

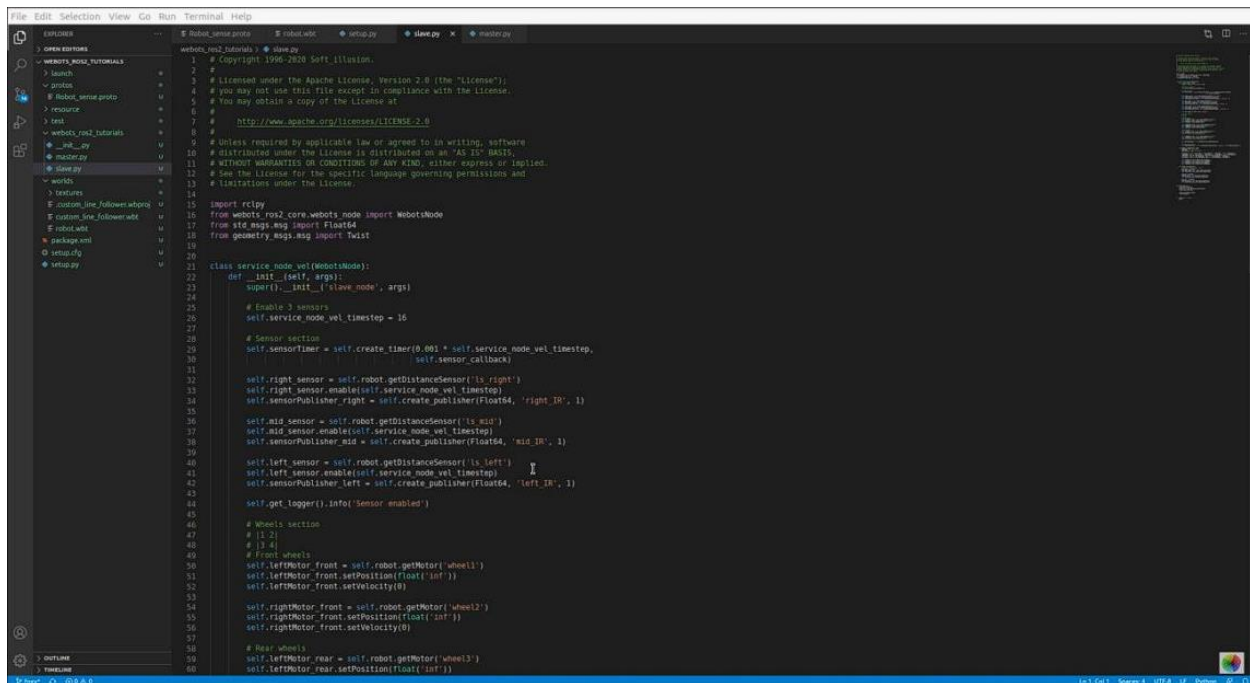
Nama : Naufal Raihan Ramadhan

Kelas : TK44G7

## Lecture 5

Dalam materi pertemuan ke-5, ada rujukan kepada Video 6 yang tampaknya tidak disertai dengan file tutorial yang relevan, sebagaimana disajikan dalam konten video tersebut. Dalam konteks pengembangan, langkah awal yang disarankan adalah melakukan modifikasi manual terhadap model robot melalui platform Webots. Selanjutnya, untuk memfasilitasi integrasi sensor dan parameter tertentu pada robot, dianjurkan untuk membuat skrip berbasis Python dengan nama `slave.py` menggunakan lingkungan pengembangan Visual Studio Code.

Saat mengembangkan skrip ini, penting untuk menjalankan proses dengan kehati-hatian ekstra guna meminimalkan potensi kesalahan. Selain itu, segmentasi arah dan kecepatan rotasi roda robot juga harus ditetapkan dengan teliti, termasuk dalam menetapkan batasan kecepatan yang optimal.



```
File Edit Selection View Go Run Terminal Help
slave.py
1 # Copyright 1996-2020 Soft Licensing.
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 #
7 # http://www.apache.org/licenses/LICENSE-2.0
8 #
9 # Unless required by applicable law or agreed to in writing, software
10 # distributed under the License is distributed on an "AS IS" BASIS,
11 # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
12 # See the License for the specific language governing permissions and
13 # limitations under the License.
14
15 import rclpy
16 from webots_ros2_core.webots_node import WebotsNode
17 from std_msgs.msg import Float64
18 from geometry_msgs.msg import Twist
19
20 class service_node_vel(WebotsNode):
21     def __init__(self, args):
22         super().__init__('slave_node', args)
23
24         # Enable 3 sensors
25         self.service_node_vel.timestep = 16
26
27         # Sensor section
28         self.sensor_timer = self.create_timer(0.001 * self.service_node_vel.timestep,
29                                             self.sensor_callback)
30
31         self.right_sensor = self.robot.getDistanceSensor('ls_right')
32         self.right_sensor.enable(self.service_node_vel.timestep)
33         self.sensor_publisher_right = self.create_publisher(Float64, 'right_IR', 1)
34
35         self.mid_sensor = self.robot.getDistanceSensor('ls_mid')
36         self.mid_sensor.enable(self.service_node_vel.timestep)
37         self.sensor_publisher_mid = self.create_publisher(Float64, 'mid_IR', 1)
38
39         self.left_sensor = self.robot.getDistanceSensor('ls_left')
40         self.left_sensor.enable(self.service_node_vel.timestep)
41         self.sensor_publisher_left = self.create_publisher(Float64, 'left_IR', 1)
42
43         self.get_logger().info('Sensor enabled')
44
45         # Wheels section
46         # [1, 2]
47         # [1, 4]
48         # Front wheels
49         self.leftMotor_front = self.robot.getMotor('wheel1')
50         self.leftMotor_front.setPosition(float('inf'))
51         self.leftMotor_front.setVelocity(0)
52
53         self.rightMotor_front = self.robot.getMotor('wheel2')
54         self.rightMotor_front.setPosition(float('inf'))
55         self.rightMotor_front.setVelocity(0)
56
57         # Rear wheels
58         self.leftMotor_rear = self.robot.getMotor('wheel3')
59         self.leftMotor_rear.setPosition(float('inf'))
60         self.leftMotor_rear.setVelocity(0)
```

Langkah selanjutnya dalam proses pengembangan melibatkan pembuatan file `master.py`. File ini dirancang untuk berlangganan atau 'subscribe' terhadap informasi sensor yang diterbitkan oleh node `slave`. Dalam konteks ini, `master.py` mengembangkan estimasi arah berdasarkan data yang diterima dan meneruskan instruksi gerak melalui pesan Twist ke topik `/cmd_vel` menggunakan kerangka kerja ROS2. Dalam implementasinya, fitur pencahayaan digunakan sebagai indikator, di mana variasi warna menunjukkan respons yang berbeda. Kecepatan linear spesifik ditetapkan pada 0,2.

Apabila terdapat disparitas antara nilai sensor kiri dan kanan — mengindikasikan deteksi warna hitam oleh satu sensor — maka perbedaan ini dikalibrasi dengan variabel sudut koreksi untuk

menyelaraskan orientasi robot. Dalam skenario di mana sensor pusat tidak mendeteksi garis hitam, logika telah diintegrasikan untuk menghentikan gerakan robot setelah durasi tertentu dengan mengatur kecepatan linear dan angular pada nilai nol. Setelah konfigurasi selesai, implementasi dapat dilaksanakan dengan mengaktifkan perintah `colcon build` untuk membangun proyek dan `ros2 launch webots_ros2_tutorial line_following.launch` untuk menjalankannya. Melalui perintah `ros2 topic list`, daftar topik yang aktif dapat ditampilkan. Dengan penyelesaian ini, robot diharapkan mampu mengikuti jalur hitam dengan akurasi dan melakukan koreksi berdasarkan input sensor.

Jika robot keluar dari jalur yang ditentukan, pergerakannya akan dihentikan. Adalah imperatif untuk memastikan keseluruhan proses dilaksanakan dengan akurat, memastikan setiap perintah dan konfigurasi sesuai untuk mencapai hasil yang optimal.