

Nama: Valrama Wardhana Hariwidjaja

NIM: 1103204152

UTS Robotika dan Sistem Cerdas

### LAUNCH WEBOTS TESLA DRIVER

SC:

```
import os
import pathlib
import launch
from launch.substitutions import LaunchConfiguration
from launch.actions import DeclareLaunchArgument
from launch.substitutions.path_join_substitution import
PathJoinSubstitution
from launch import LaunchDescription
from ament_index_python.packages import get_package_share_directory
from launch_ros.actions import Node
from webots_ros2_driver.webots_launcher import WebotsLauncher
from webots_ros2_driver.utils import controller_url_prefix

def get_ros2_nodes(*args):
    package_dir = get_package_share_directory('webots_ros2_tesla')
    robot_description = pathlib.Path(os.path.join(package_dir, 'resource',
'tesla_webots.urdf')).read_text()

    tesla_driver = Node(
        package='webots_ros2_driver',
        executable='driver',
        output='screen',
        additional_env={'WEBOTS_CONTROLLER_URL': controller_url_prefix() +
'vehicle'},
        parameters=[
            {'robot_description': robot_description},
        ]
    )

    lane_follower = Node(
        package='webots_ros2_tesla',
        executable='lane_follower',
    )

    return [
        lane_follower,
        tesla_driver,
    ]

def generate_launch_description():
    package_dir = get_package_share_directory('webots_ros2_tesla')
    world = LaunchConfiguration('world')

    webots = WebotsLauncher(
        world=PathJoinSubstitution([package_dir, 'worlds', world]),
        ros2_supervisor=True
    )

    # The following line is important!
```

```

    # This event handler respawns the ROS 2 nodes on simulation reset
    (supervisor process ends).
    reset_handler = launch.actions.RegisterEventHandler(
        event_handler=launch.event_handlers.OnProcessExit(
            target_action=webots._supervisor,
            on_exit=get_ros2_nodes,
        )
    )

    return LaunchDescription([
        DeclareLaunchArgument(
            'world',
            default_value='tesla_world.wbt',
            description='Choose one of the world files from
`/webots_ros2_tesla/worlds` directory'
        ),
        webots,
        webots._supervisor,

        # This action will kill all nodes once the Webots simulation has
        exited
        launch.actions.RegisterEventHandler(
            event_handler=launch.event_handlers.OnProcessExit(
                target_action=webots,
                on_exit=[
                    launch.actions.UnregisterEventHandler(
                        event_handler=reset_handler.event_handler
                    ),
                ],
            ),
        ),

        launch.actions.EmitEvent(event=launch.events.Shutdown())
    ])

    # Add the reset event handler
    reset_handler
] + get_ros2_nodes())

```

## Pendahuluan:

Webots Tesla Driver adalah paket untuk mengendalikan kendaraan Tesla di simulator robot Webots. Laporan teknis ini akan menguji kode Python yang meluncurkan Webots Tesla Driver menggunakan ROS2. Kode ini meluncurkan lingkungan simulasi Webots, memuat kendaraan Tesla, dan meluncurkan dua node ROS2: pengemudi Tesla dan pengikut jalur.

## Analisis Kode:

Kode dimulai dengan mengimpor modul dan paket yang diperlukan, termasuk `os`, `pathlib`, `launch`, `LaunchConfiguration`, `PathJoinSubstitution`, `LaunchDescription`, `get_package_share_directory`, `Node`, `WebotsLauncher`, dan `controller_url_prefix`.

Fungsi `get_ros2_nodes` mendefinisikan dua node ROS2: **tesla\_driver** dan **lane\_follower**. Node **tesla\_driver** diluncurkan dengan paket **webots\_ros2\_driver**, executable **driver**, dan parameter deskripsi robot. Parameter **additional\_env** digunakan untuk menetapkan variabel lingkungan

**WEBOTS\_CONTROLLER\_URL**, yang menentukan URL pengontrol Webots untuk kendaraan. Node **lane\_follower** diluncurkan dengan paket **webots\_ros2\_tesla** dan executable **lane\_follower**.

Fungsi **generate\_launch\_description** mendefinisikan konfigurasi peluncuran. Kode ini dimulai dengan mendefinisikan argumen **world**, yang menentukan nama file dunia Webots yang akan dimuat. Variabel **webots** digunakan untuk meluncurkan lingkungan simulasi Webots dengan file dunia yang ditentukan. **Parameter ros2\_supervisor** diatur menjadi True, yang mengaktifkan proses supervisor ROS2 Webots. Proses pengawas bertanggung jawab untuk mengelola node ROS2 yang berinteraksi dengan lingkungan simulasi Webots.

Variabel **reset\_handler** mendefinisikan penanganan acara yang memulai ulang node ROS2 ketika simulasi Webots direset. Hal ini dicapai dengan mendaftarkan penanganan acara yang mendengarkan proses supervisor Webots untuk keluar, dan kemudian meluncurkan fungsi **get\_ros2\_nodes** untuk memulai ulang node.

Deskripsi peluncuran didefinisikan dengan menggabungkan daftar tindakan, termasuk argumen **world**, tindakan peluncuran **webots**, tindakan peluncuran **supervisor**, **reset\_handler**, dan fungsi **get\_ros2\_nodes**. **Reset\_handler** digunakan untuk memastikan bahwa node ROS2 dijalankan kembali ketika simulasi Webots direset.

### Kesimpulan:

Laporan teknis ini telah menguji kode Python yang meluncurkan Webots Tesla Driver menggunakan ROS2. Kode ini meluncurkan lingkungan simulasi Webots, memuat kendaraan Tesla, dan meluncurkan dua node ROS2: pengemudi Tesla dan pengikut jalur. Kode menggunakan penanganan acara untuk memastikan bahwa node ROS2 dijalankan kembali ketika simulasi Webots direset. Secara keseluruhan, kode ini menyediakan platform yang kuat dan fleksibel untuk mengembangkan dan menguji algoritma kontrol untuk kendaraan Tesla di simulator Webots.