

TECHNICAL REPORT : UTS ROBOTIC_ CREATE A CLOSED LOOP SYSTEM WITH A PUBLISHER AND A SUBSCRIBER ROS2 IN WSL

Nama : Indra Andriansyah Dody Misnadin

NIM : 1103200005

KELAS : TK-44-G7

Technical Report UTS Robotika

Create a Closed Loop System with a Publisher and a Subscriber ROS2 in WSL

1. ROS 2

ROS2 adalah sistem operasi robotik open-source yang dirancang untuk membuat pengembangan robotik lebih mudah, lebih efisien, dan lebih modular. ROS2 dibangun di atas ROS, sistem operasi robotik open-source yang populer, tetapi menawarkan sejumlah fitur baru dan peningkatan yang menjadikannya pilihan yang lebih baik untuk pengembangan robotik modern.

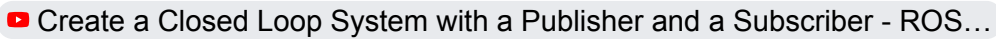
2. Subscriber Nodes

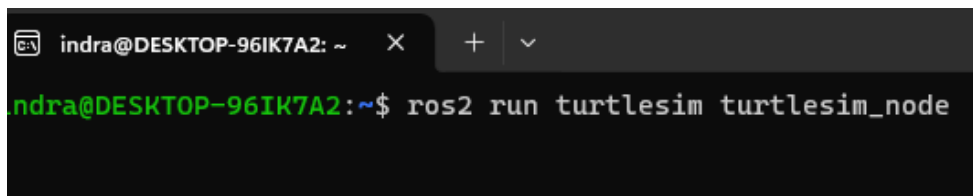
Subscriber nodes adalah jenis node/simpul ROS2 yang mendaftar untuk menerima data dari simpul lain. Subscriber nodes dapat digunakan untuk menerima data dari berbagai sumber, termasuk sensor, aktuator, atau simpul lain

3. Publisher Nodes

Publisher Nodes adalah komponen fundamental dari sistem komunikasi ROS2. Mereka bertanggung jawab untuk menerbitkan data ke topik tertentu, menjadikannya tersedia untuk simpul pelanggan mana pun yang tertarik untuk menerima data tersebut.

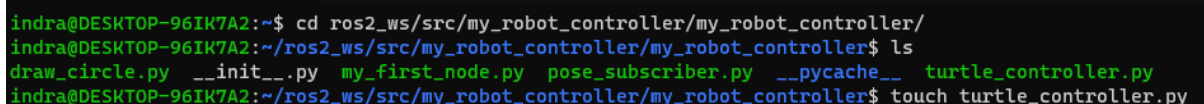
4. Langkah-langkah pengerjaan closed loop system with a publisher and a subscriber

1. Menyiapkan Virtual Machine/WSL/OS menggunakan OS Ubuntu 22.04 ataupun 20.04 (Jammy Jellyfish).
2. ikuti tutorial yang telah diberikan pada week 2 video 9 :

3. pastikan turtlesim dapat berjalan lancar dengan mengetik di terminal “**ros2 run turtlesim turtlesim_node**”



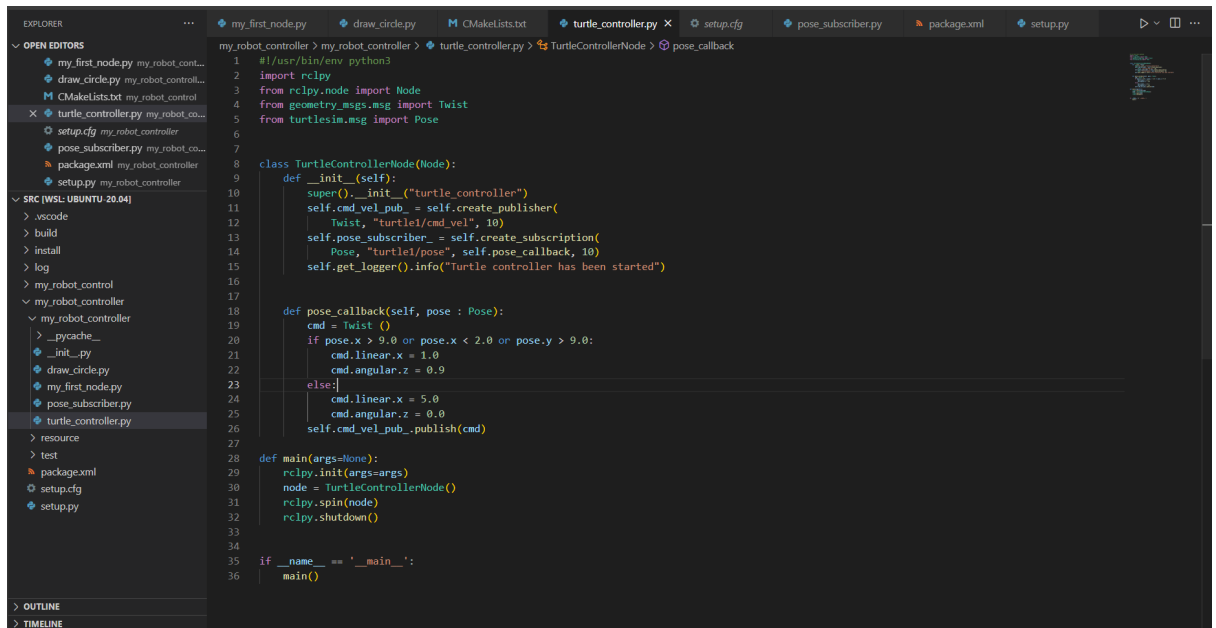
```
indra@DESKTOP-96IK7A2: ~  
indra@DESKTOP-96IK7A2:~$ ros2 run turtlesim turtlesim_node
```

4. buka “**rqt_graph**” untuk melihat isi topik pada turtlesim
5. buat file python untuk mengendalikan gerakan turtle dengan cara:



```
indra@DESKTOP-96IK7A2:~$ cd ros2_ws/src/my_robot_controller/my_robot_controller/  
indra@DESKTOP-96IK7A2:~/ros2_ws/src/my_robot_controller/my_robot_controller$ ls  
draw_circle.py  __init__.py  my_first_node.py  pose_subscriber.py  __pycache__  turtle_controller.py  
indra@DESKTOP-96IK7A2:~/ros2_ws/src/my_robot_controller/my_robot_controller$ touch turtle_controller.py
```

6. kemudian ketik “**cmhd +x turtle_controller.py**” untuk mengaktifkan file
7. lalu buka Visual Studio



```
1 #!/usr/bin/env python3
2 import rclpy
3 from rclpy.node import Node
4 from geometry_msgs.msg import Twist
5 from turtlesim.msg import Pose
6
7
8 class TurtleControllerNode(Node):
9     def __init__(self):
10         super().__init__("turtle_controller")
11         self.cmd_vel_pub_ = self.create_publisher(
12             Twist, "turtle1/cmd_vel", 10)
13         self.pose_subscriber_ = self.create_subscription(
14             Pose, "turtle1/pose", self.pose_callback, 10)
15         self.get_logger().info("Turtle controller has been started")
16
17     def pose_callback(self, pose : Pose):
18         cmd = Twist ()
19         if pose.x > 9.0 or pose.x < 2.0 or pose.y > 9.0:
20             cmd.linear.x = 1.0
21             cmd.angular.z = 0.9
22         else:
23             cmd.linear.x = 5.0
24             cmd.angular.z = 0.0
25         self.cmd_vel_pub_.publish(cmd)
26
27
28 def main(args=None):
29     rclpy.init(args=args)
30     node = TurtleControllerNode()
31     rclpy.spin(node)
32     rclpy.shutdown()
33
34
35 if __name__ == '__main__':
36     main()
```

8. Ikuti arahan sesuai dengan tutorial pada video



```
1 #!/usr/bin/env python3
2 import rclpy
3 from rclpy.node import Node
4 from geometry_msgs.msg import Twist
5 from turtlesim.msg import Pose
6
7
8 class TurtleControllerNode(Node):
9     def __init__(self):
10         super().__init__("turtle_controller")
11         self.cmd_vel_pub_ = self.create_publisher(
12             Twist, "turtle1/cmd_vel", 10)
13         self.pose_subscriber_ = self.create_subscription(
14             Pose, "turtle1/pose", self.pose_callback, 10)
15         self.get_logger().info("Turtle controller has been started")
16
17     def pose_callback(self, pose : Pose):
18         cmd = Twist ()
19         if pose.x > 9.0 or pose.x < 2.0 or pose.y > 9.0:
20             cmd.linear.x = 1.0
21             cmd.angular.z = 0.9
22         else:
23             cmd.linear.x = 5.0
24             cmd.angular.z = 0.0
25         self.cmd_vel_pub_.publish(cmd)
26
27
28 def main(args=None):
29     rclpy.init(args=args)
30     node = TurtleControllerNode()
31     rclpy.spin(node)
32     rclpy.shutdown()
33
34
35 if __name__ == '__main__':
36     main()
```

9. set code pada setup.py

```
entry_points={
    'console_scripts': [
        "test_node = my_robot_controller.my_first_node:main",
        "draw_circle = my_robot_controller.draw_circle:main",
        "pose_subscriber = my_robot_controller.pose_subscriber:main",
        "turtle_controller = my_robot_controller.turtle_controller:main"
    ]
}
```

10. lalu save

11. kembali ke terminal

```
indra@DESKTOP-96IK7A2: ~/ros2_ws/src$ code .
indra@DESKTOP-96IK7A2: ~/ros2_ws/src$ colcon build --symlink-install
indra@DESKTOP-96IK7A2: ~/ros2_ws/src$ cd ..
indra@DESKTOP-96IK7A2: ~$ source .bashrc
indra@DESKTOP-96IK7A2: ~$ ros2 run my_robot_controller turtle_controller
```

12. Buka terminal 2

```
indra@DESKTOP-96IK7A2: ~$ ros2 run turtlesim turtlesim_node
```

13. buka terminal 3

```
indra@DESKTOP-96IK7A2: ~$ ros2 run my_robot_controller pose_subscriber
```

14. Maka Output dari ketiga terminal seperti ini:

The screenshot displays three terminal windows and the VS Code editor. The top-left terminal shows the output of `ros2 run my_robot_controller turtle_controller`, which includes a summary of package builds and a list of turtle controller logs. The top-right terminal shows the output of `ros2 run turtlesim turtlesim_node`, displaying a series of "Oh no! I hit the wall!" messages from the turtlesim node. The bottom-left terminal shows the output of `ros2 run my_robot_controller pose_subscriber`, displaying a series of pose subscriber logs. The bottom-right window shows the VS Code editor with the `setup.py` file open, showing the `entry_points` section.

```
Starting >>> my_robot_controller
/home/indra/.local/lib/python3.8/site-packages/setuptools/dist.py:717: UserWarning: Usage of dash-separated 'script-dir' will not be supported in future versions. Please use the underscore name 'script_dir' instead
  warnings.warn(
/home/indra/.local/lib/python3.8/site-packages/setuptools/dist.py:717: UserWarning: Usage of dash-separated 'install_scripts' will not be supported in future versions. Please use the underscore name 'install_scripts' instead
  warnings.warn(
Finished <<< my_robot_controller [1.95s]
--- stderr: my_robot_controller
/home/indra/.local/lib/python3.8/site-packages/setuptools/dist.py:717: UserWarning: Usage of dash-separated 'script-dir' will not be supported in future versions. Please use the underscore name 'script_dir' instead
  warnings.warn(
---
Finished <<< my_robot_controller [2.68s]
Summary: 2 packages finished [5.75s]
1 package had stderr output: my_robot_controller
indra@DESKTOP-96IK7A2: ~$ source .bashrc
indra@DESKTOP-96IK7A2: ~$ ros2 run my_robot_controller turtle_controller
[INFO] [1780267266.058634196] [turtle_controller]: Turtle controller has been started

[INFO] [1780269701.638591083] [pose_subscriber]: (1.3393453359683882, 0.0)
[INFO] [1780269701.654623683] [pose_subscriber]: (1.3416582475738525, 0.0)
[INFO] [1780269701.678365984] [pose_subscriber]: (1.3481828489383589, 0.0)
[INFO] [1780269701.686888784] [pose_subscriber]: (1.3489426633927869, 0.0)
[INFO] [1780269701.781633785] [pose_subscriber]: (1.3499222135238847, 0.0)
[INFO] [1780269701.717741585] [pose_subscriber]: (1.3531416654588792, 0.0)
[INFO] [1780269701.734995686] [pose_subscriber]: (1.3565795421688342, 0.0)
[INFO] [1780269701.749683686] [pose_subscriber]: (1.3682421232372124, 0.0)
[INFO] [1780269701.765893597] [pose_subscriber]: (1.368128589638127, 0.0)
[INFO] [1780269701.781466988] [pose_subscriber]: (1.368238891468811, 0.0)
[INFO] [1780269701.797812588] [pose_subscriber]: (1.3726699186325873, 0.0)
[INFO] [1780269701.814946489] [pose_subscriber]: (1.377122992376899, 0.0)
[INFO] [1780269701.838154489] [pose_subscriber]: (1.3818964958199918, 0.0)
[INFO] [1780269701.846858618] [pose_subscriber]: (1.3868894577826367, 0.0)
[INFO] [1780269701.861217418] [pose_subscriber]: (1.3921886917953491, 0.0)
[INFO] [1780269701.877588831] [pose_subscriber]: (1.3976292444229126, 0.0)
[INFO] [1780269701.893962311] [pose_subscriber]: (1.4031744427817212, 0.0)
[INFO] [1780269701.909781812] [pose_subscriber]: (1.40898337753295898, 0.0)
[INFO] [1780269701.925731812] [pose_subscriber]: (1.415187258213623, 0.0)
[INFO] [1780269701.942829213] [pose_subscriber]: (1.4213932732689253, 0.0)
[INFO] [1780269701.957274914] [pose_subscriber]: (1.42789853916293115, 0.0)
[INFO] [1780269701.973827414] [pose_subscriber]: (1.4345976114273871, 0.0)
[INFO] [1780269701.998489715] [pose_subscriber]: (1.441513188732727, 0.0)
[INFO] [1780269702.085656415] [pose_subscriber]: (1.4486338165748967, 0.0)

[WARN] [1780269845.748856352] [turtlesim]: Oh no! I hit the wall! (Clamping from [x=6.491738, y=11.125
[WARN] [1780269845.764167152] [turtlesim]: Oh no! I hit the wall! (Clamping from [x=6.563835, y=11.125
[WARN] [1780269845.781081152] [turtlesim]: Oh no! I hit the wall! (Clamping from [x=6.634331, y=11.125
[WARN] [1780269845.796346452] [turtlesim]: Oh no! I hit the wall! (Clamping from [x=6.785627, y=11.125
[WARN] [1780269845.812264152] [turtlesim]: Oh no! I hit the wall! (Clamping from [x=6.776924, y=11.125
[WARN] [1780269845.828713151] [turtlesim]: Oh no! I hit the wall! (Clamping from [x=6.848228, y=11.125
[WARN] [1780269845.844418851] [turtlesim]: Oh no! I hit the wall! (Clamping from [x=6.919516, y=11.125
[WARN] [1780269845.860875751] [turtlesim]: Oh no! I hit the wall! (Clamping from [x=6.998813, y=11.125
[WARN] [1780269845.876848351] [turtlesim]: Oh no! I hit the wall! (Clamping from [x=7.062189, y=11.125
[WARN] [1780269845.892572451] [turtlesim]: Oh no! I hit the wall! (Clamping from [x=7.133486, y=11.125
[WARN] [1780269845.908416651] [turtlesim]: Oh no! I hit the wall! (Clamping from [x=7.284782, y=11.125
[WARN] [1780269845.924341151] [turtlesim]: Oh no! I hit the wall! (Clamping from [x=7.358189, y=11.125

name: package_name,
version: '0.0.0',
packages=[package_name],
data_files=[
    ('share/ament_index/resource_index/packages',
     ['resource/' + package_name]),
    ('share/' + package_name, ['package.xml']),
],
install_requires=['setuptools'],
zip_safe=True,
maintainer_email='indra@todo.todo',
description='TODO: Package description',
license='TODO: License declaration',
tests_requires=['pytest'],
entry_points={
    'console_scripts': [
        "test_node = my_robot_controller.my_first
        "draw_circle = my_robot_controller.draw_ci
        "pose_subscriber = my_robot_controller.pose
        "turtle_controller = my_robot_controller.tur
```

15. Output dari turtlesim:

