

## Week 8 UTS Laporan Hasil Tutorial

### 1. Tujuan

Tujuan dari tutorial ini adalah untuk memahami cara memprogram robot **e-puck** di Webots agar dapat berputar di tempat dengan menggunakan Python. Tutorial ini juga melatih kemampuan implementasi dasar pengendalian motor pada robot simulasi.

---

### 2. Persiapan

Sebelum menjalankan simulasi, langkah-langkah berikut telah dilakukan:

1. **Instalasi Webots:** Webots telah diinstal di Windows, dan contoh dunia telah dimuat.
  2. **Menambahkan Robot e-puck:** Robot e-puck ditambahkan ke dunia simulasi melalui antarmuka **Add Node**.
  3. **Konfigurasi Controller:** Menulis kode Python yang mengatur pergerakan motor robot.
- 

### 3. Kode Program

Kode Python berikut digunakan untuk mengatur pergerakan robot e-puck agar berputar di tempat:

```
from controller import Robot

# Inisialisasi robot
robot = Robot()

# Waktu langkah simulasi (milidetik)
TIME_STEP = int(robot.getBasicTimeStep())

# Mendapatkan referensi motor kiri dan kanan
left_motor = robot.getDevice("left wheel motor")
right_motor = robot.getDevice("right wheel motor")

# Mengatur motor ke mode velocity (kecepatan)
left_motor.setPosition(float('inf'))
right_motor.setPosition(float('inf'))

# Kecepatan roda untuk berputar
left_motor.setVelocity(2.0) # Roda kiri bergerak maju
right_motor.setVelocity(-2.0) # Roda kanan bergerak mundur

# Loop simulasi
while robot.step(TIME_STEP) != -1:
    pass # Robot terus berputar di tempat
```

---

## 4. Hasil Simulasi

### 1. Robot Berputar di Tempat:

- Robot e-puck berhasil berputar di tempat seperti yang diinginkan.
- Roda kiri bergerak maju dengan kecepatan 2.0, sementara roda kanan bergerak mundur dengan kecepatan -2.0.

### 2. Observasi Gerakan:

- Kecepatan roda menghasilkan rotasi konstan dengan putaran yang stabil.
- Robot tetap berada di satu lokasi tanpa bergerak maju atau mundur.

### 3. Respons Program:

- Kode berjalan tanpa error, dan robot merespons instruksi dengan baik.
- 

## 5. Analisis

### 1. Efisiensi Kode:

- Struktur kode sederhana dan fokus hanya pada pengaturan kecepatan motor.
- Loop simulasi berjalan sesuai dengan langkah waktu yang diberikan oleh Webots.

### 2. Fleksibilitas:

- Kecepatan putaran dapat diubah dengan menyesuaikan nilai **setVelocity()**.
  - Pola gerakan lain, seperti rotasi lebih cepat atau lebih lambat, dapat dibuat dengan cara serupa.
- 

## 6. Kesimpulan

Melalui tutorial ini, kami berhasil:

1. Mengonfigurasi motor robot e-puck untuk berputar di tempat menggunakan Python.
2. Memahami cara kerja pengendalian motor di Webots dengan metode **setVelocity()**.
3. Melakukan simulasi rotasi yang stabil dan terus-menerus.

Tutorial ini memberikan dasar yang kuat untuk membuat simulasi gerakan lebih kompleks di masa depan. Dengan pengetahuan ini, pengendalian robot lain di Webots juga menjadi lebih mudah.

---

## 7. Rekomendasi

### 1. Eksperimen Lebih Lanjut:

Cobalah untuk membuat variasi gerakan, seperti memodifikasi kecepatan atau menambahkan logika untuk menghentikan rotasi setelah beberapa waktu.

2. **Integrasi Lanjutan:**

Pelajari cara mengintegrasikan sensor untuk membuat robot lebih interaktif dalam simulasi.