1* dan *istrue2* akan bernilai True.
Sintaks *msg1 = String()* berarti menggunakan tipe data pesan berupa string.
Sintaks *msg.data* = str(istrue1) berguna untuk menyimpan data. *istrue1* awalnya merupakan boolean, tetapi diubah menjadi string dengan *str(istrue1)*.
Sintaks *self.publisher1.publish(msg1)* berguna untuk mengirimkan pesan kepada sistem.
Sintaks *self.get_logger().info('Publisher - 1 - (%d sec) -> "%s"' % (self.i, msg1.data))* berguna untuk menampilkan informasi pada log.

Pada subsciber_member_function.py bagian *__init__*, codingan saya seperti berikut.

```
def __init__(self):
        super().__init__('minimal_subscriber')
        self.subscription1 = self.create_subscription(
                String,
```

```
            'topic1',
            self.listener_callback1,
            10)
        self.subscription1

        self.subscription2 = self.create_subscription(
            String,
            'topic2',
            self.listener_callback2,
            10)
        self.subscription2

        self.received1 = None
        self.received2 = None
```

Terdapat 2 objek subscriber yang digunakan untuk menerima pesan dari 2 topik berbeda. Lalu dilanjutkan dengan sintaks sebagai berikut.

```
def listener_callback1(self, msg):
        self.received1 = msg.data

def listener_callback2(self, msg):
        self.received2 = msg.data
        self.check()
```

Sintaks *listerner_callback* akan dijalankan setiap ada pesan baru yang diterima subscriber.
Sintaks *self.received* berguna untuk menyimpan data dari pesan.
Sintaks *self.check()* untuk memanggil fungsi check.

```
def check(self):
        if self.received1 == 'True' and self.received2 == 'True':
        self.get_logger().info(f'pub1 - {self.received1} | pub2 - {self.received2} -> sudah siap nih gass min!')
        else:
        self.get_logger().info(f'pub1 - {self.received1} | pub2 - {self.received2} -> tunggu dulu, kami belum ready!')
        self.received1 = None
        self.received2 = None
```

Setelah itu, pesan dari publisher 1 dan publisher 2 di cek. Apabila keduanya berupa ‘True’, pada log akan dicetak *sudah siap nih gass min!*. Selain dari itu, pada log akan dicetak *tunggu dulu, kami belum ready!.*