

Tugas Week 7 Robotika

Nama : Widiono
NIM : 1103210060
Kelas : TK-45-G09

1. JetBot Basic Motion :

- Source code di atas adalah pengendali dasar untuk robot JetBot menggunakan Webots API. Fungsi utama dari program ini adalah menggerakkan JetBot ke arah yang berbeda dengan menggunakan kontrol motor berdasarkan kecepatan yang dikonversi. Fungsi ``convert_speed`` digunakan untuk menyesuaikan kecepatan yang diberikan dengan nilai yang sudah diatur dalam array ``converted_values``, yang membantu mengendalikan kecepatan motor dalam batas tertentu. Ada beberapa fungsi untuk mengontrol gerakan robot, seperti ``forward``, ``backward``, ``left``, dan ``right``, yang mengatur arah pergerakan dengan kecepatan yang sudah dikonversi. Fungsi ``step_forward``, ``step_backward``, ``step_left``, dan ``step_right`` digunakan untuk menggerakkan robot dalam langkah-langkah kecil ke arah yang berbeda selama waktu tertentu dan kemudian berhenti.

Di dalam ``main loop``, robot secara otomatis melakukan gerakan maju, mundur, kiri, dan kanan dengan langkah terjadwal yang diulang terus-menerus selama simulasi berjalan. Pada awal program, robot menginisialisasi perangkat motor untuk roda kiri dan kanan dengan menetapkan posisi motor menjadi "INFINITY" agar motor dapat bergerak bebas tanpa batas rotasi. Kamera juga diaktifkan pada langkah waktu yang sudah disesuaikan. Di akhir program, fungsi ``wb_robot_cleanup`` dipanggil untuk membersihkan sumber daya yang digunakan Webots. Dengan struktur ini, program memungkinkan robot untuk melakukan pola gerakan sederhana yang dapat digunakan sebagai dasar untuk pengembangan lebih lanjut seperti navigasi atau interaksi lingkungan yang lebih kompleks.

2. Jetbot_collect-data :

- Source code ini adalah pengendali untuk mengumpulkan data gambar yang akan digunakan dalam pelatihan model AI untuk menghindari tabrakan pada robot JetBot. Program ini memungkinkan pengguna untuk mengontrol JetBot secara manual dan menandai gambar sebagai "free" (aman untuk maju) atau "blocked" (harus berbelok) dengan menekan tombol "F" atau "B" pada keyboard. Gambar yang ditandai akan disimpan dalam folder "dataset/free" atau "dataset/blocked" sesuai dengan kategori yang dipilih. Setelah cukup banyak gambar dikumpulkan (minimal 20 per kategori), pengguna dapat menekan tombol "C" untuk memulai pelatihan model. Program akan menggunakan gambar-gambar yang sudah ditandai untuk melatih model AI yang nantinya disimpan sebagai file ``best_model.pth``.

Proses pelatihan model AI ini dilakukan dengan memanggil modul ``jetbot_train``, yang berisi metode ``train()`` untuk membangun model klasifikasi berdasarkan data yang telah dikumpulkan. Dengan cara ini, model dapat belajar untuk membedakan antara kondisi aman dan kondisi berbahaya di depan robot. Setelah model selesai dilatih, file ``best_model.pth`` dapat dipindahkan ke folder ``jetbot_collision_avoidance`` agar JetBot dapat menggunakan model ini untuk mendeteksi objek dan menghindari tabrakan secara otomatis dalam simulasi atau aplikasi nyata. Program ini merupakan bagian dari rangkaian proyek NVIDIA AI untuk

robotika, yang dirancang agar mudah diikuti dan diterapkan dalam lingkungan simulasi.

3. jetbot_collision_avoidance :

- Source code ini adalah pengendali untuk menghindari tabrakan pada robot JetBot menggunakan model AI yang sudah dilatih sebelumnya. Model ini dibangun berdasarkan arsitektur *ResNet18* dan disimpan dalam file `best_model_resnet18.pth`, yang di-load di awal program untuk mendeteksi kondisi “free” atau “blocked”. Pada setiap iterasi, kamera robot menangkap gambar lingkungan, yang kemudian diubah ke bentuk tensor menggunakan fungsi `preprocessCameraImage`. Gambar tersebut selanjutnya distandardisasi agar sesuai dengan nilai mean dan standar deviasi dari dataset ImageNet yang menjadi acuan model ResNet18. Setelah di-preprocess, gambar dimasukkan ke dalam model yang memberikan prediksi probabilitas untuk dua kelas: “free” dan “blocked”.

Berdasarkan probabilitas yang dihitung, JetBot akan memilih arah gerakan. Jika probabilitas terdeteksi sebagai “free” (probabilitas $< 0,5$), robot akan bergerak maju. Jika probabilitas menunjukkan “blocked” (probabilitas $\geq 0,5$), robot akan berbelok ke kiri untuk menghindari tabrakan. Pemilihan arah ini disimpan dalam variabel `direction` agar robot tidak berulang kali mengatur ulang arah ketika masih dalam status yang sama. Dengan mekanisme ini, program memungkinkan JetBot untuk menghindari tabrakan secara otomatis, mengandalkan kamera dan model prediksi yang sudah dilatih sebelumnya, sehingga membuat robot dapat bergerak mandiri dalam lingkungan yang kompleks.