

Laporan dokumentasi pengerjaan tugas 3 Ros2 Bayucaraka 2024

Farrel Ganendra | 5024231036

Sama seperti pada tugas 1 dan 2, saya membuat folder bernama “tugas3” di repositori lokal yang saya clone ke komputer saya. Kemudian saya mulai dengan membaca-baca dokumentasi dari ros2 untuk memahami turtlesim. Disana saya mendapat informasi bahwa node turtlesim mendapatkan perintah untuk bergerak dengan subscribe kepada topik “turtle1/cmd_vel” dengan tipe pesan “Twist” atau suatu pesan yang berisi 2 vektor 3 dimensi (1 untuk translasi dan 1 lagi untuk rotasi). Sehingga saya berniat untuk membuat publisher yang mempublish Twist ke topik tersebut.

Awalnya saya mencoba membuat publisher sama seperti pada tugas pertama dan kedua, lalu mengubah topik ke “turtle1/cmd_vel”, mengganti dependency pada std_msgs ke geometry_msgs, meng-include header “geometry_msgs/msg/twist.hpp” dan menghilangkan include header “std_msgs/msg/string.hpp”. Kemudian memodifikasi code supaya mengirimkan pesan twist. Setelah itu saya mencoba build dan ternyata gagal. Mungkin karena saya asal asalan mengubah dengan cara mencari semua “std_msgs::msg::String” dan mengubahnya menjadi “geometry_msgs::msg::Twist” dan mengubah String() menjadi Twist(). Karena saya tidak tahu apa lagi yang dapat saya ubah supaya publisher dapat di build, saya pun memutuskan untuk berpindah ke python karena di python, syntax yang digunakan tidak perlu se spesifik saat menggunakan c++.

Saya pun menghapus semua file di tugas3, membuat folder src baru, navigasi kedalam file src tersebut, dan membuat package dalam bahasa python dengan menggunakan command berikut.

```
ros2 pkg create --build-type ament_python py_pubsub
```

Lalu menavigasi ke py_pubsub/py_pubsub dan mendownload contoh file publisher dari website dokumentasi ros2 foxy dengan command berikut

```
wget  
https://raw.githubusercontent.com/ros2/examples/foxy/rclpy/topics/minimal_publisher/examples_rclpy_minimal_publisher/publisher_member_function.py
```

Lalu saya melengkapi description, email, dan license pada file package.xml dan mencocokkannya pada file setup.py. Saya menambahkan beberapa dependency berikut pada package.xml

```
<exec_depend>rclpy</exec_depend>  
<exec_depend>std_msgs</exec_depend>  
<exec_depend>geometry_msgs</exec_depend>
```

Dan menambahkan perintah untuk colcon membuat executable dengan cara menambahkan line berikut pada setup.file

```
entry_points={
    'console_scripts': [
        'talker = py_pubsub.publisher_member_function:main',
    ],
},
```

Lalu saya mengimport Twist dengan menambahkan line berikut di header

```
from geometry_msgs.msg import Twist
```

Setelah itu saya mengubah topik yang dituju menjadi “turtle1/cmd_vel”, mengubah timer antara pengiriman data menjadi 2 detik, mengganti tipe msg menjadi Twist(), dan mengirimkan pesan translasi x = 2 dan rotasi z = 2. Terakhir saya menghilangkan line untuk print log karena hal tersebut meminta string dari data msg yang dikirim sedangkan msg yang dikirim sudah saya ganti menjadi tipe Twist. Kemudian saya mencoba run turtlesim seperti biasa, membuild package yang sudah saya modifikasi, dan menjalankannya. Turtle pun berhasil bergerak ke kanan.

Mengetahui metode pengiriman perintah saya sudah berhasil, saya tinggal menentukan perintah yang dikirim supaya mengikuti pola yang sudah ditugaskan. Hal ini dengan membuat 2 list python yang masing masing adalah urutan value dari translasi x dan rotasi z (karena turtlesim hanya dapat bergerak maju mundur (parameter linear.x) dan juga berotasi pada 1 sumbu saja (parameter angular.z)). Lalu pada metode pengiriman pesan, value msg.linear.x dan value msg.angular.z saya assign sebagai anggota dari list list tadi dengan index “self.i” yang terus berhitung dari 0. Berikut detil nya.

```
def timer_callback(self):
    msg = Twist()
    msg.linear.x = translasi[self.i]
    # msg.linear.y = 0.0
    msg.angular.z = rotasi__[self.i]
    self.publisher_.publish(msg)
    self.i += 1
```

```
#commands
translasi = [0.00, 4.00, 0.00, 4.00, 0.00, 4.00, 0.00, 6.28, -0.1, 0.000, 6.28, -0.2, 0.000, 6.28]
rotasi__ = [2.08, 0.00, 2.08, 0.00, 2.08, 0.00, 0.52, 3.14, 0.00, -1.04, 3.14, 0.00, -1.04, 3.14]
```

Command yang saya berikan adalah rotasi 120 derajat, bergerak dengan kecepatan 4, rotasi 120 derajat, bergerak dengan kecepatan 4, rotasi sebesar 30 derajat, bergerak dengan kecepatan 2π dan sekaligus rotasi sebesar 180 derajat, kemudian rotasi sebesar negatif 60 derajat, bergerak dengan kecepatan 2π dan sekaligus rotasi sebesar 180 derajat, kemudian rotasi sebesar negatif 60 derajat, bergerak dengan kecepatan 2π dan sekaligus rotasi sebesar 180 derajat. Kemudian saya build ulang, jalankan turtle sim dan set posisi awal supaya di posisi pojok kanan bawah segitiga dengan command berikut.

```
ros2 service call /turtle1/teleport_absolute turtlesim/srv/TeleportAbsolute "{x: 7.5, y: 5.0, theta: 0}"
```

```
ros2 service call /clear std_srvs/srv/Empty "{}"
```

Dan lalu menjalankan publisher perintah yang saya buat. Entah kenapa pola yang dihasilkan tidak rapi, ujung tiap setengah lingkaran tidak di posisi yang sama pada ujung segitiga. Hal ini pun saya selesaikan dengan memerintah kan robot untuk mundur sedikit setiap setelah menggambar setengah lingkaran. Alhasil didapat pola yang diminta dan tugas3 pun selesai, berikut hasil akhirnya

