

Intro: "Halo semua! Pada video kali ini, kita akan melihat simulasi robot e-puck yang melakukan navigasi mengikuti dinding kiri pada sebuah labirin horizontal menggunakan Webots dan Python."

1. Pengenalan Lingkungan: "Kita memiliki sebuah lingkungan labirin yang terdiri dari beberapa lorong horizontal dengan dinding yang memisahkan tiap jalur. Robot e-puck ditempatkan di sudut kiri bawah dengan tujuan untuk berjalan mengikuti dinding kiri hingga mencapai sudut kanan atas labirin."

2. Deskripsi Kode: "Sekarang, mari kita bahas bagaimana kode Python ini bekerja. Kita akan mulai dengan inisialisasi dasar robot dan sensor yang akan digunakan."

2.1. Inisialisasi Robot dan Sensor: "Pada bagian awal kode, kita melakukan inisialisasi robot dan sensor-sensor inframerah yang terdapat pada e-puck. Robot ini dilengkapi dengan 8 sensor jarak, yaitu ps0 hingga ps7, yang membantu mendeteksi objek atau dinding di sekitar robot. Kita juga menginisialisasi motor roda kiri dan kanan untuk mengendalikan pergerakan robot."

python

Copy code

```
# Inisialisasi sensor jarak inframerah (ps0 hingga ps7)
```

```
prox_sensor = []
```

```
sensor_names = ["ps0", "ps1", "ps2", "ps3", "ps4", "ps5", "ps6", "ps7"]
```

2.2. Threshold dan Kecepatan Dasar: "Selanjutnya, kita mendefinisikan beberapa variabel penting seperti FRONT_THRESHOLD dan SIDE_THRESHOLD. Nilai ini digunakan sebagai batas deteksi jarak dinding di depan dan di samping kiri robot. Selain itu, kita juga mengatur kecepatan dasar untuk robot."

python

Copy code

```
FRONT_THRESHOLD = 150.0 # Jarak untuk deteksi dinding di depan
```

```
SIDE_THRESHOLD = 80.0 # Jarak untuk deteksi dinding di samping kiri
```

3. Algoritma Navigasi: "Kini kita masuk ke bagian utama, yaitu logika navigasi robot. Di sini, robot akan menggunakan beberapa kondisi berdasarkan sensor jarak untuk bergerak dan berbelok sesuai dengan situasi labirin."

3.1. Kondisi 1: Ada Dinding di Depan: "Jika sensor depan (ps0) atau sensor depan kiri (ps6) mendeteksi adanya dinding yang dekat, robot akan berbelok ke kanan untuk menghindari tabrakan. Kita mengatur kecepatan roda kiri lebih tinggi daripada roda kanan agar robot dapat berbelok."

python

Copy code

```
if front > FRONT_THRESHOLD or front_left > FRONT_THRESHOLD:
```

```
    left_speed = 0.5 * MAX_SPEED
```

```
    right_speed = -0.2 * MAX_SPEED
```

3.2. Kondisi 2: Tidak Ada Dinding di Samping Kiri: "Jika sensor samping kiri (ps7) mendeteksi jarak yang lebih besar dari SIDE_THRESHOLD, ini berarti tidak ada dinding di sebelah kiri robot. Dalam kondisi ini, robot akan sedikit berbelok ke kiri untuk mendekati dinding."

python

Copy code

```
elif left_side < SIDE_THRESHOLD:
```

```
    left_speed = 0.3 * MAX_SPEED
```

```
    right_speed = 0.5 * MAX_SPEED
```

3.3. Kondisi 3: Robot Berada di Ujung Lorong: "Saat robot berada di ujung lorong dan sensor depan kanan (ps1) mendeteksi bahwa tidak ada dinding di depannya, robot akan berbelok ke atas atau bawah untuk berpindah ke lorong berikutnya. Logika ini memungkinkan robot untuk menyelesaikan labirin dengan bergerak zig-zag."

python

Copy code

```
elif front_right > TURN_THRESHOLD:
```

```
    left_speed = -0.2 * MAX_SPEED
```

```
    right_speed = 0.5 * MAX_SPEED
```

3.4. Berjalan Lurus Mengikuti Dinding: "Jika tidak ada kondisi khusus yang terdeteksi, robot akan berjalan lurus dengan menjaga jarak dari dinding di sebelah kiri."

python

Copy code

```
else:
```

```
    left_speed = 0.5 * MAX_SPEED
```

```
    right_speed = 0.5 * MAX_SPEED
```

4. Hasil Simulasi: "Pada simulasi ini, Anda dapat melihat bagaimana robot e-puck berhasil mengikuti dinding kiri dan bergerak dari kiri bawah ke kanan atas labirin secara efisien. Robot melakukan zig-zag melewati lorong-lorong horizontal dengan menggunakan sensor inframerah untuk mendeteksi jarak dinding dan menyesuaikan arah pergerakannya."

5. Kesimpulan: "Dengan menggunakan algoritma navigasi sederhana ini, kita dapat membuat robot e-puck bergerak mengikuti dinding kiri dalam berbagai pola labirin. Metode ini sangat efektif pada labirin dengan pola yang terstruktur seperti yang kita lihat pada simulasi kali ini."

Outro: "Terima kasih telah menonton video ini! Jangan lupa untuk memberikan komentar jika Anda memiliki pertanyaan atau saran. Sampai jumpa di video berikutnya, dan semoga sukses dengan simulasi Webots Anda!"