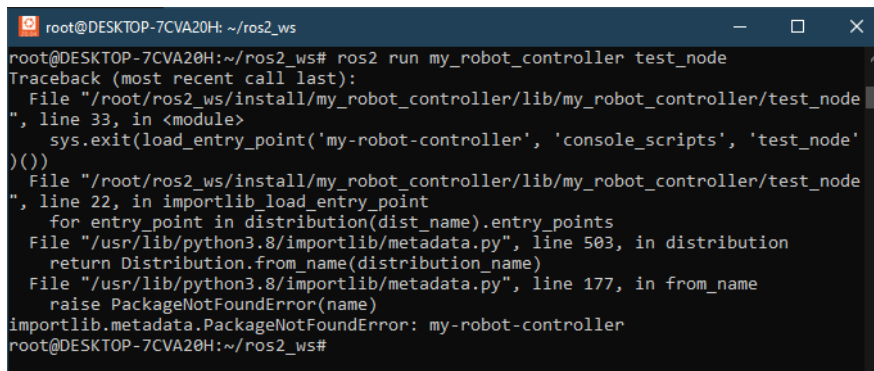


Muhammad Fauzan Nur'ilham
1103204085
TK-44-G7
Robotika

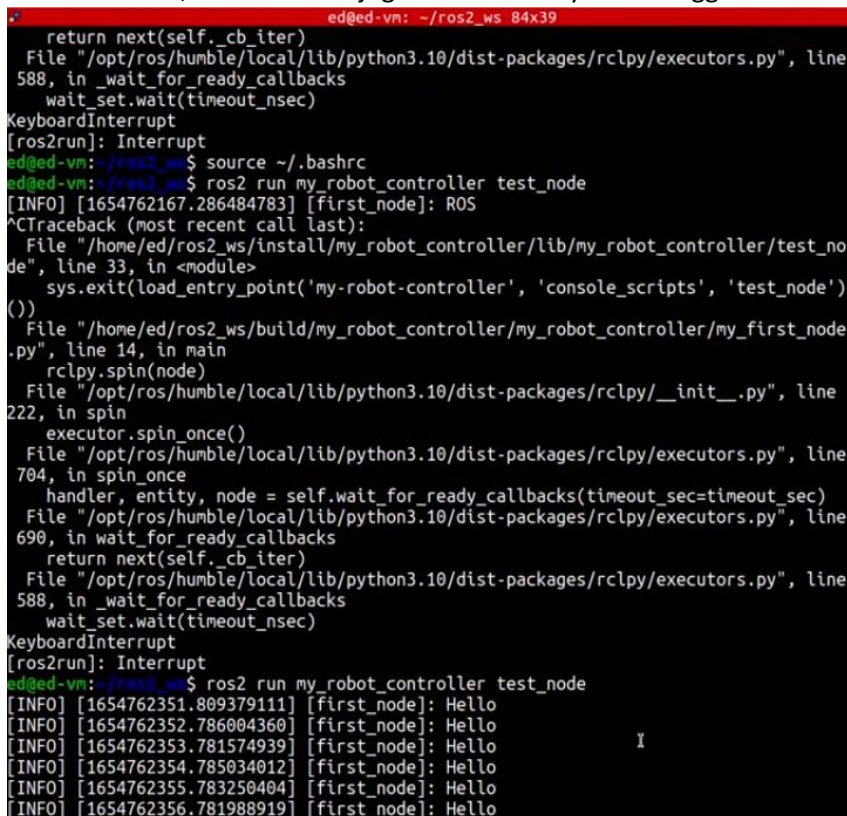
ROS2 Tutorial 5

Pada tutorial 5, saya telah mengikuti tata cara sama persis seperti yang ada di video tutorial hanya saja terdapat error ketika sedang melakukan perintah `ros2 run my_robot_controller test_node` dimana kita mengubah nama domain dalam terminal. Namun nama yang dipanggil tidak keluar dikarenakan error sudah dilakukan pengecekan pada visual studio code dan package list didalamnya dan masih tidak diketahui penyebabnya.



```
root@DESKTOP-7CVA20H: ~/ros2_ws
root@DESKTOP-7CVA20H:~/ros2_ws# ros2 run my_robot_controller test_node
Traceback (most recent call last):
  File "/root/ros2_ws/install/my_robot_controller/lib/my_robot_controller/test_node", line 33, in <module>
    sys.exit(load_entry_point('my-robot-controller', 'console_scripts', 'test_node')())
  File "/root/ros2_ws/install/my_robot_controller/lib/my_robot_controller/test_node", line 22, in importlib.load_entry_point
    for entry_point in distribution(dist_name).entry_points
  File "/usr/lib/python3.8/importlib/metadata.py", line 503, in distribution
    return Distribution.from_name(distribution_name)
  File "/usr/lib/python3.8/importlib/metadata.py", line 177, in from_name
    raise PackageNotFoundError(name)
importlib.metadata.PackageNotFoundError: my-robot-controller
root@DESKTOP-7CVA20H:~/ros2_ws#
```

Gambar dibawah ini merupakan video dari tutorial yang berhasil memanggil nama "ROS" dari main visual studio code, terlihat juga dibawahnya mengganti nama "ROS" menjadi "Hello"



```
ed@ed-vm: ~/ros2_ws 84x39
return next(self.cb_iter)
File "/opt/ros/humble/local/lib/python3.10/dist-packages/rclpy/executors.py", line 588, in _wait_for_ready_callbacks
    wait_set.wait(timeout_nsec)
KeyboardInterrupt
[ros2run]: Interrupt
ed@ed-vm: ~/ros2_ws$ source ~/.bashrc
ed@ed-vm: ~/ros2_ws$ ros2 run my_robot_controller test_node
[INFO] [1654762167.286484783] [first_node]: ROS
^CTraceback (most recent call last):
  File "/home/ed/ros2_ws/install/my_robot_controller/lib/my_robot_controller/test_node", line 33, in <module>
    sys.exit(load_entry_point('my-robot-controller', 'console_scripts', 'test_node')())
  File "/home/ed/ros2_ws/build/my_robot_controller/my_robot_controller/my_first_node.py", line 14, in main
    rclpy.spin(node)
  File "/opt/ros/humble/local/lib/python3.10/dist-packages/rclpy/_init_.py", line 222, in spin
    executor.spin_once()
  File "/opt/ros/humble/local/lib/python3.10/dist-packages/rclpy/executors.py", line 704, in spin_once
    handler, entity, node = self.wait_for_ready_callbacks(timeout_sec=timeout_sec)
  File "/opt/ros/humble/local/lib/python3.10/dist-packages/rclpy/executors.py", line 690, in wait_for_ready_callbacks
    return next(self.cb_iter)
  File "/opt/ros/humble/local/lib/python3.10/dist-packages/rclpy/executors.py", line 588, in _wait_for_ready_callbacks
    wait_set.wait(timeout_nsec)
KeyboardInterrupt
[ros2run]: Interrupt
ed@ed-vm: ~/ros2_ws$ ros2 run my_robot_controller test_node
[INFO] [1654762351.809379111] [first_node]: Hello
[INFO] [1654762352.786004360] [first_node]: Hello
[INFO] [1654762353.781574939] [first_node]: Hello
[INFO] [1654762354.785034012] [first_node]: Hello
[INFO] [1654762355.783250404] [first_node]: Hello
[INFO] [1654762356.781988919] [first_node]: Hello
```

Yang dimana nantinya akan ditambahkan timer dan dilakukan callback dari dalam visual studio code dengan output seperti dibawah ini yaitu “Hello (timer)”

```

ed@ed-vn: ~/ros2_ws 84x39
[INFO] [1654762514.798127365] [first_node]: Hello 31
[INFO] [1654762515.799785173] [first_node]: Hello 32
[INFO] [1654762516.797838693] [first_node]: Hello 33
[INFO] [1654762517.797433472] [first_node]: Hello 34
[INFO] [1654762518.798215562] [first_node]: Hello 35
[INFO] [1654762519.798064125] [first_node]: Hello 36
[INFO] [1654762520.797539153] [first_node]: Hello 37
[INFO] [1654762521.797644900] [first_node]: Hello 38
[INFO] [1654762522.797758072] [first_node]: Hello 39
[INFO] [1654762523.798617418] [first_node]: Hello 40
[INFO] [1654762524.799004110] [first_node]: Hello 41
[INFO] [1654762525.797319507] [first_node]: Hello 42
[INFO] [1654762526.797991164] [first_node]: Hello 43
[INFO] [1654762527.797622059] [first_node]: Hello 44
[INFO] [1654762528.798782108] [first_node]: Hello 45
[INFO] [1654762529.797351847] [first_node]: Hello 46
[INFO] [1654762530.797450806] [first_node]: Hello 47
[INFO] [1654762531.798556629] [first_node]: Hello 48
[INFO] [1654762532.798103395] [first_node]: Hello 49
[INFO] [1654762533.797627857] [first_node]: Hello 50
[INFO] [1654762534.798249165] [first_node]: Hello 51
[INFO] [1654762535.797542903] [first_node]: Hello 52
[INFO] [1654762536.797476998] [first_node]: Hello 53
[INFO] [1654762537.798198424] [first_node]: Hello 54
[INFO] [1654762538.797940500] [first_node]: Hello 55
[INFO] [1654762539.797325910] [first_node]: Hello 56
[INFO] [1654762540.797333518] [first_node]: Hello 57
[INFO] [1654762541.798104240] [first_node]: Hello 58
[INFO] [1654762542.797728272] [first_node]: Hello 59
[INFO] [1654762543.811593806] [first_node]: Hello 60
[INFO] [1654762544.797812845] [first_node]: Hello 61
[INFO] [1654762545.797709323] [first_node]: Hello 62
[INFO] [1654762546.798066019] [first_node]: Hello 63
[INFO] [1654762547.798151879] [first_node]: Hello 64
[INFO] [1654762548.797591438] [first_node]: Hello 65
[INFO] [1654762549.797359132] [first_node]: Hello 66
[INFO] [1654762550.798516073] [first_node]: Hello 67
[INFO] [1654762551.797786756] [first_node]: Hello 68
  
```

ROS2 Tutorial 6

Pada tutorial ini memiliki 3 node yang akan berkomunikasi satu sama lain yaitu talker, chatter, dan listener. Kemudian sama seperti video tutorial 2, pada tutorial 6 ini juga menggunakan turtlesim pada terminal berfungsi untuk memanggil turtlesim, menggerakkan ikon kura-kura pada turtlesim menggunakan keyboard G, B, V, C, D, E, R, T, dan juga melihat interfacenya.

What is a ROS2 Topic? - ROS2 Tutorial 6

```

ed@ed-vn: ~$ ros2 run turtlesim turtlesim
Warning: Ignoring XDG_SESSION_TYPE=wayland to run on Wayland anyway.
[INFO] [1654763729.163572107] [turtlesim]: Starting turtlesim with node name /turtlesim
[INFO] [1654763729.168210284] [turtlesim]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]

ed@ed-vn: ~$ ros2 topic info /turtle1/cmd_vel
Type: geometry_msgs/msg/Twist
Publisher count: 1
Subscription count: 1
# This expresses velocity in free space broken into its linear and angular parts.
Vector3 linear
  float64 x
  float64 y
  float64 z
Vector3 angular
  float64 x
  float64 y
  float64 z
  
```

ed@ed-vn: ~\$ ros2 run turtlesim

ed@ed-vn: ~\$ ros2 run turtlesim turtle_teleop_key

Reading from keyboard

Use arrow keys to move the turtle.

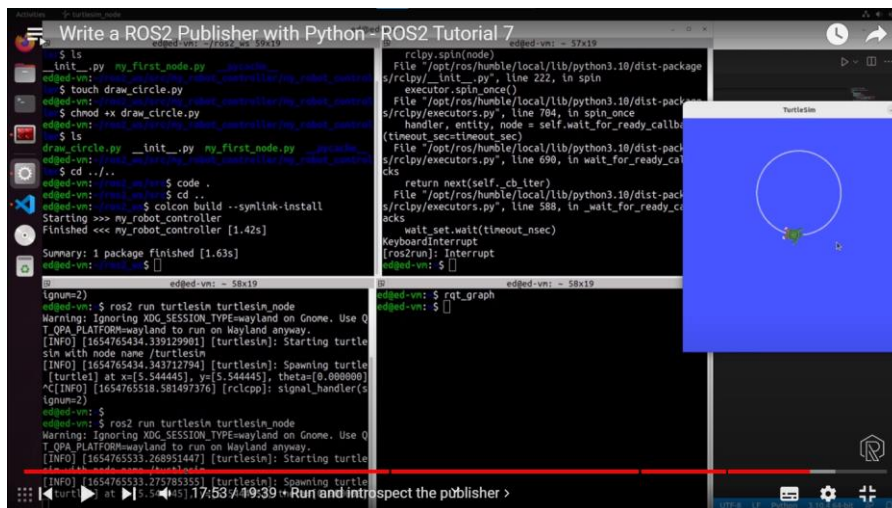
Use G[B|V|C|D|E|R|T] keys to rotate to absolute orientations. 'F' to cancel a rotation.

'Q' to quit.

10:35 / 10:58 • ROS2 Topic explanation 2 with Turtlesim >

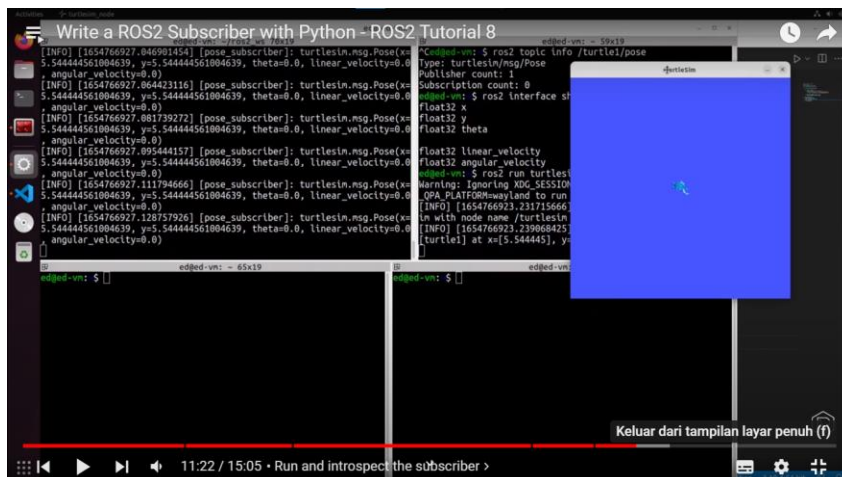
ROS2 Tutorial 7

Sama seperti tutorial sebelumnya memiliki 3 node yang akan berkomunikasi satu sama lain, pada tutorial 7 ini masih menggunakan turtlesim yang menjalankan perintah draw_circle dimana ikon kura-kura tersebut akan membuat sebuah lingkaran secara otomatis tanpa menggunakan keyboard seperti tutorial 6. Turtlesim bisa di interrupt dengan stop dan juga bisa dijalankan kembali menggunakan perintah draw_circle



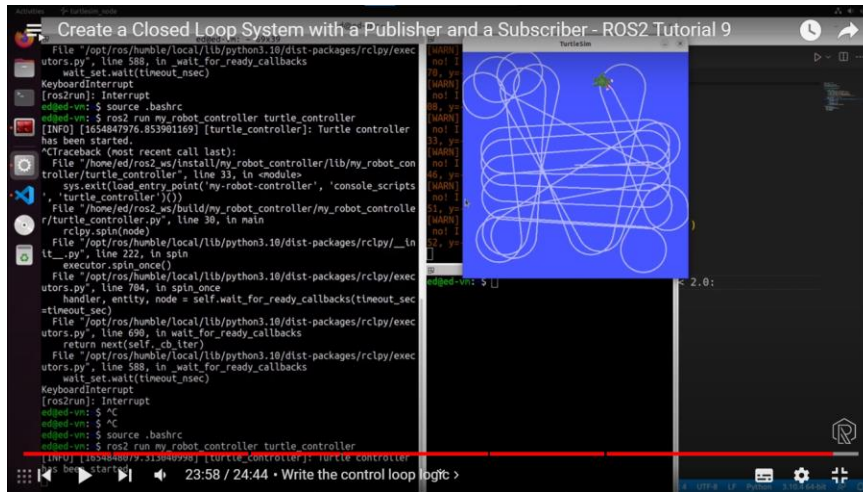
ROS2 Tutorial 8

Sama seperti pada tutorial 7 dimana ikon kura-kura pada turtlesim akan melakukan gerakan melingkar, hanya saja pada tutorial ini ikon kura-kura pada turtlesim akan dilihat nilai pergerakannya pada "pose subscriber" menggunakan nilai x, y, theta, linear_velocity dan angular_velocity contohnya seperti yang ada pada gambar dibawah. Pada terminal menunjukkan ada beberapa nilai yang berbeda dikarenakan ikon kura-kura pada turtlesim akan selalu bergerak secara melingkar, Setiap pergerakan ikon kura-kura akan terekam jejak nilainya.



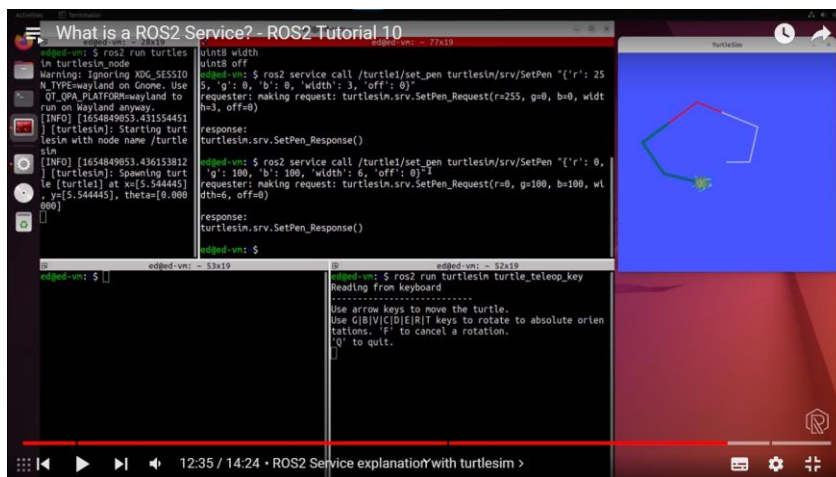
ROS2 Tutorial 9

Pada video tutorial 9 ini, ikon kura-kura pada turtlesim dimanipulasi agar bisa bergerak dengan otomatis tanpa menabrak bagian pinggir terminal yang dianggap sebagai danger zone dengan memasukkan nilai pose_callback pada Visual Studio Code contohnya seperti "if pose.x > 9.0 or pose.x < 2.0 or pose.y > 9.0 or pose.y < 2.0:.. Yang dimana nilai tersebut diambil dari terminal dan jika ikon kura-kura mendekati nilai tersebut akan berbelok tergantung dari nilai yang kita masukkan.



ROS2 Tutorial 10

Pada video tutorial 10 ini, kita bisa mengatur pergerakan ikon kura-kura pada turtlesim seperti pada terminal dibawah menggunakan nilai parameter x, y, dan theta dimana nilainya adalah x=[5.544445], y=[5.544445], theta=[0.000000]. Dan juga Setiap path yang dilewati bisa diubah warnanya dengan r, g, b, width, dan off. Contohnya ros2 service call "{r': 0, 'g': 100, 'b': 100, 'width': 6, 'off': 0}" yang dimana pada parameter tertentu path yang dilewati oleh ikon kura-kura akan berubah warna.



ROS2 Tutorial 11

Sama seperti tutorial sebelumnya, ikon kura-kura tidak akan bisa menabrak danger zone atau daerah pinggir dari terminal dan ikon kura-kura tersebut akan otomatis membelokkan arah dan juga bisa memberikan warna pada path si ikon kura-kura tersebut. Namun untuk video tutorial kali ini terminal akan dibagi menjadi 2 zona yaitu merah dan hijau sudah di atur pada Visual Studio Code dengan kode sebagai berikut.

```
If pose.x > 5.5 and self.previous_x_ <= 5.5;  
Self.get_logger().info("set color to red!")  
Elif pose.x <= 5.5 and self.previous_x_ > 5.5:  
Self.get_logger().info("set color to green!")
```

Jika ikon kura-kura masuk ke zona hijau maka akan menghasilkan output "Set color to green", dan jika ikon kura-kura masuk ke zona merah maka akan menghasilkan output "Set color to red" dilihat dari output yang ada pada gambar dibawah.

