

Dokumentasi Lecture 5

Playlist Video:

https://www.youtube.com/watch?v=mUmOwr-U_68&list=PLt69C9MnPhkP0ZXZOqmlGRT0ch8o9GiQ&index=6

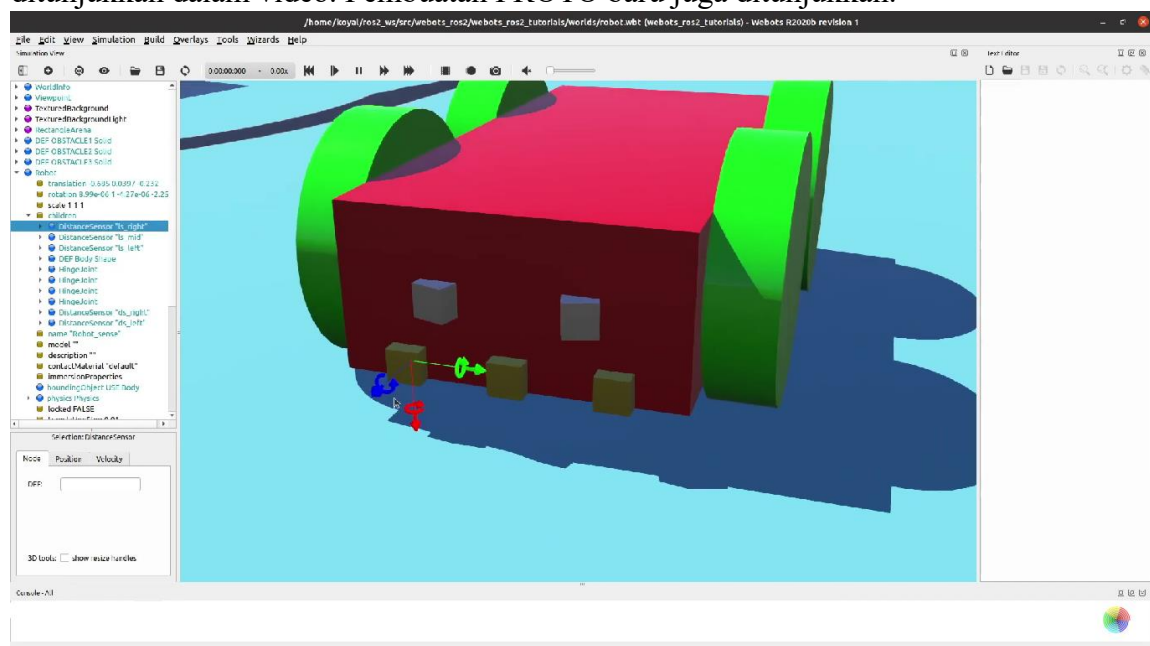
Video 6 - Line Following Custom Robot Project | Webots ROS2 project Tutorial | [Tutorial 6]

Video keenam ini mengimplementasikan konsep master dan slave di mana master *publishes* pesan pada suatu topik, dan slave *subscribes* pada mereka. Di sini, robot khusus dibuat untuk mengikuti garis. Tiga sensor di depan robot mendeteksi warna di bawahnya, dengan menggunakan informasi ini, kita dapat mengetahui perbedaan antara warna hitam dan biru dan memastikan bahwa robot mengikuti warna hitam. Dengan menggunakan `rqt_graph` (visualisasi node dan topik), dapat terlihat bahwa node master *subscribes* topik sensor dan melakukan perhitungan, dan *publishes* ke topik `/cmd_vel` *subscribed* oleh node slave. Di sini, node slave berperan sebagai jembatan antara Webots dan ROS2, dan node master adalah otak yang ditulis dalam ROS2.

Video 6 terbagi dalam beberapa sub, antara lain:

- *Edit custom robot in webots*

Pembuatan Robot Kustom dan Integrasi Sensor Jarak pada Robot. Tiga sensor jarak tambahan ditambahkan mengarah ke bawah. Penetapan posisi dan orientasi ini ditunjukkan dalam video. Pembuatan PROTO baru juga ditunjukkan.



- *Overview of slave*

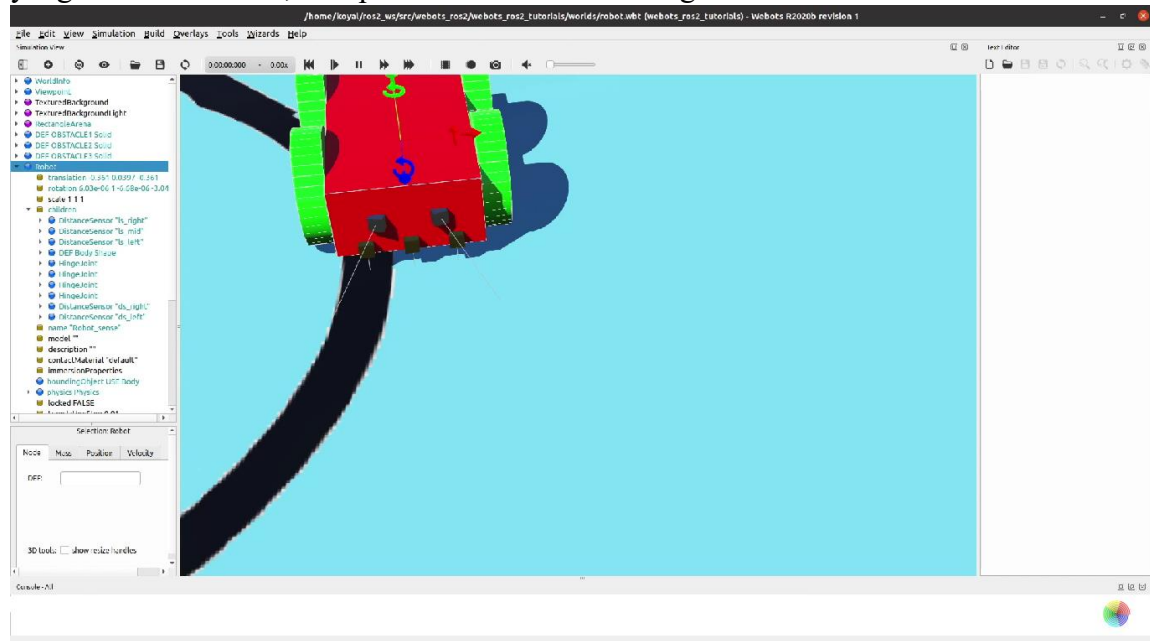
Node untuk berinteraksi dengan Webots. Metode ditunjukkan untuk menulis pelanggan ROS2 ke topik `/cmd_vel`. Nilai sensor kanan-kiri dan tengah dipublikasikan di sini untuk digunakan oleh node master. Sebuah timer digunakan untuk menentukan frekuensi penerbitan ini. Pastikan nama sensor di Webots sesuai dengan nama yang digunakan dengan layanan Webots. Rotasi roda robot kiri dan kanan juga ditangani di sini menggunakan layanan untuk mengatur posisi dan kecepatan roda berdasarkan nilai `/cmd_vel`. Perhitungan dilakukan menggunakan dinamika penggerak roda diferensial. Pemangkas kecepatan juga dilakukan di sini.

[illegible][illegible]

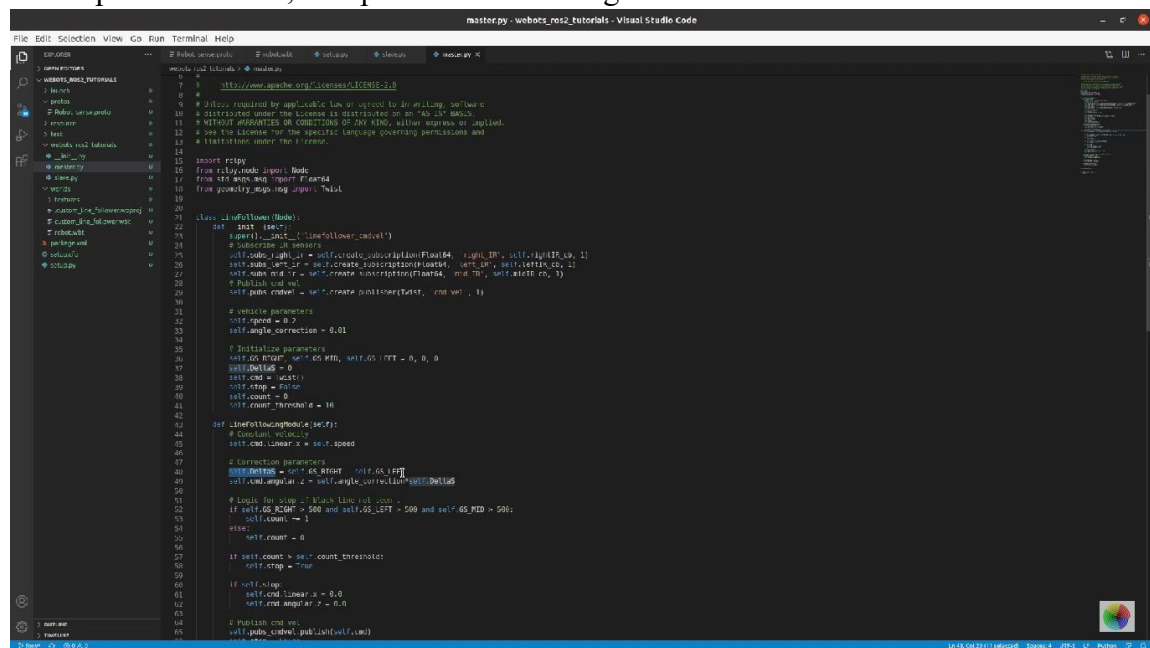
- *Overview of master*
Node ini berlangganan pada pesan sensor yang dipublikasikan oleh node slave dan membuat perkiraan arah serta membuat penerbit ROS2 ke /cmd_vel dalam bentuk pesan Twist.

[illegible]

Properti penerangan digunakan di sini. Warna yang berbeda memiliki properti penerangan yang berbeda. Di sini, kecepatan linear diambil sebagai 0.2.



Jika ada perbedaan nilai antara sensor kiri dan kanan, itu berarti bahwa robot bergerak ke arah yang salah karena satu sensor melihat warna hitam. Oleh karena itu, kita mengalikan perbedaan ini dengan variabel koreksi sudut dan mengarahkan robot. Jika sensor tengah tidak melihat garis hitam, logika ditulis untuk menghentikan robot setelah menunggu beberapa saat. Di sini, kecepatan linear dan angular diberikan nilai 0.



- *Setup project*

Struktur file peluncuran di ROS2 dijelaskan di sini yang mencakup paket dan direktori inti. Ini akan memiliki file peluncuran robot. Ini juga akan memiliki nama paket dan eksekutor.

The screenshot shows the Visual Studio Code editor with the file `~/ros2_ws/src/webots/webots_ros2_tutorials/webots_ros2_tutorials/setup.py` open. The editor displays the following Python code:

```
1 from glob import glob
2 from setuptools import setup
3
4 package_name = 'webots_ros2_tutorials'
5
6 data_files = []
7 data_files.append(share/install_index/resource_index/packages, [
8     'resources/1 package name'
9 ])
10 data_files.append(share/ + package_name, [
11     'resources/1e/launch/launch.py'
12 ])
13 data_files.append(share/ + package_name + '/urdf', [
14     'urdf/custom_line_follow.urdf'
15 ])
16 data_files.append(share/ + package_name + '/urdf', [
17     'urdf/robot_base.urdf'
18 ])
19
20 entry_files.append(share/ + package_name + '/gazebo/rooms', glob('rooms/*.urdf'))
21 data_files.append(share/ + package_name + '/gazebo/rooms', glob('rooms/*.urdf'))
22 data_files.append(share/ + package_name + '/gazebo/textures', glob('textures/*.png'))
23
24 data_files.append(share/ + package_name, [
25     'package.xml'
26 ])
27
28 setup(
29     name=package_name,
30     version='0.0.1',
31     packages=[package_name],
32     data_files=data_files,
33     install_requires=['setuptools', 'launch'],
34     zip_safe=True,
35     keywords=['ROS2', 'webots', 'Swift_Illusion', 'Tutorials', 'Webots', 'Simulation'],
36     maintainer_email='Harsh B. Khandelwal@phdmail.in',
37     description='Projects for videos for webots ros2 tutorial series on youtube.',
38     license='Apache License, Version 2.0',
39     tests_require=['pytest'],
40
41     entry_points={
42         'console_scripts': [
43             'web_bot = webots_ros2_tutorials.web_bot:main',
44             'web_bot_sim = webots_ros2_tutorials.web_bot_sim:main',
45             'line_follower = webots_ros2_tutorials.line_follower:main'
46         ],
47         'launch.frontend.launch_extensions': ['launch_ros = launch_ros']
48     }
49 )
50
```

- *Build and demo*
colcon build dari proyek ini ditunjukkan di sini.

The screenshot shows the Visual Studio Code interface with the `setup.py` file for `webots_ros2_tutorials` open in the editor. The file contains metadata for the package, including its name, version, maintainer, and entry points. A terminal window in the bottom right corner shows the output of a `colcon build` command, indicating that the build was successful and the packages were installed.

```

1 from glob import glob
2 from setuptools import setup
3
4 package_name = 'webots_ros2_tutorials'
5
6 data_files = []
7 data_files.append([share/install_index/resource_index/packages], [
8     'resource/' + package_name
9 ])
10 data_files.append([share/ + package_name, [
11     'launch/line_following.launch.py'
12 ]])
13 data_files.append([share/ + package_name + '/worlds', [
14     'worlds/custom_line_follower.world'
15 ]])
16 data_files.append([share/ + package_name + '/protos', [
17     'protos/RobotSense.proto'
18 ]])
19
20 data_files.append([share/ + package_name + '/protos/icons', glob('protos/icons/*')])
21 data_files.append([share/ + package_name + '/worlds/textures', glob('worlds/textures/*')])
22 data_files.append([share/ + package_name + '/protos/textures', glob('protos/textures/*')])
23
24 data_files.append([share/ + package_name, [
25     'package.xml'
26 ]])
27
28 setup(
29     name=package_name,
30     version='0.0.1',
31     packages=[package_name],
32     data_files=data_files,
33     install_requires=['setuptools', 'launch'],
34     zip_safe=True,
35     keywords='ROS2', 'Webots', 'Soft-Illusion', 'Tutorials', 'Youtube', 'Simulation',
36     maintainer='Soft-Illusion',
37     maintainer_email='hurns.kakashi@gmail.com',
38     description='Projects for videos for webots ros2 tutorial series on youtube.',
39     license='Apache License, Version 2.0',
40     tests_require=['pytest'],
41     entry_points={
42         'console_scripts': [
43             'sub_tof = webots_ros2_tutorials.sub_tof:main',
44             'line_follower = webots_ros2_tutorials.line_follower:main',
45             'line_follower = webots_ros2_tutorials.master:main'
46         ],
47         'launch.frontend.launch_extensions': ['launch_ros = launch_ros']
48     }
49 )
50

```

```

colcon build [4/12 done] [6 ongoing]
koyal@soft-illusion-pci:~/ros2_ws/src/webots_ros2/webots_ros2_tutorials/worlds$ w
ebots custom_line_follower.wbt
^Ckoyal@soft-illusion-pci:~/ros2_ws/src/webots_ros2/webots_ros2_tutorials/worlds$
d --/..
koyal@soft-illusion-pci:~/ros2_ws/src/webots_ros2$ cd ..
koyal@soft-illusion-pci:~/ros2_ws/src$ cd ..
koyal@soft-illusion-pci:~/ros2_ws$ colcon build
colcon: command not found
koyal@soft-illusion-pci:~/ros2_ws$ colcon build
Starting >>> webots_ros2_msgs
Starting >>> webots_ros2_ur_e_description
Starting >>> webots_ros2_importer
Finished <<< webots_ros2_msgs [0.79s]
Starting >>> webots_ros2_core
Finished <<< webots_ros2_importer [1.04s]
Finished <<< webots_ros2_ur_e_description [1.05s]
Finished <<< webots_ros2_core [0.56s]
Starting >>> webots_ros2_abb
Starting >>> webots_ros2_universal_robot
Starting >>> webots_ros2_epuck
Starting >>> webots_ros2_examples
Starting >>> webots_ros2_tiago
Starting >>> webots_ros2_tutorials
[2.4s] [4/12 complete] [6 ongoing] [webots_ros2_abb - 0.9s] ...

```

Proyek ini kemudian diluncurkan menggunakan: `ros2 launch webots_ros2_tutorial line_following.launch` ros2 topic list akan menunjukkan daftar semua topik. Setiap topik dapat di-echo menggunakan `ros2 topic echo /nama_topik`.

The screenshot shows the Visual Studio Code interface with the `setup.py` file for `webots_ros2_tutorials` open in the editor. The file contains metadata for the package, including its name, version, maintainer, and entry points. A terminal window in the bottom right corner shows the output of a `ros2 launch` command, indicating that the launch was successful and the packages were installed.

```

1 from glob import glob
2 from setuptools import setup
3
4 package_name = 'webots_ros2_tutorials'
5
6 data_files = []
7 data_files.append([share/install_index/resource_index/packages], [
8     'resource/' + package_name
9 ])
10 data_files.append([share/ + package_name, [
11     'launch/line_following.launch.py'
12 ]])
13 data_files.append([share/ + package_name + '/worlds', [
14     'worlds/custom_line_follower.world'
15 ]])
16 data_files.append([share/ + package_name + '/protos', [
17     'protos/RobotSense.proto'
18 ]])
19
20 data_files.append([share/ + package_name + '/protos/icons', glob('protos/icons/*')])
21 data_files.append([share/ + package_name + '/worlds/textures', glob('worlds/textures/*')])
22 data_files.append([share/ + package_name + '/protos/textures', glob('protos/textures/*')])
23
24 data_files.append([share/ + package_name, [
25     'package.xml'
26 ]])
27
28 setup(
29     name=package_name,
30     version='0.0.1',
31     packages=[package_name],
32     data_files=data_files,
33     install_requires=['setuptools', 'launch'],
34     zip_safe=True,
35     keywords='ROS2', 'Webots', 'Soft-Illusion', 'Tutorials', 'Youtube', 'Simulation',
36     maintainer='Soft-Illusion',
37     maintainer_email='hurns.kakashi@gmail.com',
38     description='Projects for videos for webots ros2 tutorial series on youtube.',
39     license='Apache License, Version 2.0',
40     tests_require=['pytest'],
41     entry_points={
42         'console_scripts': [
43             'sub_tof = webots_ros2_tutorials.sub_tof:main',
44             'line_follower = webots_ros2_tutorials.line_follower:main',
45             'line_follower = webots_ros2_tutorials.master:main'
46         ],
47         'launch.frontend.launch_extensions': ['launch_ros = launch_ros']
48     }
49 )
50

```

```

koyal@soft-illusion-pci:~/ros2_ws$
Finished <<< webots_ros2_abb [1.89s]
Starting >>> webots_ros2_demo
Finished <<< webots_ros2_examples [2.09s]
Finished <<< webots_ros2_demo [0.54s]
Starting >>> webots_ros2
Finished <<< webots_ros2 [0.56s]
Summary: 12 packages finished [4.47s]
koyal@soft-illusion-pci:~/ros2_ws$ ros2 launch webots_ros2_tutorials line_followi
ng launch.py
[INFO] [launch]: All log files can be found below /home/koyal/.ros/log/2020-10-3
2-18-18-02-127789-soft-illusion-pci-44000
[INFO] [launch]: Default logging verbosity is set to INFO
[WARNING] [launch_ros.actions.node]: Parameter file path is not a file: /dev/nul
l
[INFO] [robot_state_publisher-1]: process started with pid [40897]
[INFO] [custom_line_follower.wbt --node realtime-2]: process started with pid [4
0899]
[INFO] [enable_robot-3]: process started with pid [40817]
[INFO] [line_follower-4]: process started with pid [40614]
[robot_state_publisher-1] parsing robot urdf xml string.
[robot_state_publisher-1] [INFO] [1004180205.200240920] [robot_state_publisher]:
got segment

```