

Nama : Ardhien Fadhillah Suhartono

NIM : 1103204137

Kelas : TK4402

Tutorial 7 ROS 2

Pada tugas tutorial 7 ROS2 saya mengalami error dimana codingan saya tidak terbaca dan dikatakan terdapat error pada kodingannya pada line 20 dan 33. Untuk kesalahan di line 20 seharusnya tidak ada walaupun sudah diubah ubah tetapi sama saja dan untuk line 33, semua kodingan yang terdapat di dalam vs code tidak ada yang sampai linanya berjumlah 33.

```
ardhien@DESKTOP-KNN07D0: $ ros2 run my_robot_controller draw_circle
Traceback (most recent call last):
  File "/home/ardhien/ros2_ws/install/my_robot_controller/lib/my_robot_controller/draw_circle", line 33, in <module>
    sys.exit(load_entry_point('my-robot-controller', 'console_scripts', 'draw_circle')())
  File "/home/ardhien/ros2_ws/build/my_robot_controller/my_robot_controller/draw_circle.py", line 20, in main
    rclpy.init(args=args)
TypeError: unsupported operand type(s) for +: 'NoneType' and 'NoneType'
[ros2run]: Process exited with failure 1
```

Gambar Error

```
my_robot_controller > my_robot_controller > draw_circle.py > main
1  #!/usr/bin/env python3
2  import rclpy
3  from rclpy.node import Node
4  from geometry_msgs.msg import Twist
5
6  class DrawCircleNode(Node):
7      def __init__(self):
8          super().__init__("draw_circle")
9          self.cmd_vel_pub_ = self.create_publisher(Twist, "/turtle1/cmd_vel", 10)
10         self.timer = self.create_timer(0.5, self.send_velocity_command)
11         self.get_logger().info("Draw circle node has been started")
12
13     def send_velocity_command(self):
14         msg = Twist()
15         msg.linear.x = 2.0
16         msg.angular.z = 1.0
17         self.cmd_vel_pub_.publish(msg)
18
19     def main(args=None):
20         rclpy.init(args=args)
21         node = DrawCircleNode()
22         rclpy.spin(node)
23         rclpy.shutdown()
```

Kodingan di Node draw_circle

```

my_robot_controller > package.xml
1  <?xml version="1.0"?>
2  <?xml-model href="http://download.ros.org/schema/package_format3.xsd" schematypens="http://www.w3.org/2001/XMLSchema"?>
3  <package format="3">
4    <name>my_robot_controller</name>
5    <version>0.0.0</version>
6    <description>TODO: Package description</description>
7    <maintainer email="ardhien@todo.todo">ardhien</maintainer>
8    <license>TODO: License declaration</license>
9
10   <depend>roscpp</depend>
11   <depend>geometry_msgs</depend>
12   <depend>turtlesim</depend>
13
14   <test_depend>ament_copyright</test_depend>
15   <test_depend>ament_flake8</test_depend>
16   <test_depend>ament_pep257</test_depend>
17   <test_depend>python3-pytest</test_depend>
18
19   <export>
20     <build_type>ament_python</build_type>
21   </export>
22 </package>
23

```

Kodingan pada package

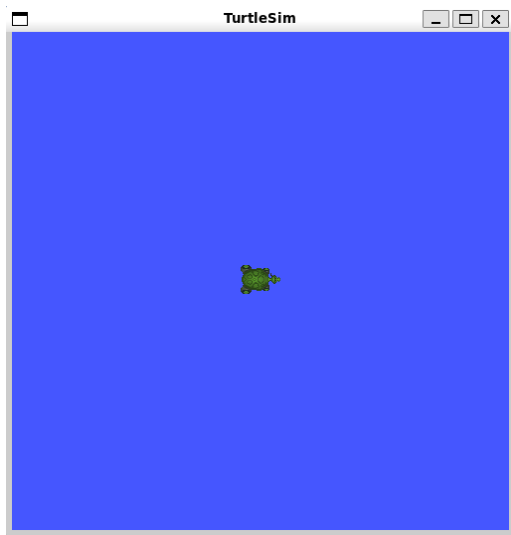
```

my_robot_controller > setup.py ...
1  from setuptools import find_packages, setup
2
3  package_name = 'my_robot_controller'
4
5  setup(
6      name=package_name,
7      version='0.0.0',
8      packages=find_packages(exclude=['test']),
9      data_files=[
10         ('share/ament_index/resource_index/packages',
11          ['resource/' + package_name]),
12         ('share/' + package_name, ['package.xml']),
13     ],
14     install_requires=['setuptools'],
15     zip_safe=True,
16     maintainer='ardhien',
17     maintainer_email='ardhien@todo.todo',
18     description='TODO: Package description',
19     license='TODO: License declaration',
20     tests_require=['pytest'],
21     entry_points={
22         'console_scripts': [
23             'test_node = my_robot_controller.my_first_node:main',
24             'draw_circle = my_robot_controller.draw_circle:main'
25         ],
26     },
27 )

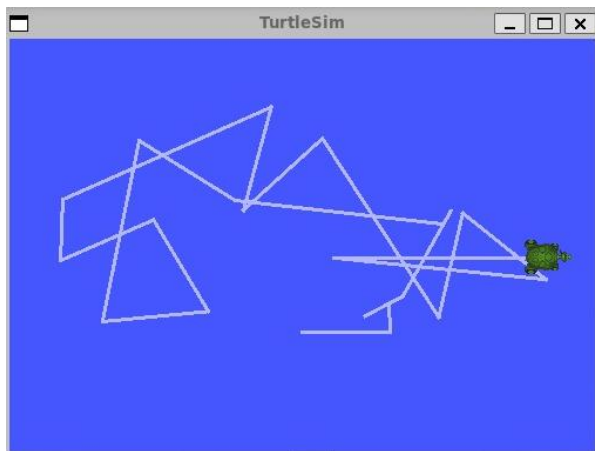
```

Gambar Kodingan Setup

Walaupun masih bisa mengeluarkan turtle pada turtlesim akan tetapi itu tidak dapat bergerak membuat lingkaran sesuai dengan yang ada pada di tutorial.



Dimana saya sudah mencoba mencari berbagai cara melauai git hub dan chatgpt tetapi hasilnya sama saja. Dimana masih muncul error pada line2 pada kodingan seperti diatas. Saya juga sudah meminta tolong kepada teman saya bagaimana cara membenarkannya tapi hasilnya pun masih sama saja. Walaupun pada tutorial sebelumnya sudah bisa bergerak kura kura nya.



Catatan Tutorial

Lecture 2

ROS2 For Beginner Tutorial 8

Pada tutorial ini kita akan dimulai dengan membuat node baru yang dinamakan `pose_subscriber.py`, dimana nantinya kita akan membuat bagaimana subscriber bisa menyambung dengan topic. Langkah selanjutnya ialah dengan membuat subscriber mendapatkan callback dengan menambahkan kodingan di dalam python `pose_callback`. Lalu setelahnya jangan lupa untuk menambahkan `pose_subscriber` ke dalam `setup.py`. Lalu setelah kita membuka `turtlesim` maka nantinya kita akan mendapatkan pesan dari letak `x`, `y`, `theta` dan `linear_velocity`. Lalu apabila kita mencoba menggerakkan kura kuranya maka pesan yang didapatkan juga berubah. Lalu dengan menambahkan kodingan seperti gambar di bawah maka kita akan mendapat nilai nilai yang lebih lengkap.

```
def pose_callback(self, msg: Pose):
    self.get_logger().info("(" + str(msg.x) + ", " + str(msg.y) + ")")
```

[INFO]	[1654767218.694489353]	[pose_subscriber]: (6.345942974090576, 9.369718551635742)
[INFO]	[1654767218.704831442]	[pose_subscriber]: (6.316634654998779, 9.382564544677734)
[INFO]	[1654767218.721614276]	[pose_subscriber]: (6.2871246337890625, 9.394940376281738)
[INFO]	[1654767218.736825851]	[pose_subscriber]: (6.257420063018799, 9.406841278076172)
[INFO]	[1654767218.753871845]	[pose_subscriber]: (6.227529048919678, 9.418266296386719)
[INFO]	[1654767218.769062313]	[pose_subscriber]: (6.197458744049072, 9.429210662841797)
[INFO]	[1654767218.784153132]	[pose_subscriber]: (6.16721773147583, 9.43967342376709)
[INFO]	[1654767218.800116786]	[pose_subscriber]: (6.136812686920166, 9.449650764465332)
[INFO]	[1654767218.816711387]	[pose_subscriber]: (6.106252193450928, 9.45914077758789)
[INFO]	[1654767218.832887229]	[pose_subscriber]: (6.0755438804626465, 9.4681396484375)
[INFO]	[1654767218.849070463]	[pose_subscriber]: (6.0446953773498535, 9.47664737701416)
[INFO]	[1654767218.864007919]	[pose_subscriber]: (6.01371431350708, 9.484659194946289)
[INFO]	[1654767218.880596665]	[pose_subscriber]: (5.982609748840332, 9.492175102233887)
[INFO]	[1654767218.900028456]	[pose_subscriber]: (5.951388359069824, 9.499192237854004)
[INFO]	[1654767218.912753844]	[pose_subscriber]: (5.9200592041015625, 9.505708694458008)
[INFO]	[1654767218.930693547]	[pose_subscriber]: (5.88862943649292, 9.511723518371582)
[INFO]	[1654767218.945016963]	[pose_subscriber]: (5.857107639312744, 9.517233848571777)
[INFO]	[1654767218.963885210]	[pose_subscriber]: (5.825501441955566, 9.522239685058594)

ROS2 For Beginner Tutorial 9

Pada tutorial ini kita akan dimulai dengan membuat node baru yang dinamakan `turtle_controller.py`, dimana setelah membuat node baru jangan lupa untuk menambahkan `turtle_controller` kedalam `setup.py`. Lalu setelahnya kita akan mengimport `pose_subscriber` kedalam `turtle_controller`. Setelah itu kita akan membuat `cmd_vel_publisher` dan akan kita publis dari callback yang kita buat sebelumnya. Kita juga akan mengambil `draw_circle` kedalam `turtle_controller`, lalu kita masukkan kedalam `pose_callback`. Lalu saat kita coba menjalankan `turtlesim`, maka akan berhasil hanya saja akan mentok ke salah satu dinding. Lalu setelahnya kita akan mencoba menambahkan kodingan agar mendapatkan Batasan sebelum menabrak dinding.

```
def pose_callback(self, pose: Pose):
    cmd = Twist()
    if pose.x > 9.0 or pose.x < 2.0 or pose.y > 9.0 or pose.y < 2.0:
        cmd.linear.x = 1.0
        cmd.angular.z = 0.9
    else:
        cmd.linear.x = 5.0
        cmd.angular.z = 0.0
    self.cmd_vel_publisher.publish(cmd)
```



ROS2 For Beginner Tutorial 10

Disini kita akan mencoba menjalankan `add_two_ints_server` akan tetapi tidak terjadi apa apa. Lalu setelahnya kita akan mencoba mengubah service call `add_two_ints_server` menjadi `{'a' : 2, 'b' : 5}` dimana nantinya akan memunculkan incoming request `a: 2 b: 5` pada `add_two_ints_server` sebelumnya. Lalu setelahnya kita akan mencobanya pada `turtlesim` dimana awalnya melakukan service kepada `turtle1/set_pen`, lalu kita memasukkan inputan seperti gambar dibawah maka akan membuat jalur kura kura tersebut berubah.

```
ed@ed-vn: $ ros2 service call /turtle1/set_pen turtlesim/srv/SetPen "{r: 255, 'g': 0, 'b': 0, 'width': 3, 'off': 0}"
requester: making request: turtlesim.srv.SetPen_Request(r=255, g=0, b=0, width=3, off=0)
```



ROS2 For Beginner Tutorial 11

Disini kita akan menambahkan kodingan baru didalam turtle_controller dimana berupa call_set_pen_service. Lalu setelahnya kita akan mencoba memanggil service yang sudah kita buat dengan memasukkan kodingan seperti dibawah.

```
def call_set_pen_service(self, r, g, b, width, off):
    client = self.create_client(SetPen, "/turtle1/set_pen")
    while not client.wait_for_service(1.0):
        self.get_logger().warn("Waiting for service...")

    request = SetPen.Request()
    request.r = r
    request.g = g
    request.b = b
    request.width = width
    request.off = off

    future = client.call_async(request)
    future.add_done_callback(partial(self.callback_set_pen))

def callback_set_pen(self, future):
    try:
        response = future.result()
    except Exception as e:
        self.get_logger().error("Service call failed: %r" % (e,))
```

Setelah itu kita akan menambahkan loop control system di dalam kodingan tersebut. Lalu setelah semua jadi kita akan mencoba menjalankan turtlesim kembali dan dapat dilihat kalau jalur yang lewat kura kura akan berubah menjadi merah dan hijau apabila melawti tengah.

