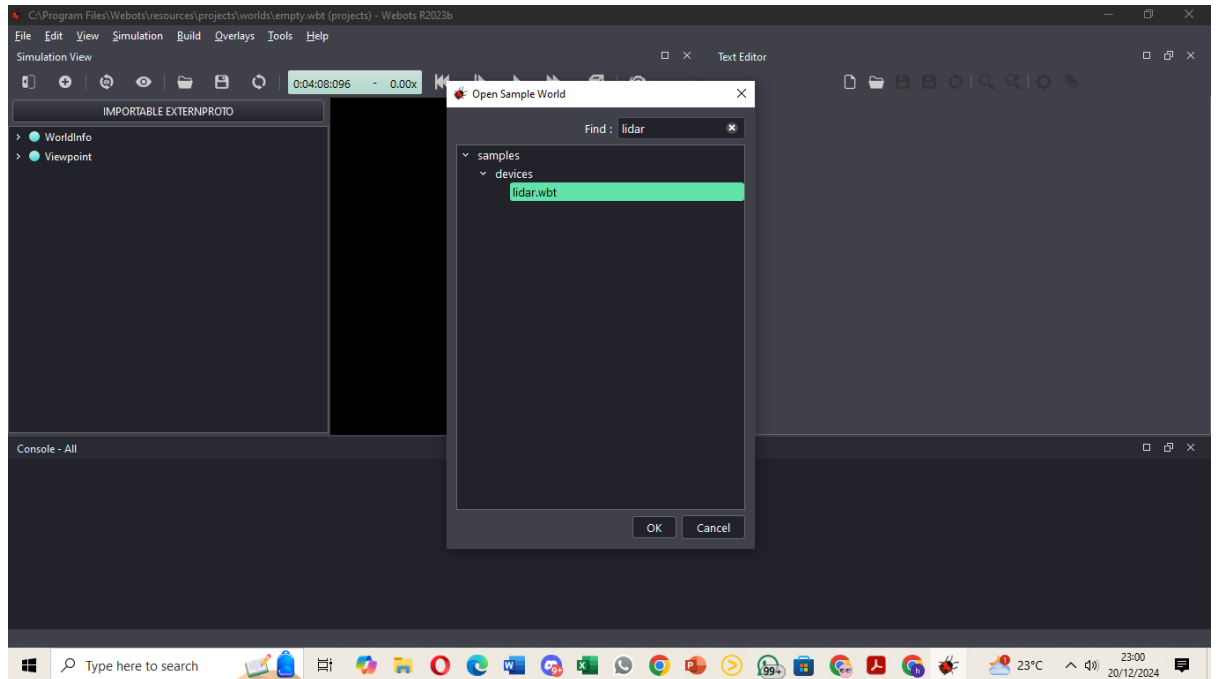


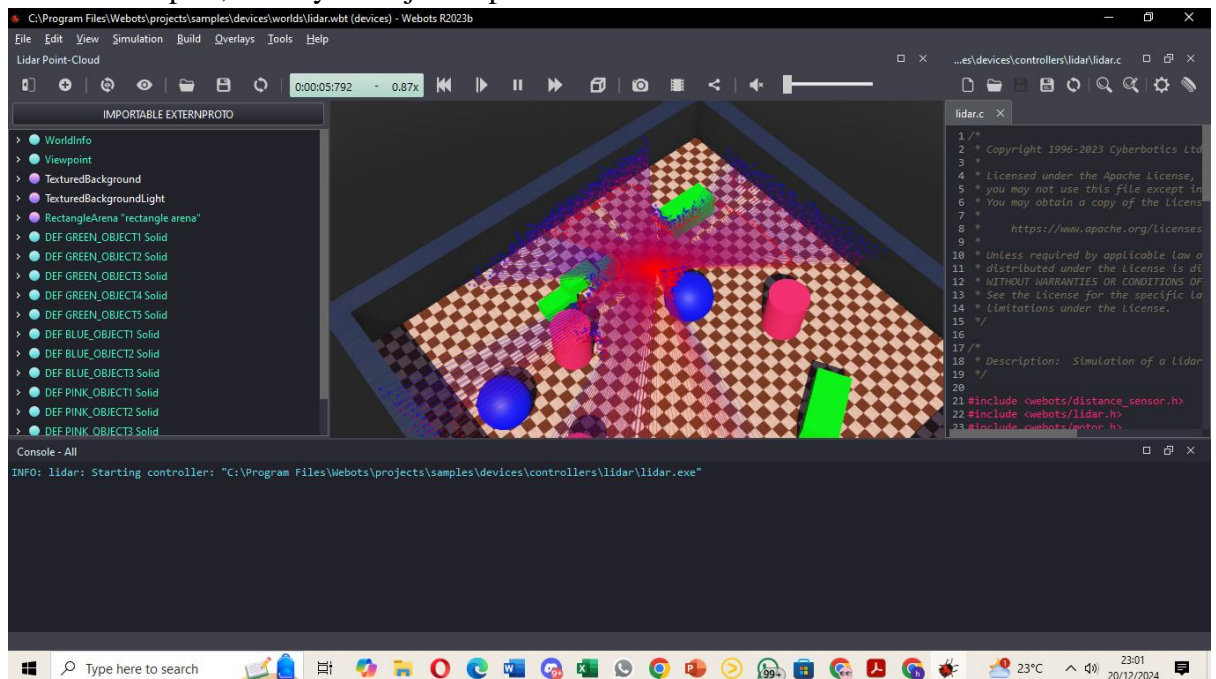
Ade Tirta Rahmat Hidayat – 1103203212 – TK45G06

## Dokumentasi Lidar Data Extraction dan Obstacle Detection pada Webots

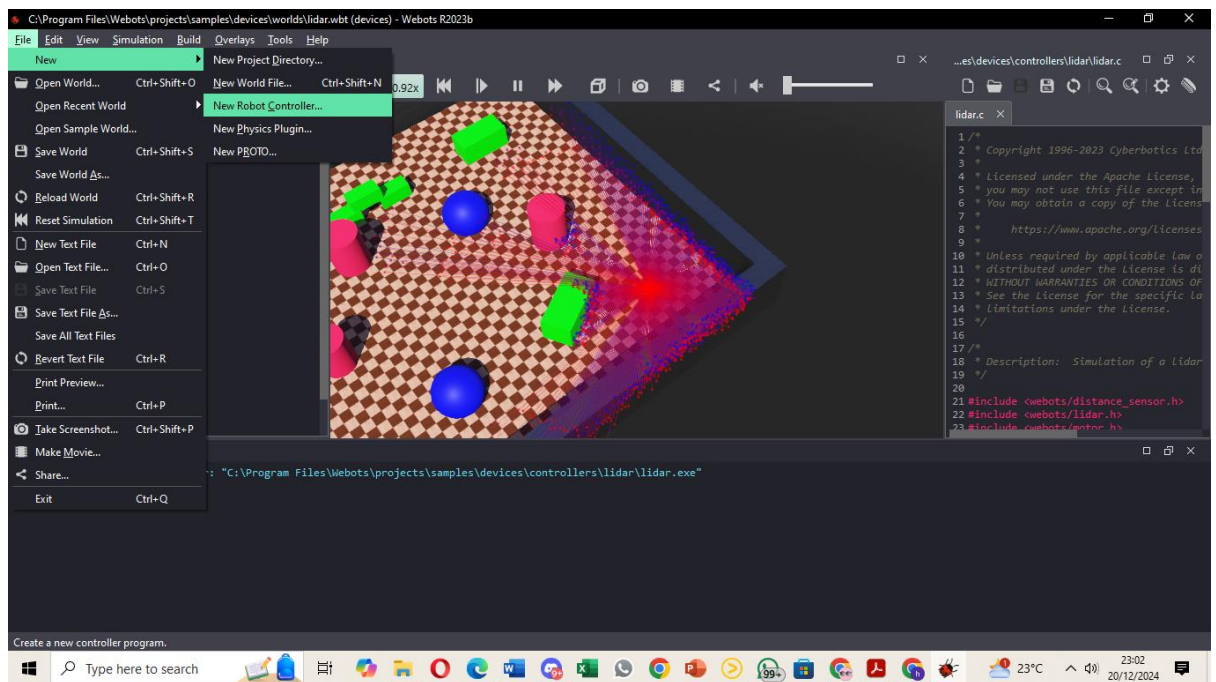
1. Pada webots, buka sample world dan cari 'lidar'. Kemudian pilih yang lidar.wbt



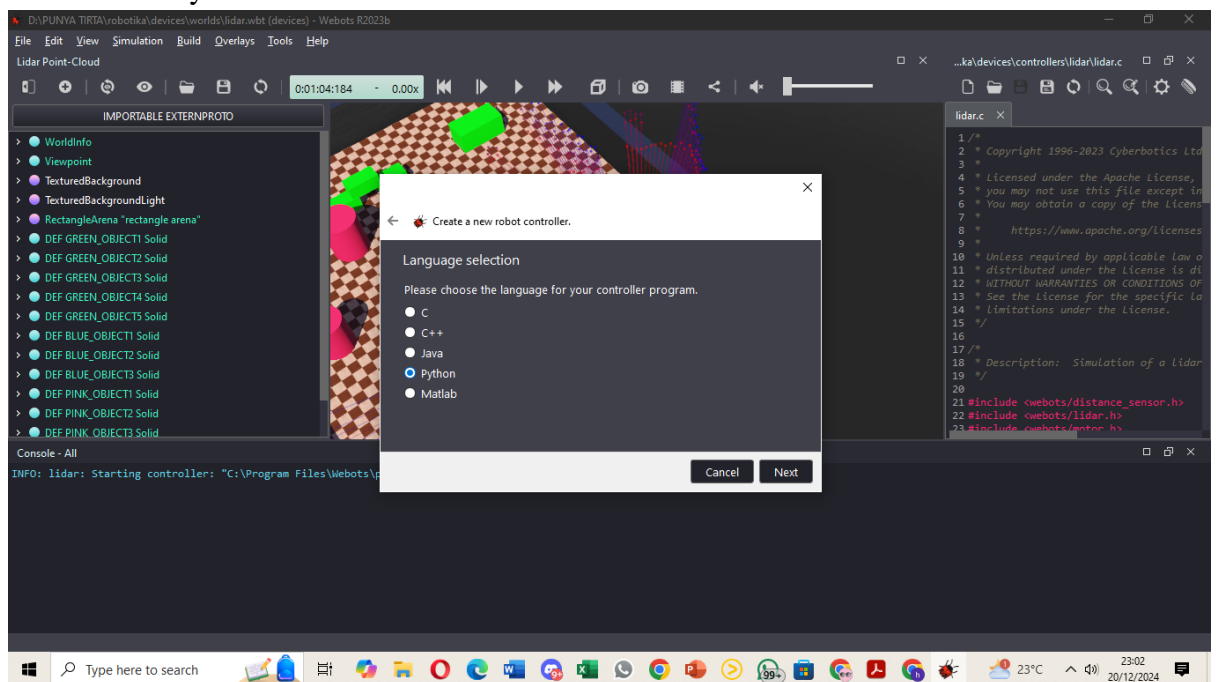
Setelah diimport, hasilnya menjadi seperti ini



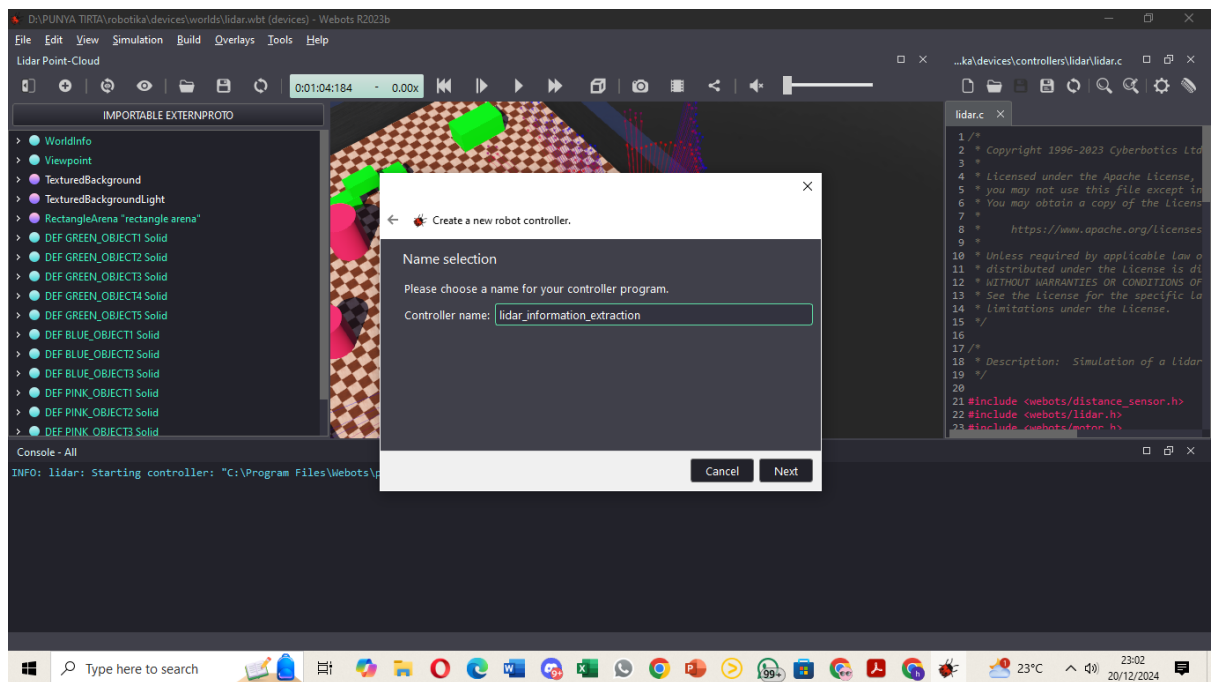
2. Buat file controller baru



## Pilih bahasa Python



## Beri nama controller tersebut



### 3. Masukkan codingan pada kotak di bawah ini ke controller yang baru saja dibuat

```
from controller import Robot

# Konstanta waktu langkah simulasi
TIME_STEP = 32

# Indeks untuk sensor kiri dan kanan
LEFT = 0
RIGHT = 1

# Inisialisasi robot
robot = Robot()

# Inisialisasi lidar
lidar = robot.getDevice("lidar")
lidar.enable(TIME_STEP)
lidar.enablePointCloud()

# Inisialisasi sensor jarak (ultrasonic)
us = [robot.getDevice("us0"), robot.getDevice("us1")]
for sensor in us:
    sensor.enable(TIME_STEP)

# Inisialisasi motor
left_motor = robot.getDevice("left wheel motor")
right_motor = robot.getDevice("right wheel motor")
left_motor.setPosition(float('inf')) # Aktifkan mode kecepatan
right_motor.setPosition(float('inf')) # Aktifkan mode kecepatan
left_motor.setVelocity(0.0)
right_motor.setVelocity(0.0)

# Koefisien empiris untuk penghindaran tabrakan
coefficients = [[12.0, -6.0], [-10.0, 8.0]]
base_speed = 6.0

# Fungsi untuk membaca data lidar
def extract_lidar_data():
```

```

        lidar_data = lidar.getRangeImage()
        print(f"Lidar Data {lidar_data[:10]}...") # Menampilkan 10 data
        pertama
        return lidar_data

# Fungsi untuk membaca data dari sensor jarak
def read_distance_sensors():
    distances = [sensor.getValue() for sensor in us]
    print(f"Distance Sensor Readings Left={distances[LEFT]:.2f},
    Right={distances[RIGHT]:.2f}")
    return distances

# Fungsi untuk menghitung kecepatan berdasarkan data sensor
def compute_speeds(us_values):
    speed = [0.0, 0.0]
    for i in range(2):
        for k in range(2):
            speed[i] += us_values[k] * coefficients[i][k]
    return speed

# Loop utama
while robot.step(TIME_STEP) != -1:
    # Baca data lidar dan ekstrak informasi
    lidar_data = extract_lidar_data()

    # Baca data sensor jarak
    us_values = read_distance_sensors()

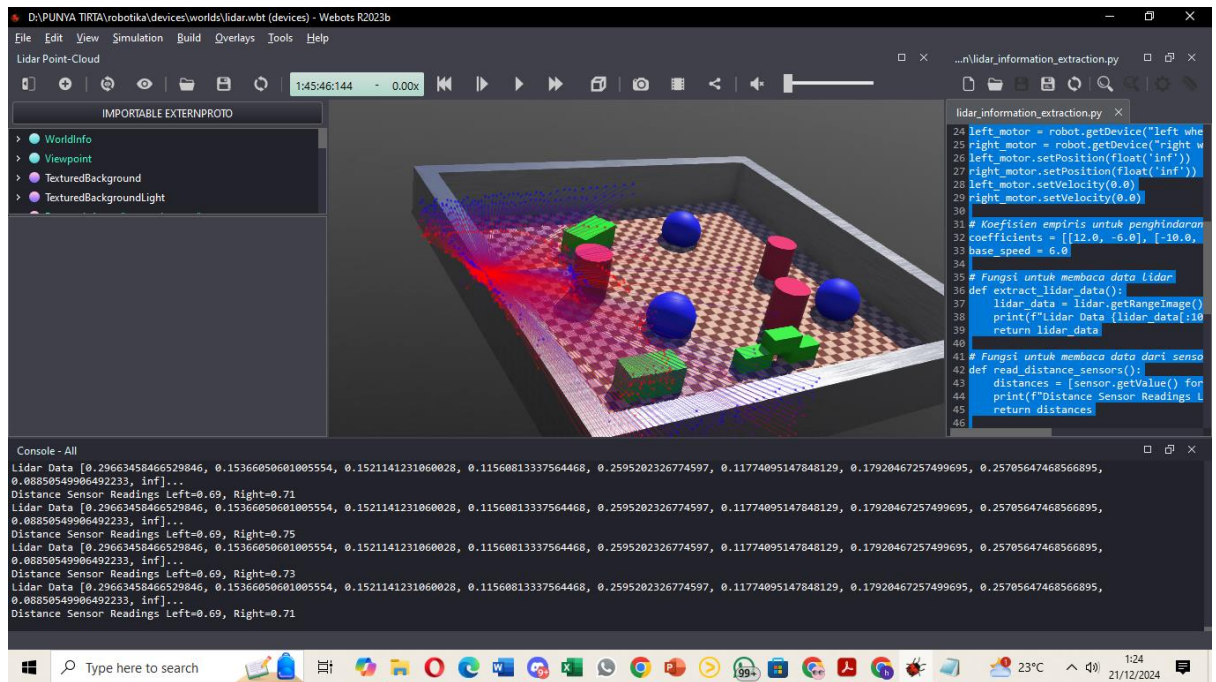
    # Hitung kecepatan roda berdasarkan data sensor
    speeds = compute_speeds(us_values)

    # Atur kecepatan motor
    left_motor.setVelocity(base_speed + speeds[LEFT])
    right_motor.setVelocity(base_speed + speeds[RIGHT])

    # Membersihkan memori setelah simulasi selesai
robot.cleanup()

```

4. Pasangkan controller tersebut pada robot dan jalankan.



5. Hasil simulasinya dapat dilihat pada link berikut:
- <https://drive.google.com/file/d/1j0SCuQ7BWW8HftOxfa3Un64ySWRMLWV/view?usp=sharing>