

## Control of Mobile Robots – Lab Demos

Nama : Al Ghifary Akmal Nasheeri

NIM : 1103201242

Kelas : TK-44-06

### Webots Lab 1 - Basic Robot Programming

Video ini membahas Lab 1 Pemrograman Robot Dasar di Webots. Ada 2 task yaitu menggerakkan robot dalam garis lurus dan belajar membaca encoder pada robot.

Task pertama adalah menggerakkan robot dalam garis lurus selama jarak tertentu dengan kecepatan konstan. Pembicara menguji kecepatan yang berbeda, termasuk kecepatan yang terlalu tinggi untuk motor dan kecepatan nol. Robot bergerak sesuai harapan untuk kecepatan yang berada dalam rentang motor.

Task kedua adalah belajar membaca encoder pada robot. Pembicara menggunakan simulator untuk memutar robot mengelilingi lingkaran dengan radius tertentu dengan kecepatan tertentu. Pembicara menguji kecepatan dan jari-jari yang berbeda, dan robot bergerak sesuai harapan.

Secara keseluruhan, video ini memberikan gambaran dasar tentang cara memprogram robot untuk bergerak dalam garis lurus dan membaca encoder pada robot. Pembicara memberikan penjelasan yang jelas tentang task-task dan hasil tes.

Berikut adalah penjelasan lebih rinci tentang setiap task:

#### Task 1: Gerak Lurus

Untuk menggerakkan robot dalam garis lurus, kita perlu menentukan kecepatan dan arah robot. Kecepatan dapat ditentukan dengan menggunakan fungsi `setVelocity()`, dan arah dapat ditentukan dengan menggunakan fungsi `setRotation()`.

Pada video, pembicara menggunakan fungsi `setVelocity()` untuk menetapkan kecepatan robot. Pembicara juga menggunakan fungsi `getDistance()` untuk mengukur jarak yang telah ditempuh robot.

Berikut adalah kode yang digunakan pembicara untuk menggerakkan robot dalam garis lurus:

```
# Tentukan kecepatan robot
velocity = 0.5

# Atur kecepatan robot
robot.setVelocity(velocity)

# Ulangi hingga robot mencapai jarak tertentu
```

```
while robot.getDistance() < 10:
```

```
    # Tunggu satu detik
```

```
    time.sleep(1)
```

Kode ini akan menggerakkan robot dengan kecepatan 0,5 meter per detik hingga robot mencapai jarak 10 meter.

## Task 2: Membaca Encoder

Encoder adalah sensor yang digunakan untuk mengukur sudut atau jarak yang telah ditempuh oleh robot. Untuk membaca encoder, kita dapat menggunakan fungsi `getEncoder()`.

Pada video, pembicara menggunakan fungsi `getEncoder()` untuk mengukur sudut yang telah ditempuh oleh robot. Pembicara juga menggunakan fungsi `getVelocity()` untuk menghitung kecepatan robot.

Berikut adalah kode yang digunakan pembicara untuk membaca encoder:

```
# Atur robot untuk berputar
```

```
robot.setRotation(0.5)
```

```
# Ulangi hingga robot berputar 360 derajat
```

```
while robot.getEncoder() < 360:
```

```
    # Tunggu satu detik
```

```
    time.sleep(1)
```

```
# Cetak kecepatan robot
```

```
print(robot.getVelocity())
```

Kode ini akan membuat robot berputar dengan kecepatan 0,5 radian per detik hingga robot berputar 360 derajat. Setelah robot berputar 360 derajat, kode akan mencetak kecepatan robot.

## Webots Lab 2 – Kinematics

Webots Lab 2 berfokus pada kinematika robot, yaitu studi tentang gerak robot tanpa mempertimbangkan gaya dan torsi yang terlibat. Dalam lab ini, kita menggunakan simulator Webots untuk memprogram robot penggerak diferensial untuk bergerak di sepanjang berbagai jalur menggunakan Python.

Task:

Lab terdiri dari tiga task utama:

1. Kotak: Program robot untuk bergerak mengelilingi kotak dengan panjang sisi yang ditentukan.
2. Lingkaran: Buat robot mengikuti jalur melingkar dengan jari-jari yang diberikan.
3. Oval: Arahkan robot di sepanjang jalur berbentuk oval yang ditentukan oleh jari-jari dan arahnya.

Konsep Kunci:

- Penggerak diferensial: Jenis robot ini menggunakan dua motor yang dikendalikan secara independen di setiap sisi untuk mencapai gerakan. Dengan menyesuaikan kecepatan motor, Kita dapat mengontrol rotasi dan translasi robot.
- Persamaan kinematika: Persamaan ini menghubungkan kecepatan roda robot ke kecepatan linier dan sudutnya. Kita akan menggunakan persamaan ini untuk menghitung kecepatan motor yang diperlukan untuk setiap task.
- Pemrograman Python: Task lab melibatkan penulisan kode Python untuk mengontrol motor robot dan membaca nilai sensor.

### **Webots Lab 3 - Navigation with Distance Sensors**

Lab ini berfokus pada mengajarkan robot untuk menavigasi menggunakan sensor jarak. Secara khusus, Kita akan bekerja dengan robot e-puck yang dilengkapi dengan tiga sensor jarak laser: satu di depan dan satu di setiap sisi. Sensor ini akan membantu robot mengamati lingkungannya dan menghindari rintangan saat bergerak.

Task:

Lab terdiri dari dua task utama:

1. Penggerak Garis Lurus dengan Penghindaran Hambatan:
  - Program robot untuk melaju dalam garis lurus sambil mempertahankan jarak tertentu dari setiap dinding yang ditemui menggunakan sensor depan dan samping.
  - Terapkan algoritma kontrol proporsional untuk menyesuaikan kecepatan robot berdasarkan pembacaan sensor.
  - Tujuannya adalah agar robot menavigasi dengan lancar di sepanjang jalur lurus sambil menghindari tabrakan dengan rintangan apa pun.
2. Navigasi Koridor:
  - Program robot untuk mengikuti dinding kiri koridor sambil mempertahankan jarak tertentu dari kedua dinding menggunakan ketiga sensor.
  - Perluas algoritma kontrol untuk menangani tikungan dan belokan di koridor.
  - Robot harus dapat menavigasi koridor secara otonom tanpa terjebak atau bertabrakan dengan dinding.

Konsep Kunci:

- Sensor jarak: Sensor ini mengukur jarak ke objek dalam bidang penglihatannya.
- Kontrol proporsional: Algoritma kontrol ini menyesuaikan output (misalnya, kecepatan motor) berdasarkan kesalahan antara nilai yang diinginkan dan aktual (misalnya, jarak ke rintangan).
- Kinematika robot: Memahami bagaimana roda dan motor robot diterjemahkan ke dalam gerakannya sangat penting untuk memprogram perilaku navigasinya.

## **Webots Lab 4 - Navigation with Camera**

Webots Lab 4 - Navigasi dengan Kamera oleh AJ Barea menjelajahi dunia menarik navigasi robot berbasis penglihatan! Dalam lab ini, Kita akan melengkapi robot Kita dengan kekuatan penglihatan menggunakan kamera bawaan dan melatihnya untuk menavigasi lingkungannya berdasarkan petunjuk visual.

Berikut adalah pembagian aspek-aspek kuncinya:

Task:

- Mengikuti Garis: Arahkan robot Kita untuk mengikuti garis berwarna di tanah menggunakan kamera sebagai "matanya." Ini mengajarkan pemrosesan gambar dasar dan kontrol robot berdasarkan umpan balik visual.
- Deteksi dan Penghindaran Objek: Latih robot Kita untuk mengidentifikasi dan menghindari rintangan dengan mengenali warna atau bentuk spesifik yang ditangkap oleh kamera. Ini melibatkan segmentasi gambar dan perencanaan jalur berdasarkan persepsi visual.
- Mencapai Tujuan: Program robot Kita untuk menavigasi ke arah objek target menggunakan kamera untuk melacak lokasinya dan menyesuaikan jalurnya sesuai. Ini menggabungkan pengenalan objek dengan penghindaran rintangan dan perencanaan berorientasi tujuan.

Konsep Kunci:

- Pemrosesan gambar: Teknik untuk mengekstrak informasi yang relevan dari gambar kamera, seperti deteksi warna, pengenalan bentuk, dan penemuan tepi.
- Penglihatan komputer: Ilmu dan teknik membuat komputer "melihat" dan memahami dunia visual.
- Perencanaan jalur: Algoritma untuk menentukan rute optimal bagi robot untuk mencapai tujuannya sambil menghindari rintangan.
- Kontrol umpan balik: Menggunakan data sensor (dalam hal ini, masukan kamera) untuk terus menyesuaikan perilaku robot dan memastikannya tetap di jalur.

## **Webots Lab 5 – Localization**

Webots Lab 5 oleh AJ Barea mengajak Kita menyelami dunia robot localization! Di sini, Kita akan beralih dari sekadar mengendalikan robot mengikuti jalur yang ditentukan, dan melangkah ke wilayah yang belum terpetakan. Tantangannya adalah melatih robot untuk memahami lokasinya sendiri dalam lingkungan yang tidak dikenalnya.

Task:

- Kalibrasi Sensor: Pastikan sensor robotimu, seperti odometer dan enkoder, bekerja akurat untuk pengukuran dasar seperti jarak dan heading.
- Pemetaan Odometri: Gunakan data sensor untuk membangun peta sederhana lingkungan sekitar robot secara real-time. Visualisasikan peta ini dan pantau posisi robot di dalamnya.
- Landmarking (Penkita Landmark): Identifikasi landmark atau fitur unik dalam lingkungan dan gunakan untuk memperbaiki estimasi lokasi robot. Ini meningkatkan akurasi dan ketahanan pemetaan odometri.
- Filter Kalman: Terapkan filter Kalman untuk menggabungkan data dari sensor yang berbeda (odometer, landmark) dan menghasilkan estimasi lokasi yang lebih stabil dan akurat.

Konsep Kunci:

- Localization (Pelokalan): Kemampuan robot memahami posisinya sendiri dalam lingkungannya.
- Odometri: Teknik estimasi posisi robot berdasarkan pengukuran jarak yang ditempuh.
- Pemetaan: Membangun representasi lingkungan sekitarnya secara digital.
- Landmark (Penkita Landmark): Fitur unik dan mudah dikenali dalam lingkungan yang digunakan untuk koreksi posisi.
- Filter Kalman: Algoritma estimasi yang menggabungkan data dari sensor yang berbeda dengan mempertimbangkan ketidakpastian masing-masing.

## **Webots Lab 6 - Mapping & Path Planning**

Webots Lab 6 oleh AJ Barea menjelajahi dunia menarik pemetaan dan perencanaan jalur robot, dua keterampilan penting bagi setiap agen otonom yang menavigasi lingkungannya.

Task:

- Pemetaan: Robot Kita akan menggunakan pembacaan sensor, seperti sensor jarak atau LIDAR, untuk membangun representasi lingkungan sekitarnya. Peta ini dapat berupa grid atau representasi grafis yang lebih kompleks.

- Perencanaan Jalur: Berdasarkan peta yang dibangun dan posisi serta tujuan robot saat ini, Kita akan mengimplementasikan algoritma seperti Dijkstra atau A\* untuk menemukan jalur terpendek atau optimal untuk mencapai tujuannya. Ini melibatkan pencarian di peta dan evaluasi jalur potensial yang berbeda.
- Mengikuti Jalur: Gunakan umpan balik dari posisi dan arah robot untuk memandunya di sepanjang jalur yang dipilih, mengoreksi penyimpangan apa pun dan memastikannya mencapai tujuannya secara efisien.

#### Konsep Kunci:

- Pemetaan: Membangun representasi lingkungan berdasarkan data sensor, termasuk rintangan, dinding, dan ruang terbuka.
- Perencanaan jalur: Mencari rute optimal antara titik awal dan tujuan dalam peta yang diketahui.
- Algoritma Dijkstra: Algoritma penemuan jalur yang menemukan jalur terpendek berdasarkan bobot tepi.
- *Algoritma A\**: Algoritma heuristik yang memperkirakan jarak ke tujuan, mengarah pada penemuan jalur yang berpotensi lebih cepat.
- Kontrol umpan balik: Menggunakan data sensor untuk terus menyesuaikan gerakan robot dan menjaganya tetap di jalur.