

Nama : Cathleen Gracia
NRP : 5025231018

Laporan Penugasan 1

1. Langkah pertama dalam pengerjaan penugasan ini adalah membuat workspace dengan command `mkdir -p (nama-workspace)/src`
 - `mkdir -p PenugasanBayucaka/src`
2. Setelah itu navigasikan ke workspace tersebut dengan command `cd (nama-workspace)/src`
 - `cd PenugasanBayucaka/src`
3. Clone repository dengan command `git clone (link)`
`git clone https://github.com/cthleen/MagangBayu24-ROS2.git`
4. Buat package dengan `ros2 pkg create --build-type ament_python {nama-package}` untuk python
 - `ros2 pkg create --build-type ament_python tugas1`
5. Tambahkan publisher node pada direktori `tugas1/tugas1`
 - `cd tugas1/tugas1`
 - `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 = tugas1.publisher_member_function:main',`
 `],`
 `}`
8. Tambahkan subscriber node
 - `wget`
`https://raw.githubusercontent.com/ros2/examples/humble/rclpy/topics/minimal_subscriber/examples_rclpy_minimal_subscriber/subscriber_member_function.py`
9. Tambahkan lagi entry point
 - `entry_points={`
 `'console_scripts': [`
 `'talker = tugas1.publisher_member_function:main',`
 `'listener = tugas1.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 `timer_callback`, saya membuat 3 variabel yaitu `num1`, `num2`, dan `num3` yang akan digunakan untuk menempatkan 3 angka random.

Untuk mendapatkan angka randomnya, saya menggunakan sintaks `randint(a, b)` untuk mendapatkan bilangan bulat acak.

```
num 1 = random.randint (1, 100000)
```

```
num 2 = random.randint (1, 100000)
```

```
num 3 = random.randint (1, 100000)
```

Dari sintaks tersebut akan didapat nilai bilangan bulat acak pada rentang 1-100.000.

Lalu, saya juga membuat variabel `list_op` untuk menyimpan operator.

```
list_op = ['+', '-', '*', '/', '%']
```

Lalu, saya menggunakan sintaks `random.choice(list_op)` untuk memilih operator secara acak.

```
operator1 = random.choice(list_op)
```

```
operator2 = random.choice(list_op)
```

Setelah itu, saya melanjutkannya dengan ini.

```
msg = String()
```

```
msg.data = f'{num1} {operator1} {num2} {operator2} {num3}'
```

```
self.publisher_.publish(msg)
```

```
self.get_logger().info("Publishing: \"%s\" % msg.data)
```

Sintaks `msg = String()` berarti menggunakan tipe data pesan berupa string.

Sintaks `msg.data = f'{num1} {operator1} {num2} {operator2} {num3}'` berguna untuk menyimpan data.

Sintaks `self.publisher_.publish(msg)` berguna untuk mengirimkan pesan kepada sistem.

Sintaks `self.get_logger().info("Publishing: \"%s\" % msg.data)` berguna untuk menampilkan informasi pada log.

Pada `subscriber_member_function.py` bagian `listener_callback`, saya membuat variabel `expression` yang menerima data dari pesan.

```
expression = msg.data
```

Lalu, saya menggunakan sintaks `eval()` untuk menghitung operasi yang ada pada pesan tersebut.

```
result = eval(expression)
```

Setelah itu, hasil dari operasi tersebut dicetak pada log dengan sintaks berikut.

```
self.get_logger().info(f'Result: {expression} = {result}')
```