

SCRIPT

Pendahuluan :

Pada kesempatan kali ini, saya akan menjelaskan tutorial simulasi Robot Line Follower di Webots.

Line Follower adalah jenis robot atau algoritma yang dirancang untuk mengikuti jalur atau garis tertentu di permukaan, biasanya digunakan untuk belajar tentang control dan navigasi robot.

Persiapan :

- Komputer atau PC untuk menjalankan simulasi secara virtual.
- Webots = software / perangkat lunak open source untuk simulasi robotic 3D yang digunakan untuk membuat dan menguji algoritma control robot di lingkungan virtual.
- Virtual E-puck = robot edukasi kecil yang sering digunakan untuk penelitian dan pengajaran robotika. Dilengkapi sensor jarak, kamera, dan motor, robot ini cocok untuk simulasi dalam webots.
- Virtual Track = Merupakan jalur yang sudah dibuat sedemikian rupa untuk menjalankan simulasi.
- Python = Sebagai bahasa programming yang akan digunakan untuk mengatur algoritma robot.

Langkah - Langkah :

- Download Webots, dan Track pada link yang sudah saya cantumkan di deskripsi.
- Install Webots & Buka.
- New Project Directory > Beri Nama > Continue.
- Centang "add a rectangle arena" and click done.
- Klik Rectangle Arena pada area sebelah kiri > Kemudian Klik tombol "+" pada bagian atas.
- Klik PROTO nodes > gctronic > e-puck > e-puck (robot) > add.
- Save pekerjaan sebagai check point.
- Tambahkan sensor dengan klik E-puck > ground sensor slot > klik "+" > Base Nodes > Distance Sensor > Add.
- Setting Distance Sensor > Type "infra-red".
- Klik View > Optical Rendering > Show Distance Sensor.
- Setting posisi sensor > transition > ubah bagian Y menjadi "-0.01" > beri nama ir0
- Kemudian tambahkan sensor distance lagi dengan mengubah bagian Y menjadi "0.02" > beri nama ir1
- Posisikan rotasi sensor agar menghadap ke bawah
- Untuk menambahkan track, pindahkan file track ke folder protos
- Kemudian klik add > "TrackOne" > Atur Posisi Track
- Ubah arena menjadi 2x2 dan berwarna putih

- Posisikan Robot E-puck dengan jalur berada diantara kedua sensor
- Klik Wizard > New Robot controller > Python
- Beri Nama > Continue > Done
- Tulis algoritma pada teks editor dengan kode program :
from controller import Robot

```
def run_robot(robot):
    # Inisialisasi waktu dan kecepatan
    time_step = 32
    max_speed = 6.28 * 0.3

    # Inisialisasi motor
    left_motor = robot.getDevice('left wheel motor')
    right_motor = robot.getDevice('right wheel motor')

    # Set posisi dan kecepatan awal
    left_motor.setPosition(float('inf'))
    right_motor.setPosition(float('inf'))
    left_motor.setVelocity(0.0)
    right_motor.setVelocity(0.0)

    # Inisialisasi sensor IR
    left_ir = robot.getDevice('ir0')
    right_ir = robot.getDevice('ir1')
    left_ir.enable(time_step)
    right_ir.enable(time_step)

    # Konstanta threshold untuk deteksi garis
    BLACK_THRESHOLD = 5 # Sesuaikan dengan nilai pembacaan garis hitam

    while robot.step(time_step) != -1:
        # Baca nilai sensor
        left_ir_value = left_ir.getValue()
        right_ir_value = right_ir.getValue()

        print(f"left: {left_ir_value:.2f} right: {right_ir_value:.2f}")

        # Logika line following
        left_speed = max_speed
        right_speed = max_speed

        # Jika kedua sensor mendeteksi garis (hitam)
        if left_ir_value < BLACK_THRESHOLD and right_ir_value < BLACK_THRESHOLD:
            # Jalan lurus
            left_speed = max_speed
            right_speed = max_speed
```

```

# Jika sensor kiri mendeteksi garis
elif left_ir_value < BLACK_THRESHOLD:
    # Belok kiri
    left_speed = -max_speed * 0.5
    right_speed = max_speed
    print("Turn Left")

# Jika sensor kanan mendeteksi garis
elif right_ir_value < BLACK_THRESHOLD:
    # Belok kanan
    left_speed = max_speed
    right_speed = -max_speed * 0.5
    print("Turn Right")

# Set kecepatan motor
left_motor.setVelocity(left_speed)
right_motor.setVelocity(right_speed)

if __name__ == "__main__":
    my_robot = Robot()
    run_robot(my_robot)

```

- Klik save, kemudian ubah source controller pada E-puck dengan kode yg sudah dibuat.
- Klik Run.

Hasil :

Robot E-puck sudah mengikuti jalur sesuai dengan yang diharapkan.

Kesimpulan :

Simulasi Line Follower bertujuan untuk mengedukasi dengan menggunakan sensor yang bisa mendeteksi warna, antara warna hitam dan putih akan menghasilkan data yang berbeda yang kemudian diberi threshold untuk mengambil Keputusan. Ketika sensor sebelah kiri mendeteksi warna hitam, maka motor sebelah kanan akan bergerak sedangkan motor sebelah kiri mati, begitu juga sebaliknya. Sedangkan Ketika kedua sensor mendeteksi warna yang sama, maka kedua motor akan berjalan lurus. Sehingga Robot E-puck dapat berjalan mengikuti jalur yang disediakan.