

Nama : Reihan Arianza

NRP : 5025231274

- Membuat ros2 package pada file yang telah di clone dari fork
- Pada file publisher_member_function.py dan pada def __init__(self) saya ketik seperti berikut :

```
/
def __init__(self):
    super().__init__('minimal_publisher')
    self.publisher1 = self.create_publisher(String, 'pub1', 10)
    self.publisher2 = self.create_publisher(String, 'pub2', 10)
    timer_period = 1 # seconds
    self.timer = self.create_timer(timer_period, self.timer_callback)
    self.i = 0
```

- Saya membuat 2 publisher langsung dengan timer period 1 detik
- Pada def timer_callback(self) saya ketik seperti berikut :

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

    cek1 = self.i % 2 == 0 #ngecek pub 1 setiap true
    cek2 = self.i % 3 == 0 #ngecek pub 2 setiap true

    # publisher 1
    msg1 = String()
    msg1.data = str(cek1)
    self.publisher1.publish(msg1)
    self.get_logger().info('Publisher - 1 - (%d sec) -> %s' %
(self.i,msg1.data))

    # publisher 2
    msg2 = String()
    msg2.data = str(cek2)
    self.publisher2.publish(msg2)
    self.get_logger().info('Publisher - 2 - (%d sec) -> %s' %
(self.i,msg2.data))
```

- Self.i += 1 digunakan untuk menghitung timer yang bertambah tiap detik
- cek1 dan cek2 digunakan untuk mengecek apakah true atau false dengan ketentuan yang telah diberi dimana cek1 interval true/false 2 detik dan cek2 interval true/false 3detik
- Mengoutput berapa detik dan string untuk true atau false pada cek1 dan cek2
- Pada file subscriber_member_function.py dan pada def __init__(self) saya ketik seperti berikut :

```

/
def __init__(self):
    super().__init__('minimal_subscriber')
    self.subscription1 =
self.create_subscription(String,'pub1',self.listener_callback1,10)
    self.subscription1 # prevent unused variable warning

    self.subscription2 =
self.create_subscription(String,'pub2',self.listener_callback2,10)
    self.subscription2 # prevent unused variable warning

    self.recieve1 = None
    self.recieve2 = None

```

- Saya mendeklarasikan 2 subscription agar bisa menangkap kedua publisher
- Self.recieve1 dan 2 = None agar tidak terjadi *UnboundLocalError* dimana menggunakan variable sebelum terdeklarasi
- Pada listener callback yaitu ada 2 saya ketik seperti berikut :

```

/
def listener_callback1(self, msg):
    self.recieve1 = msg.data

def listener_callback2 (self, msg):
    self.recieve2 = msg.data
    self.cek()

```

- Saya menggunakan recieve 1 dan 2 untuk menyimpan data yang telah dikirim oleh publisher 1 dan 2
- Saya mendeklarasikan cek untuk nanti mengecek true dan false-nya

- Pada cek saya ketik sebagai berikut:

```
/
def cek(self):
    if self.recieve1 == 'True' and self.recieve2 == 'True':
        self.get_logger().info(f"pub1 - {self.recieve1} | pub2 - {self.recieve2} -> sudah siap nih, gass min!")
    else:
        self.get_logger().info(f"pub1 - {self.recieve1} | pub2 - {self.recieve2} -> tunggu dulu, kami belum ready!")
        self.recieve1 = None
        self.recieve2 = None
```

- Disini saya membuat ketentuan dimana jika padadata reciever 1 dan 2 adalah true, maka akan mengoutput data yang diterima dan “sudah siap nih, gass min!”, dan jika salah satunya pun bukan true maka akan tetap mengoutput data yang diterima tapi akan mengeluarkan “tunggu dulu, kami belom ready!”
- Saya mengubah data kedua reciever menjadi none agar subscriber selanjutnya pada saat menerima data dari publisher bisa mencegah error