

Nama : Irman Prayista

NIM : 1103210094

Kelas : TK-45-02

Hasil Analisis Simulasi

a. Gerakan Maju

Pada simulasi gerakan maju, untuk mengakses dan mengontrol perangkat robot menggunakan `robot = Robot()`. Waktu langkah simulasi dan kecepatan maksimal yang digunakan `TIME_STEP = 64` dan `MAX_SPEED = 6.28`. Setelah itu, buat variabel yang menyimpan roda kiri dan kanan robot seperti `leftMotor = robot.getDevice("left wheel motor")` dan `rightMotor = robot.getDevice("right wheel motor")`. Posisi roda kiri dan kanan robot diatur dengan tidak memiliki batasan posisi seperti `leftMotor.setPosition(float("inf"))` dan `rightMotor.setPosition(float("inf"))` sehingga akan berjalan dalam mode kecepatan bukan dalam mode posisi. Sebelum bergerak kecepatan awal diatur menjadi 0.0 pada roda kiri dan kanan robot seperti untuk memastikan robot tidak bergerak pada posisi awal. Untuk menggerakkan robot, digunakan loop `while robot.step(TIME_STEP) != -1`: yang berarti jika `TIME_STEP` tidak sama dengan -1 maka robot tidak akan berhenti dan akan menjalankan perintah `leftMotor.setVelocity(5.0)` dan `rightMotor.setVelocity(5.0)` yang fungsinya menjalankan roda kiri dan kanan dengan kecepatan 5.0.

b. Gerakan Melingkar

Pada simulasi gerakan melingkar, untuk mengakses dan mengontrol perangkat robot menggunakan `robot = Robot()`. Waktu langkah simulasi dan kecepatan maksimal yang digunakan `TIME_STEP = 64` dan `MAX_SPEED = 6.28`. Setelah itu, buat variabel yang menyimpan roda kiri dan kanan robot seperti `leftMotor = robot.getDevice("left wheel motor")` dan `rightMotor = robot.getDevice("right wheel motor")`. Posisi roda kiri dan kanan robot diatur dengan tidak memiliki batasan posisi seperti `leftMotor.setPosition(float("inf"))` dan `rightMotor.setPosition(float("inf"))` sehingga akan berjalan dalam mode kecepatan bukan dalam mode posisi. Sebelum bergerak kecepatan awal diatur menjadi 0.0 pada roda kiri dan kanan robot seperti untuk memastikan robot tidak bergerak pada posisi awal. Untuk menggerakkan robot, digunakan loop `while robot.step(TIME_STEP) != -1`: yang berarti jika `TIME_STEP` tidak sama dengan -1 maka robot tidak akan berhenti dan akan menjalankan perintah `leftMotor.setVelocity(0.5 * MAX_SPEED)` yang berfungsi untuk menggerakkan roda

kiri dengan kecepatan setengah dari kecepatan maksimal dan `rightMotor.setVelocity(MAX_SPEED)` yang berfungsi untuk menggerakkan roda kanan dengan kecepatan maksimal sehingga robot akan berjalan melingkar melawan arah jarum jam.

c. Penghentian dengan Sensor Proximity

Pada simulasi penghentian robot dengan sensor proximity, untuk mengakses dan mengontrol perangkat robot menggunakan `robot = Robot()`. Waktu langkah simulasi dan kecepatan maksimal yang digunakan `TIME_STEP = 64` dan `MAX_SPEED = 6.28`. Setelah itu, buat variabel yang menyimpan roda kiri dan kanan robot seperti `leftMotor = robot.getDevice("left wheel motor")` dan `rightMotor = robot.getDevice("right wheel motor")`. Posisi roda kiri dan kanan robot diatur dengan tidak memiliki batasan posisi seperti `leftMotor.setPosition(float("inf"))` dan `rightMotor.setPosition(float("inf"))` sehingga akan berjalan dalam mode kecepatan bukan dalam mode posisi. Sebelum bergerak kecepatan awal diatur menjadi 0.0 pada roda kiri dan kanan robot seperti untuk memastikan robot tidak bergerak pada posisi awal. Lalu definisikan sensor proximity menggunakan pengulangan `for` dengan `range(8)` karena robot yang digunakan pada simulasi ini memiliki 8 sensor dan hasil pengulangan akan disimpan pada list `proximity_sensor`. Untuk membuat robot berhenti digunakan fungsi `stop_robot` dengan kecepatan roda kiri dan kanan sama dengan 0. Selain itu, agar robot tetap berjalan ketika tidak ada objek digunakan fungsi `move_forward` dengan parameter `speed` pada roda kiri dan kanan robot. Untuk menggerakkan robot, digunakan loop `while robot.step(TIME_STEP) != -1`: yang berarti jika `TIME_STEP` tidak sama dengan -1 maka robot tidak akan berhenti dan dibuat variabel `front_left_value` dan `front_right_value` untuk menyimpan nilai dari sensor lalu ditampilkan. Setelah itu, dibuat kondisi jika sensor `front_left_value` mendapatkan nilai lebih dari 80 atau sensor `front_right_value` mendapatkan nilai lebih dari 80 maka robot berhenti dengan memanggil fungsi `stop_robot`. Jika diluar perintah tersebut maka robot akan tetap jalan dengan memanggil fungsi `move_forward(5)` dengan kecepatan sama dengan 5.