

NAMA : Albizhar Zidane Budi Laksana
NIM : 1103202116
KELAS : TK-44-G7

“Lecture Week 5”

REPORT WEEK 5 Video 6

Pada minggu pembelajaran ke-3, saya mengalami kesulitan dalam menghadapi error yang tidak bisa terinstallnya `sudo apt-get install ros-foxy-webots-ros2` dalam integrasi antara ros2 dan webots. Dikarenakan pada video 1 saya sudah mengalami error sehingga saya tidak bisa melakukan progres pada video 6.

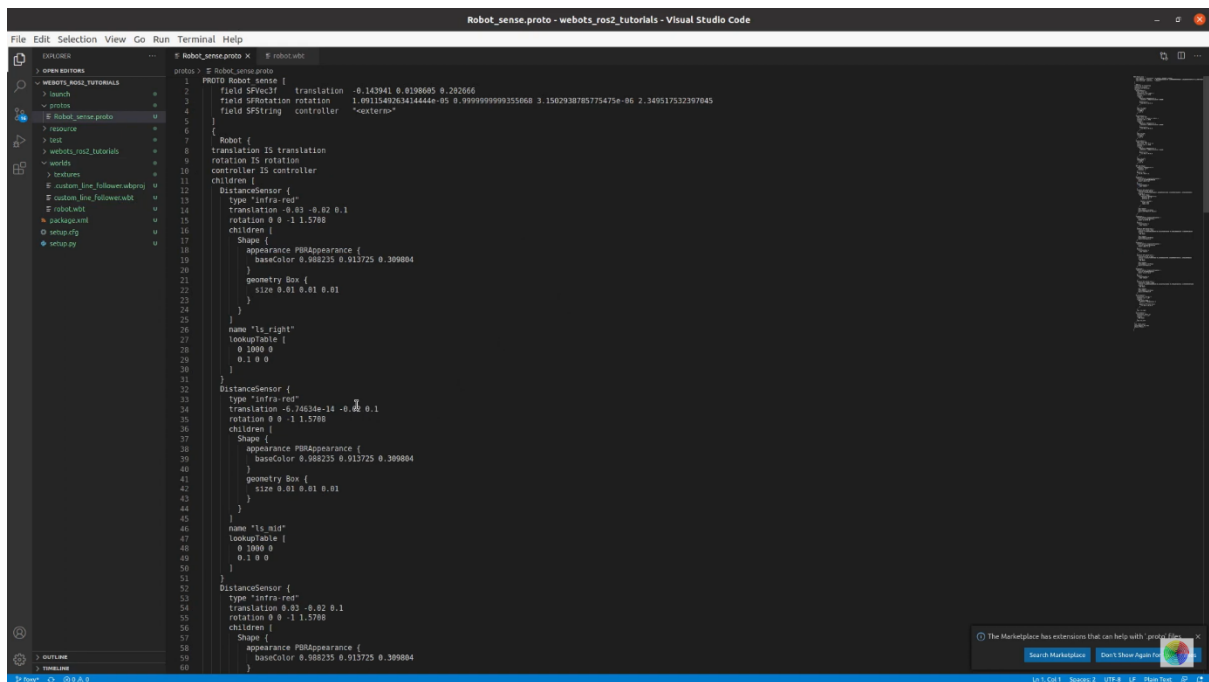
Video 6

https://www.youtube.com/watch?v=ZTJa5f5F5fU&list=PLt69C9MnPchkP0ZXZOqmIGRTOch8o9GiQ&index=7&ab_channel=Softillusion

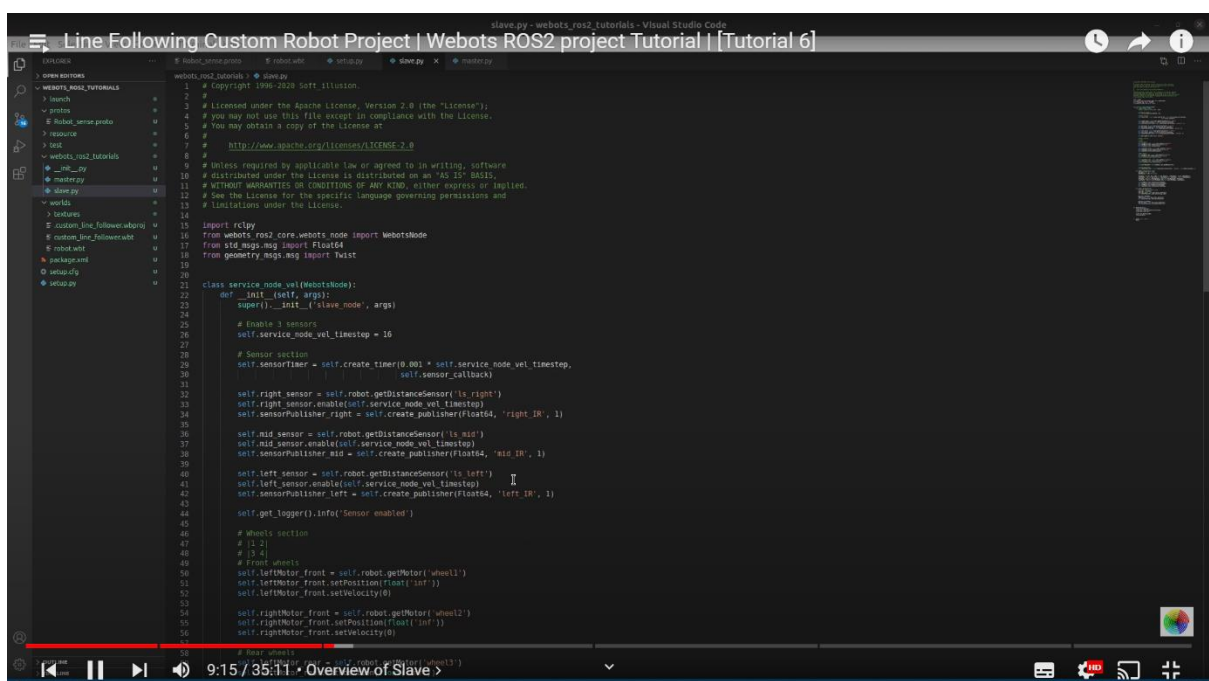


Pada video ke-6 week 4 pembelajaran robotika kali ini membahas mengenai project sederhana bagaimana cara pengoperasian Line Following custom Robot Project pada ubuntu.

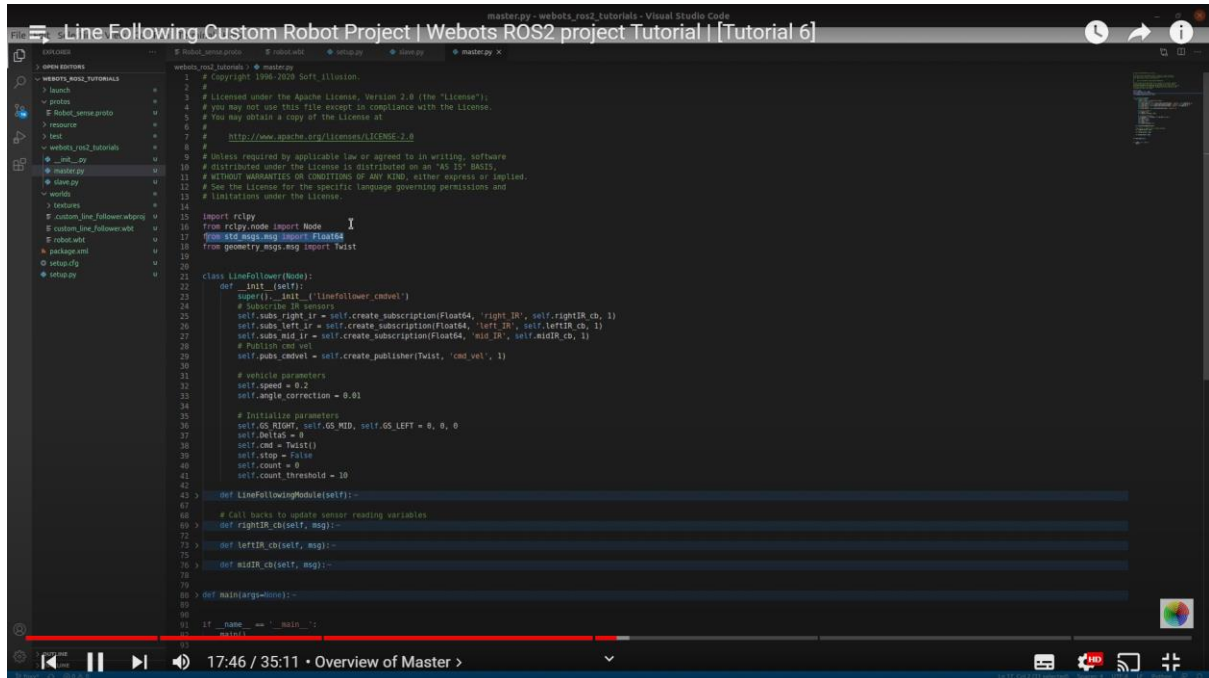
1. Langkah pertama dalam kode ini adalah mendeklarasikan file `line_following_launch.py` sebagai kode utama dalam proyek ini. Tujuan dari kode ini adalah untuk menentukan lokasi sensor, sedangkan pada file `robot.twb` juga terdapat berbagai sensor yang akan dimanfaatkan dalam pembuatan robot.



- Langkah berikutnya melibatkan deklarasi file slave.py agar bisa berinteraksi dengan robot. Di dalam slave.py, dapat dilakukan penyesuaian terhadap konfigurasi roda dan sensor pada robot, serta penyesuaian jarak sensor terhadap hambatan sesuai kebutuhan yang diinginkan.

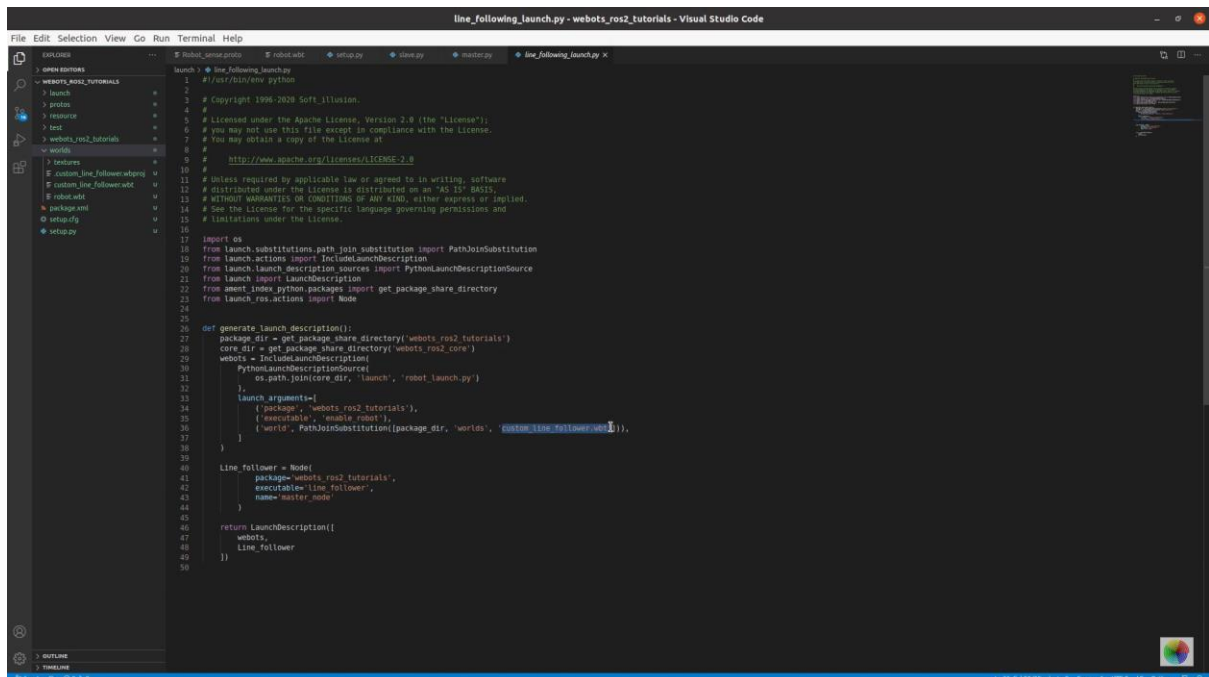


3. Selanjutnya ada berkas node master.py yang bertugas untuk mengevaluasi data dari slave.py dengan mengontrol arah gerakan robot kita, sehingga menghasilkan keluaran data yang diinginkan.



```
1 # Copyright 1996-2020 Soft Illusion.
2 #
3 # Licensed under the Apache License, Version 2.0 (the "License");
4 # you may not use this file except in compliance with the License.
5 # You may obtain a copy of the License at
6 # http://www.apache.org/licenses/LICENSE-2.0
7 #
8 # Unless required by applicable law or agreed to in writing, software
9 # distributed under the License is distributed on an "AS IS" BASIS,
10 # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
11 # See the License for the specific language governing permissions and
12 # limitations under the License.
13
14 import rclpy
15 from rclpy.node import Node
16 from std_msgs.msg import Float64
17 from geometry_msgs.msg import Twist
18
19 class LineFollower(Node):
20     def __init__(self):
21         super().__init__('line_follower_cmsvel')
22         # Subscribe IR sensors
23         self.sub_right_ir = self.create_subscription(Float64, 'right_IR', self.rightIR_cb, 1)
24         self.sub_left_ir = self.create_subscription(Float64, 'left_IR', self.leftIR_cb, 1)
25         self.sub_mid_ir = self.create_subscription(Float64, 'mid_IR', self.midIR_cb, 1)
26         # Publish cmd vel
27         self.pub_cmdvel = self.create_publisher(Twist, 'cmd_vel', 1)
28
29         # vehicle parameters
30         self.speed = 0.2
31         self.angle_correction = 0.01
32
33         # Initialize parameters
34         self.OS_RIGHT = self.OS_MID, self.OS_LEFT = 0, 0, 0
35         self.DeltaS = 0
36         self.cmd = Twist()
37         self.stop = False
38         self.count = 0
39         self.count_threshold = 10
40
41     def lineFollowingModule(self):
42         # Call back to update sensor reading variables
43         def rightIR_cb(self, msg):
44             self.OS_RIGHT = msg.data
45         def leftIR_cb(self, msg):
46             self.OS_LEFT = msg.data
47         def midIR_cb(self, msg):
48             self.OS_MID = msg.data
49
50     def main(self):
51         self.lineFollowingModule()
52         while True:
53             # Logic for line following
54             # ...
55             self.cmd = Twist()
56             self.pub_cmdvel.publish(self.cmd)
57             rclpy.spin_once(self, timeout_sec=0.1)
58
59 if __name__ == '__main__':
60     main()
```

4. Setelah mendapatkan data tersebut, kita kembali ke dalam file node line_following_launch.py yang bertugas untuk menyimpan direktori paket yang sudah dibuat pada langkah-langkah sebelumnya. Setelah itu, kita melakukan integrasi antara file line following ini dengan setup.py untuk memperkuat fungsionalitasnya.



```
1 # Copyright 1996-2020 Soft Illusion.
2 #
3 # Licensed under the Apache License, Version 2.0 (the "License");
4 # you may not use this file except in compliance with the License.
5 # You may obtain a copy of the License at
6 # http://www.apache.org/licenses/LICENSE-2.0
7 #
8 # Unless required by applicable law or agreed to in writing, software
9 # distributed under the License is distributed on an "AS IS" BASIS,
10 # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
11 # See the License for the specific language governing permissions and
12 # limitations under the License.
13
14 import os
15 from launch.actions import IncludeLaunchDescription
16 from launch.launch_description_sources import PythonLaunchDescriptionSource
17 from launch.substitutions import PathJoinSubstitution
18 from launch_ros.actions import Node
19
20 def generate_launch_description():
21     package_dir = get_package_share_directory('webots_ros2_tutorials')
22     core_dir = get_package_share_directory('webots_ros2_core')
23     webots = IncludeLaunchDescription(
24         PythonLaunchDescriptionSource(
25             os.path.join(core_dir, 'launch', 'robot_launch.py')
26         ),
27         launch_arguments=[
28             ('package', 'webots_ros2_tutorials'),
29             ('executable', 'line_follower'),
30             ('world', PathJoinSubstitution([package_dir, 'worlds', 'line_follower.world'])),
31         ],
32     )
33
34     line_follower = Node(
35         package='webots_ros2_tutorials',
36         executable='line_follower',
37         name='master_node'
38     )
39
40     return LaunchDescription([
41         webots,
42         line_follower
43     ])
44
45 if __name__ == '__main__':
46     generate_launch_description()
```

5. Pada tahap akhir, kita melakukan demonstrasi terhadap kode line following di Webots untuk menguji fungsionalitasnya.

