

Nama : Fadhil Dzikri Aqila

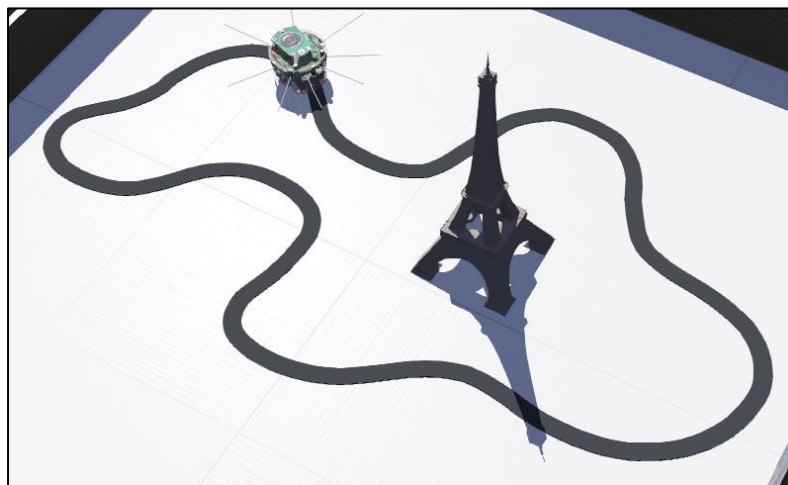
NIM : 1103213136

Kelas : TK-45-G09

Laporan Webots Tutorial Line Follower Robot using e-puck

1. Pendahuluan

Tutorial ini dibuat untuk memberikan pemahaman tentang cara membuat dan menguji robot line follower menggunakan Webots dan bahasa pemrograman C. Line follower adalah robot yang mengikuti garis tertentu (biasanya hitam pada permukaan putih) dengan memanfaatkan sensor. Tujuan tutorial ini adalah untuk memberikan pengetahuan dasar tentang pembuatan program line follower agar bisa digunakan dalam proyek robotika yang lebih kompleks di masa depan.



Line follower banyak digunakan di industri untuk otomatisasi, serta di bidang pendidikan untuk memperkenalkan prinsip dasar pemrograman dan robotika. Tutorial ini cocok untuk pemula maupun pengguna yang memiliki dasar robotika, agar dapat mengikuti langkah-langkah dan membangun program line follower dengan mudah.

2. Persiapan

Alat dan Perangkat Lunak:

1. Koneksi Internet – Diperlukan untuk mengunduh Webots jika belum terpasang.
2. Webots versi 2022a – Aplikasi simulator robot untuk membuat lingkungan virtual robotik.

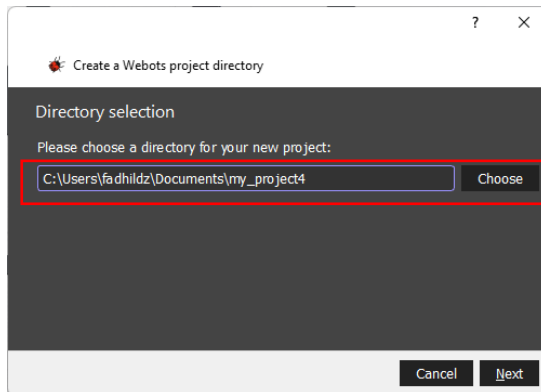


3. Langkah Implementasi

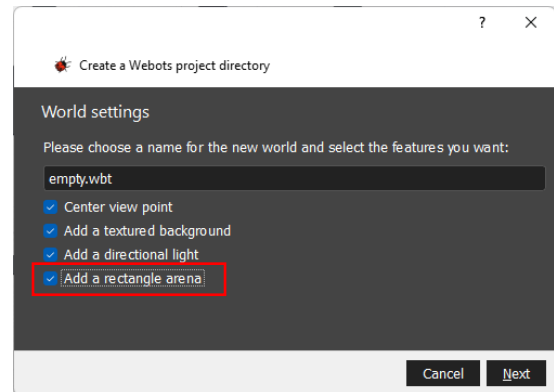
Berikut adalah langkah-langkah detail untuk membuat line follower robot dengan e-puck di Webots:

1. Buat Project Directory baru

- Buka Webots dan buat Project Directory baru dengan memilih menu Wizards → New Project Directory...



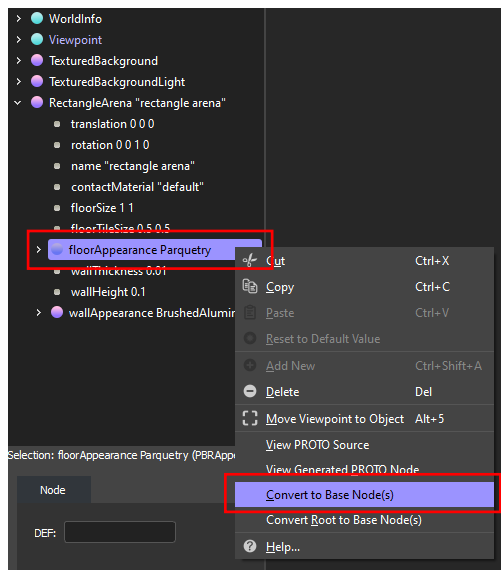
Pilih lokasi project ingin disimpan.



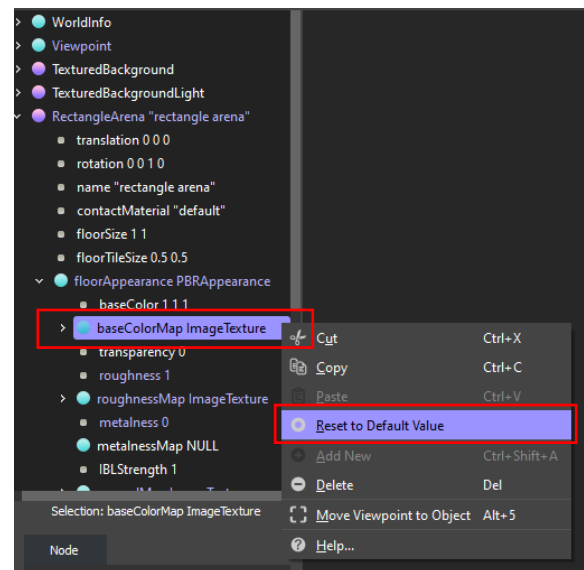
Beri nama world Webots nya, lalu ceklis 'Add a rectangle arena' untuk menambahkan arena.

2. Ubah Warna dan Ukuran Arena

- Atur warna arena menjadi putih untuk memberikan kontras yang jelas bagi sensor robot saat membaca garis hitam.



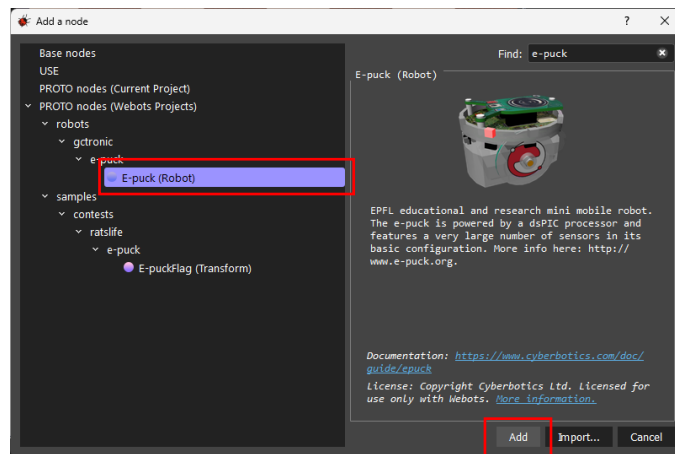
Klik "Convert to Base Nodes(s)" pada "floorAppearance Parquetry"



Klik "Reset to Default Value" pada "baseColorMap Image Texture"

3. Tambahkan Robot e-puck

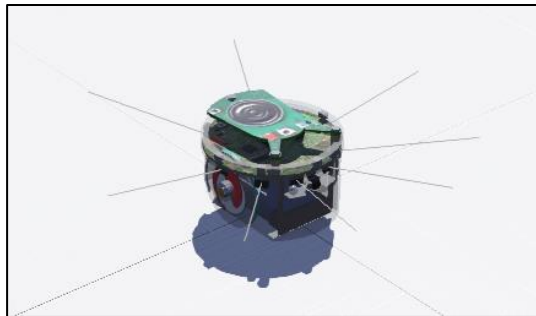
- Klik Arena, lalu klik ikon plus atau menggunakan shortcut (Ctrl+Shift+A), lalu tambahkan robot e-puck ke dalam proyek.



- Robot ini akan kita jadikan line follower karena memiliki fitur yang mendukung pemasangan sensor dan kontroler.

4. Aktifkan DistanceSensor Rays

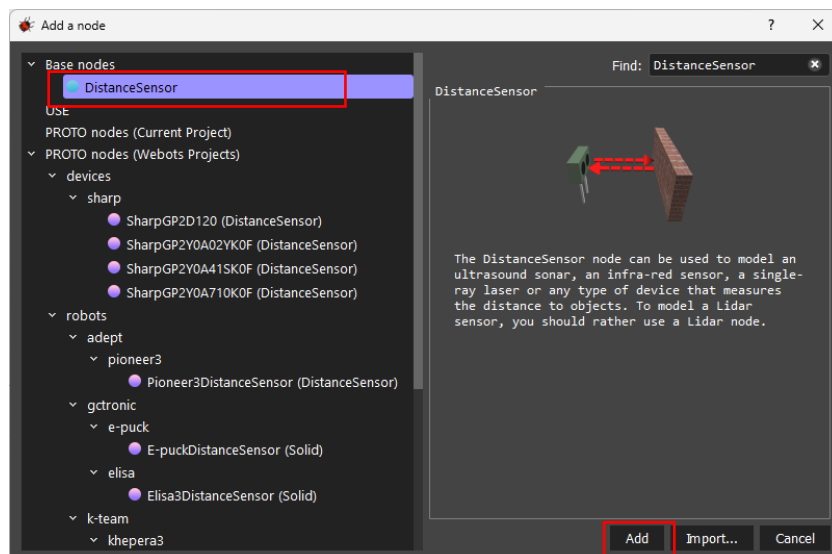
- Aktifkan DistanceSensor Rays di robot e-puck dengan memilih menu View → Optional Rendering → Show DistanceSensor Rays atau menggunakan shortcut (Ctrl+F10)



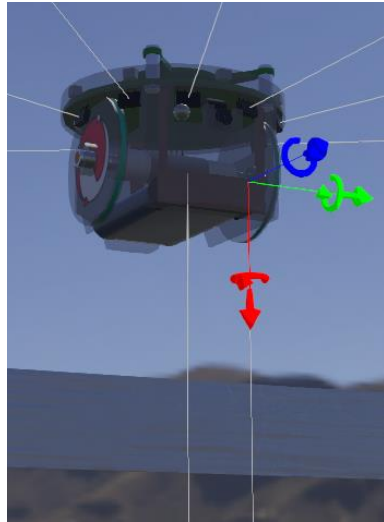
- Show DistanceSensor Rays ini membantu untuk melihat bagaimana sensor mendeteksi garis selama simulasi.

5. Pasang Sensor di Robot e-puck

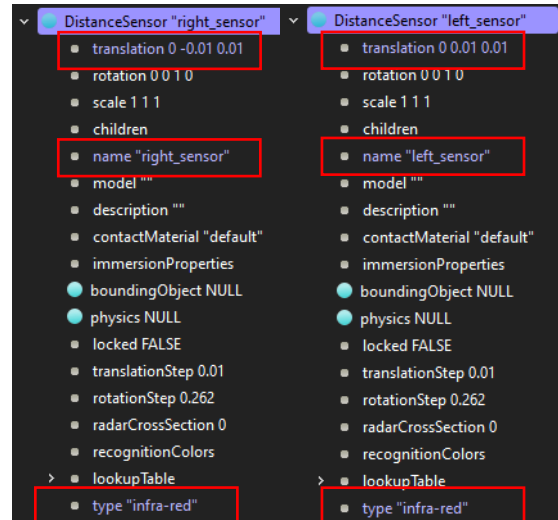
- Klik robot e-puck, Add new pada groundSensorsSlot lalu tambahkan 2 DistanceSensor.



- Setting kedua DistanceSensor.



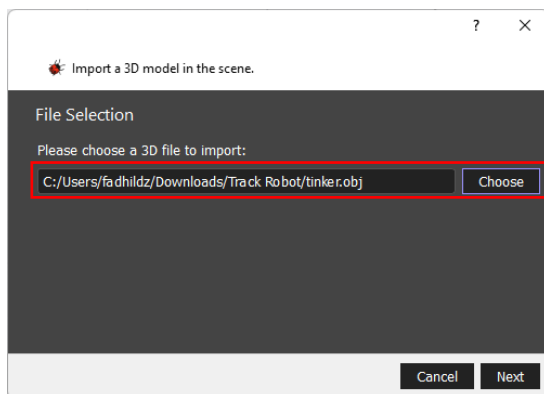
Posisikan sensor ke bawah agar dapat membaca warna jalur robot.



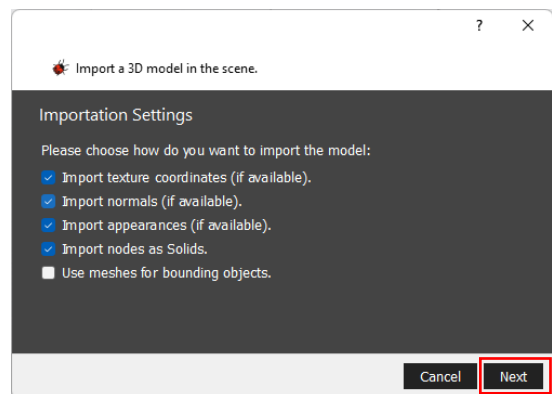
Sesuaikan translation, name, dan type untuk masing-masing DistanceSensor.

6. Tambah Track Robot

- Tambahkan track (jalur) dengan memilih menu File → Import 3D Model...

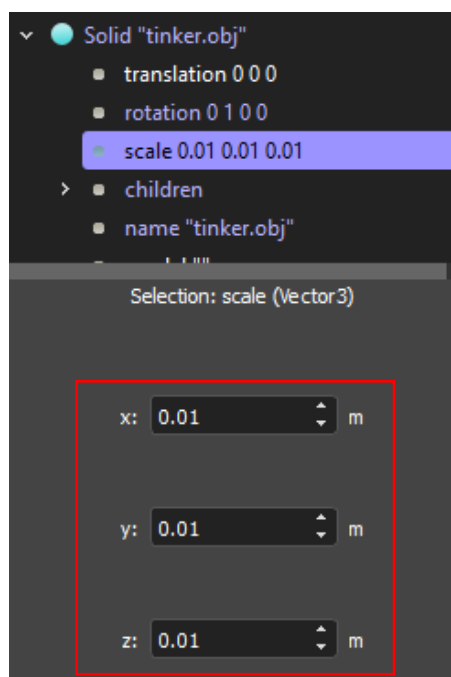


Cari file dengan format .dae, .stl, dan .obj



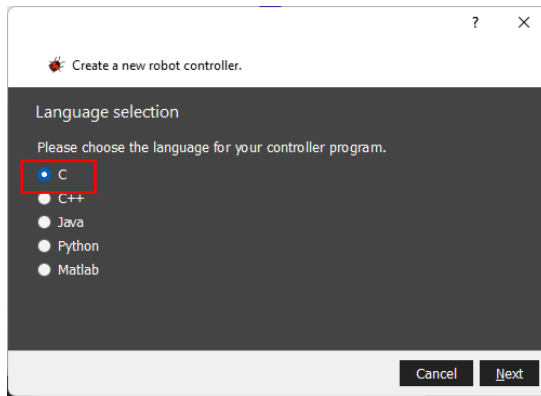
Next untuk import 3D Model yang dipilih

- Atur scale nya menjadi 0,01 untuk menyesuaikan ukuran track.

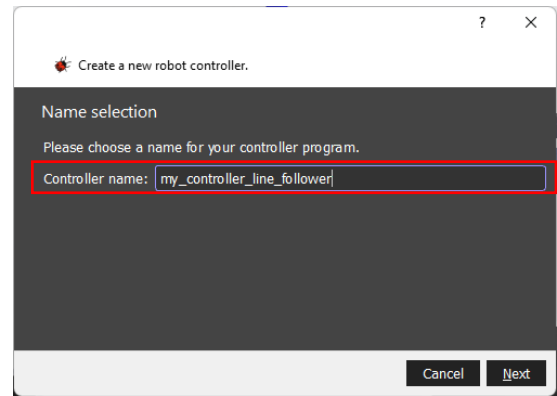


7. Tambah Robot Controller

- Tambah Robot Controller dengan memilih menu Wizards → New Robot Controller...



Pilih bahasa pemrograman untuk
controller robot.



Beri nama controller robot nya.

- Tulis kode berikut untuk controller robot line follower.

```
#include <webots/motor.h> // Library untuk fungsi utama robot Webots
#include <webots/motor.h> // Library untuk kontrol motor pada robot
#include <webots/distance_sensor.h> // Library untuk kontrol sensor jarak pada robot
#include <stdio.h> // Library standar C untuk input-output

#define TIME_STEP 64 // Mendefinisikan interval waktu dalam milidetik untuk setiap langkah simulasi

int main(int argc, char **argv) {
    wb_robot_init(); // Inisialisasi fungsi robot dari Webots

    // Mendeklarasikan motor kanan dan kiri
    WbDeviceTag motor_kanan = wb_robot_get_device("right wheel motor"); // Mendapatkan akses ke motor kanan
    WbDeviceTag motor_kiri = wb_robot_get_device("left wheel motor"); // Mendapatkan akses ke motor kiri

    // Mendeklarasikan sensor kanan dan kiri
    WbDeviceTag sensor_kanan = wb_robot_get_device("right_sensor"); // Mendapatkan akses ke sensor kanan
    WbDeviceTag sensor_kiri = wb_robot_get_device("left_sensor"); // Mendapatkan akses ke sensor kiri

    // Mengaktifkan sensor jarak dengan interval waktu yang telah ditentukan
    wb_distance_sensor_enable(sensor_kanan, TIME_STEP); // Mengaktifkan sensor kanan
    wb_distance_sensor_enable(sensor_kiri, TIME_STEP); // Mengaktifkan sensor kiri

    // Mengatur posisi motor agar berputar terus-menerus tanpa batas
    wb_motor_set_position(motor_kanan, INFINITY); // Motor kanan berputar tanpa batas
    wb_motor_set_position(motor_kiri, INFINITY); // Motor kiri berputar tanpa batas

    double readIR_kanan, readIR_kiri; // Variabel untuk menyimpan nilai sensor kanan dan kiri

    // Loop utama simulasi robot
    while (wb_robot_step(TIME_STEP) != -1) { // Menjalankan loop simulasi dengan interval waktu tertentu
        readIR_kanan = wb_distance_sensor_get_value(sensor_kanan); // Membaca nilai sensor kanan
        readIR_kiri = wb_distance_sensor_get_value(sensor_kiri); // Membaca nilai sensor kiri

        printf("kanan 1 "); // Mencetak label untuk nilai sensor kanan
        printf("%f\n", readIR_kanan); // Mencetak nilai sensor kanan
        printf("kiri 1 "); // Mencetak label untuk nilai sensor kiri
        printf("%f\n", readIR_kiri); // Mencetak nilai sensor kiri

        // Kondisi jika sensor kiri mendeteksi objek lebih dekat dibanding sensor kanan
        if (readIR_kiri < readIR_kanan) {
            wb_motor_set_velocity(motor_kanan, 1); // Memperlambat motor kanan
            wb_motor_set_velocity(motor_kiri, 2.5); // Mempercepat motor kiri untuk berbelok ke kanan
        }

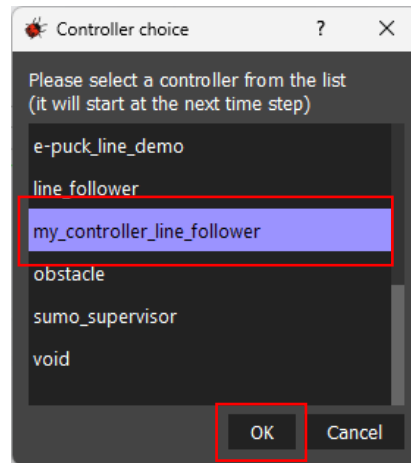
        // Kondisi jika sensor kanan mendeteksi objek lebih dekat dibanding sensor kiri
        if (readIR_kiri > readIR_kanan) {
            wb_motor_set_velocity(motor_kanan, 2.5); // Mempercepat motor kanan untuk berbelok ke kiri
            wb_motor_set_velocity(motor_kiri, 1); // Memperlambat motor kiri
        }

        wb_robot_cleanup(); // Membersihkan memori yang digunakan oleh robot
    }

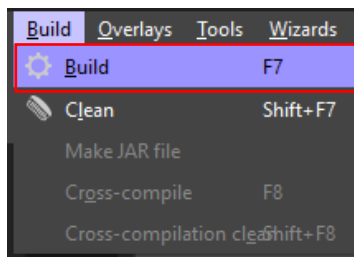
    return 0; // Mengakhiri program
}
```

8. Ubah Controller Default di Robot e-puck

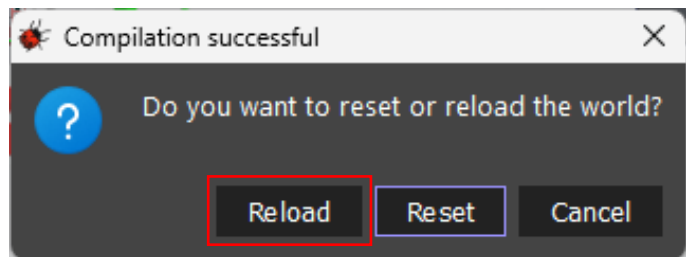
- Pilih controller yang sudah dibuat.



- Jika menggunakan bahasa C, maka harus dilakukan Build terlebih dahulu dengan memilih menu Build → Build atau dengan shortcut (F7).



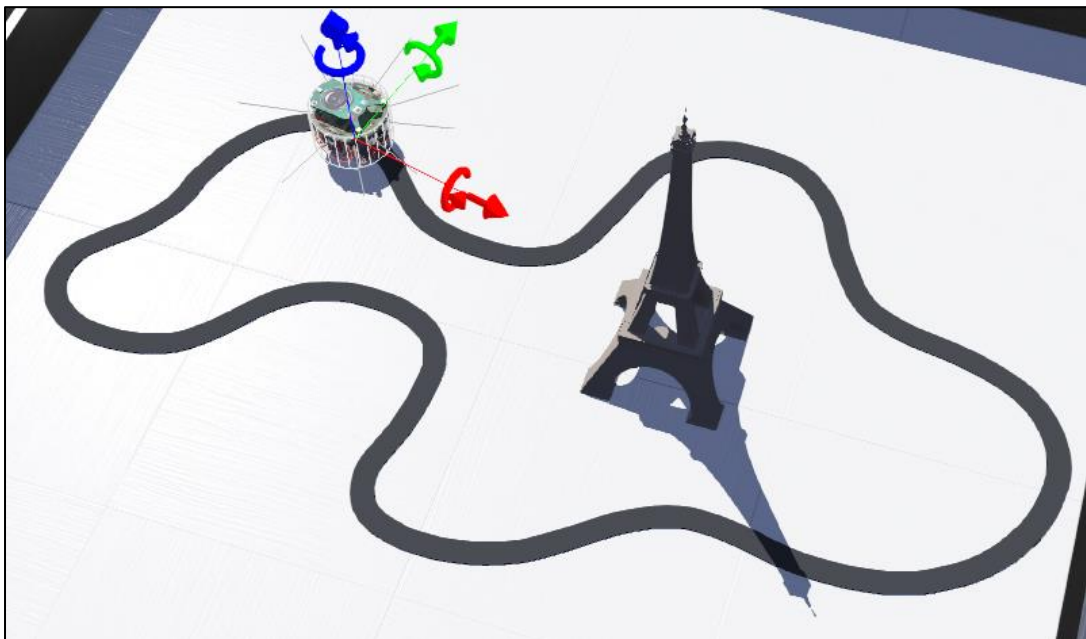
Klik Build untuk build kode yang sudah dibuat



Klik Reload untuk melanjutkan ke tahap selanjutnya.

9. Posisi dan Simulasi

- Posisikan robot e-puck di atas track sesuai jalur yang telah disiapkan.



- Simpan semua perubahan, lalu jalankan simulasi untuk menguji apakah robot dapat mengikut garis dengan benar.

4. Hasil

Pada akhir tutorial ini, robot-epuck berhasil mengikuti garis hitam pada arena putih yang telah disiapkan, menunjukkan bahwa sensor jarak berhasil mendeteksi garis dengan baik. Robot dapat mengikuti jalur secara akurat sesuai dengan program yang telah dibuat, berbelok ke kanan atau kiri tergantung pada sensor yang mendeteksi warna jalur. Kecepatan motor kanan dan kiri yang diatur secara dinamis memungkinkan robot berbelok dengan lancar, sehingga hasil akhir sesuai dengan yang diharapkan. Namun terdapat beberapa tantangan yang dihadapi, seperti sensitivitas sensor yang memerlukan penyesuaian untuk jalur yang lebih tajam atau jalur yang mengalami perubahan kontras. Secara keseluruhan, tutorial ini berhasil mencapai hasil yang diinginkan, yaitu membuat robot e-puck menjadi line follower yang dapat beroperasi dalam simulasi Webots.

5. Kesimpulan

Tutorial ini memberikan pemahaman dasar tentang pembuatan program line follower pada robot e-puck dengan menggunakan Webots dan bahasa pemrograman C. Melalui implementasi ini, kita dapat memahami konsep pengendalian motor, penggunaan sensor jarak, serta logika dasar pengaturan kecepatan yang memungkinkan robot mengikuti garis dengan baik.

Terdapat beberapa kelebihan dari penerapan robot line follower dengan robot e-puck ini yaitu memberikan pengalaman langsung dengan simulasi dan pemrograman robot, memungkinkan kita mempelajari dasar robotika dan pengaturan sensor dalam lingkungan virtual yang aman, dan mengembangkan keterampilan dalam memanfaatkan sensor untuk mendeteksi objek dan mengimplementasikan logika keputusan.

Namun, terdapat kekurangan yaitu terbatas pada simulasi di lingkungan virtual yang mungkin memiliki perbedaan dengan kondisi dunia nyata dan memerlukan penyesuaian jika ingin mengaplikasikan program ini pada hardware robotik sesungguhnya.

6. Referensi

1. Fauzansuper11. (2022, 19 Maret). *Tutorial Webots Follower Line | Bahasa Indonesia* [Video]. YouTube. https://www.youtube.com/watch?v=QCHp_DXU-4k
2. Kajal Gada. (2020, 11 Desember). *Webots Tutorial: Line Follower Robot using epuck // Controller code in Python* [Video]. YouTube. <https://www.youtube.com/watch?v=D0jhvFZJ5Ok>