

LAPORAN UJIAN TENGAH SEMESTER ROBOTIKA

“Wall Following Robot With Camera”

Ditujukan Untuk Memenuhi Ujian Akhir Semester Robotika



Disusun Oleh:

April Hamonangan Marbun (1103202039)

Universitas Telkom

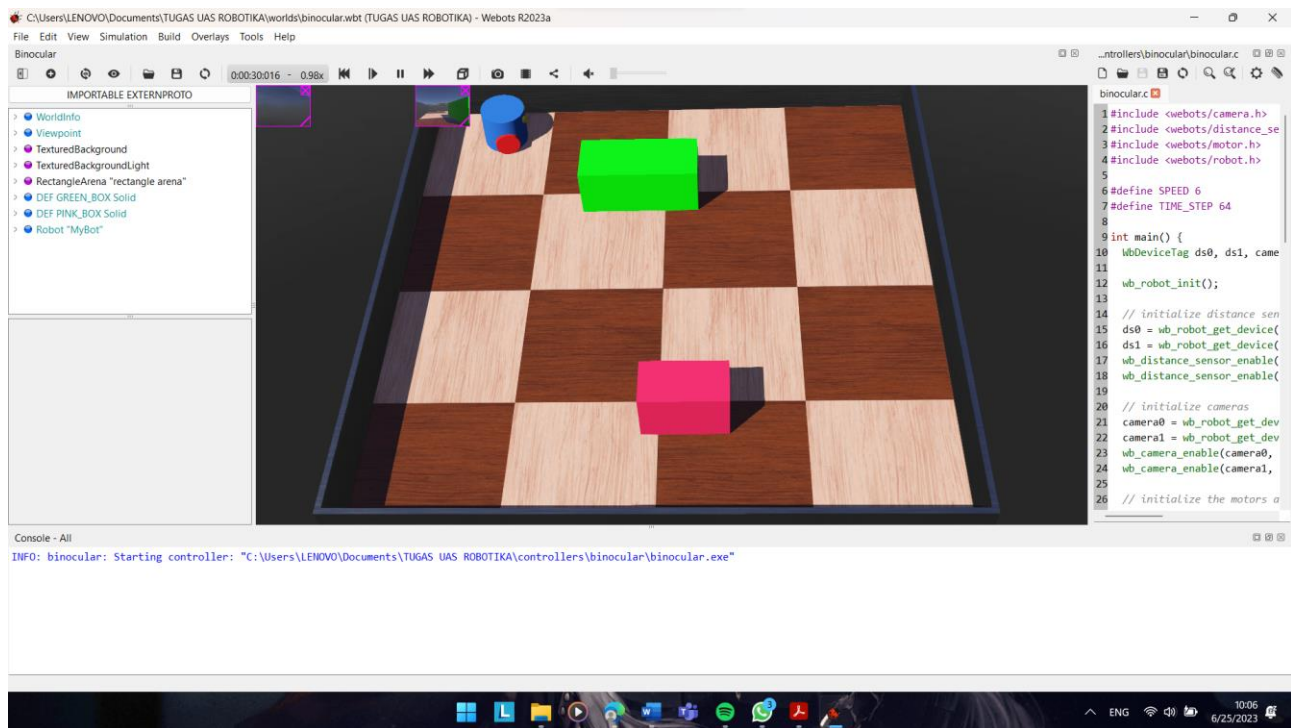
Bandung

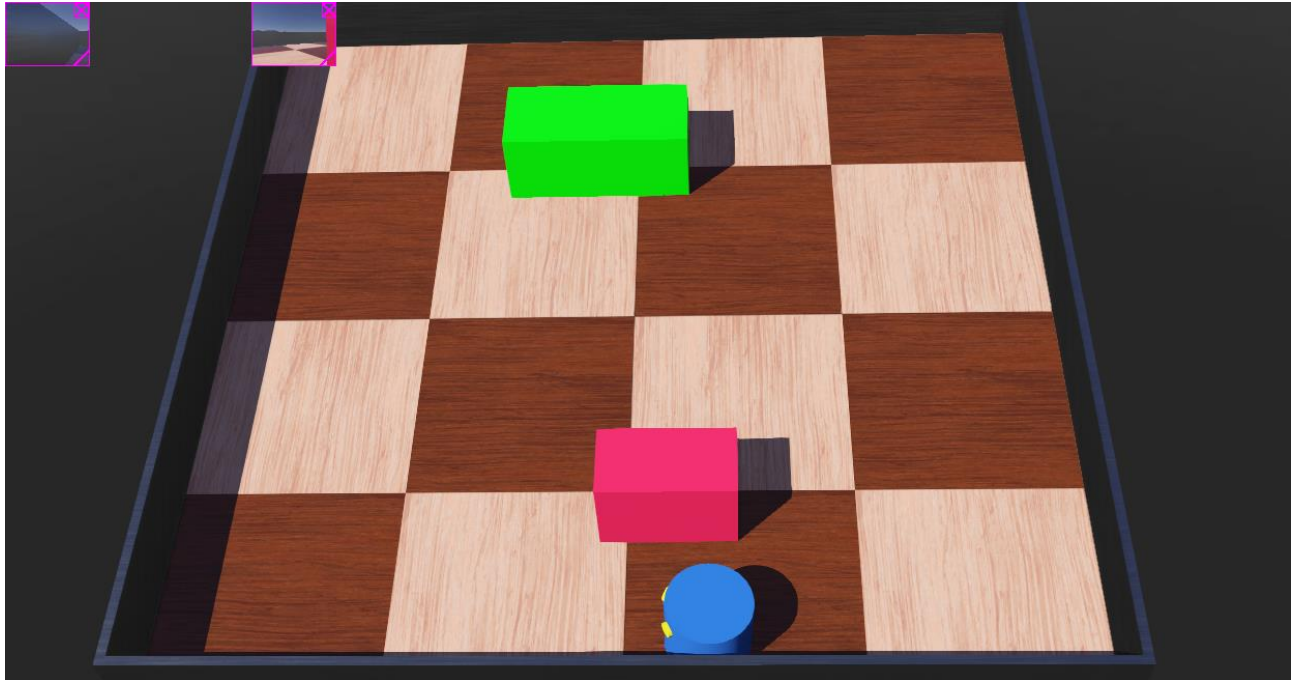
2023

1. Pendahuluan

Laporan teknis ini membahas implementasi robot pengikut dinding menggunakan perangkat lunak simulasi robot Webots. Konsep pengikut dinding menjadi penting dalam robotika karena memungkinkan robot untuk bergerak secara otonom di sekitar lingkungan dengan mengikuti suatu objek, seperti dinding. Robot pengikut dinding dapat digunakan dalam berbagai aplikasi, mulai dari kebersihan lantai otomatis hingga navigasi robot di sekitar rumah atau gudang. Dalam laporan ini, kita akan melihat implementasi sebuah robot pengikut dinding yang menggunakan sensor jarak dan kamera dalam lingkungan simulasi Webots. Sensor jarak digunakan untuk mendeteksi jarak antara robot dan dinding, sementara kamera dapat memberikan informasi visual tentang lingkungan sekitar robot. Kombinasi antara sensor jarak dan kamera memungkinkan robot untuk mengambil keputusan berdasarkan informasi yang diterima dan mengikuti dinding dengan kecepatan dan arah yang sesuai.

Melalui laporan ini, kita akan menjelaskan kode yang telah disediakan, langkah-langkah yang diambil oleh robot dalam mengikuti dinding, dan fungsionalitas masing-masing komponen dalam kode. Laporan ini juga akan memberikan pemahaman yang lebih dalam tentang bagaimana robot tersebut berinteraksi dengan lingkungan simulasi di Webots, serta memberikan dasar bagi pengembang untuk mengembangkan dan mengadaptasi perilaku robot ini sesuai dengan kebutuhan atau tujuan tertentu.





2. Tinjauan Kode

Kode yang diberikan ditulis dalam bahasa C dan terdiri dari fungsi utama yang melakukan inisialisasi perangkat dan mengimplementasikan perilaku pengikut dinding. Kode menggunakan API Webots untuk mengakses komponen robot seperti sensor jarak, kamera, dan motor.

```
#include <webots/camera.h>
#include <webots/distance_sensor.h>
#include <webots/motor.h>
#include <webots/robot.h>

#define SPEED 6
#define TIME_STEP 64

int main() {
    WbDeviceTag ds0, ds1, camera0, camera1, left_motor, right_motor;

    wb_robot_init();

    ds0 = wb_robot_get_device("ds0");
    ds1 = wb_robot_get_device("ds1");
    wb_distance_sensor_enable(ds0, TIME_STEP);
    wb_distance_sensor_enable(ds1, TIME_STEP);

    camera0 = wb_robot_get_device("camera0");
    camera1 = wb_robot_get_device("camera1");
    wb_camera_enable(camera0, 2 * TIME_STEP);
    wb_camera_enable(camera1, 2 * TIME_STEP);
    left_motor = wb_robot_get_device("left wheel motor");
    right_motor = wb_robot_get_device("right wheel motor");
```

```

wb_motor_set_position(left_motor, INFINITY);
wb_motor_set_position(right_motor, INFINITY);
wb_motor_set_velocity(left_motor, 0.0);
wb_motor_set_velocity(right_motor, 0.0);

while (wb_robot_step(TIME_STEP) != -1) {

    wb_camera_get_image(camera0);
    wb_camera_get_image(camera1);

    double ds0_value = wb_distance_sensor_get_value(ds0);
    double ds1_value = wb_distance_sensor_get_value(ds1);

    double left_speed, right_speed;
    if (ds1_value > 500) {
        if (ds0_value > 200) {
            left_speed = -SPEED;
            right_speed = -SPEED / 2;
        } else {
            left_speed = -ds1_value / 100;
            right_speed = (ds0_value / 100) + 0.5;
        }
    } else if (ds0_value > 500) {
        left_speed = (ds1_value / 100) + 0.5;
        right_speed = -ds0_value / 100;
    } else {
        left_speed = SPEED;
        right_speed = SPEED;
    }
    wb_motor_set_velocity(left_motor, left_speed);
    wb_motor_set_velocity(right_motor, right_speed);
}
return 0;
}

```

2.1. Insialisasi

Kode diawali dengan melakukan inisialisasi lingkungan Webots dan mendapatkan referensi perangkat yang diperlukan. Sensor jarak (ds0 dan ds1), kamera (camera0 dan camera1), dan motor (left_motor dan right_motor) diinisialisasi dan diaktifkan untuk penggunaan selanjutnya.

2.2. Loop Utama

Loop utama kode dieksekusi secara terus menerus sampai simulasi berakhir. Di dalam loop, langkah-langkah berikut dilakukan:

2.2.1. Kamera

Pada setiap iterasi loop utama, fungsi **wb_camera_get_image** digunakan untuk memperbarui gambar dari kedua kamera (**camera0** dan **camera1**). Meskipun gambar yang diambil tidak digunakan dalam kode yang diberikan, langkah ini memastikan bahwa kamera-kamera tersebut selalu memiliki gambar terbaru. Pembaruan gambar kamera ini penting jika kode dikembangkan lebih lanjut untuk memanfaatkan informasi visual dari sekitar robot.

2.2.2. Sensor Jarak

Nilai sensor jarak diperoleh menggunakan fungsi **wb_distance_sensor_get_value**. Nilai dari ds0 dan ds1 disimpan dalam variabel **ds0_value** dan **ds1_value** secara berturut-turut.

2.2.3. Perilaku Wall Follower

Berdasarkan pembacaan sensor jarak, robot menentukan perilakunya dalam mengikuti dinding. Skenario berikut dipertimbangkan:

a. Kedua sensor jarak mendeteksi objek (dinding):

- Jika $ds0_value > 200$, robot menghadapi dinding dan perlu bergerak mundur. Kecepatan motor kiri dan kanan diatur untuk bergerak mundur.
- Jika $ds0_value \leq 200$, robot perlu berbelok proporsional berdasarkan nilai sensor. Kecepatan motor kiri diatur menjadi $-ds1_value / 100$, dan kecepatan motor kanan diatur menjadi $(ds0_value / 100) + 0.5$.

b. Hanya ds0 yang mendeteksi objek (dinding):

- Robot perlu berbelok ke kiri saat mengikuti dinding. Kecepatan motor kiri diatur menjadi $(ds1_value / 100) + 0.5$, dan kecepatan motor kanan diatur menjadi $-ds0_value / 100$.

c. Tidak ada objek yang terdeteksi:

- Robot dapat bergerak maju dengan kecepatan maksimal. Kecepatan motor kiri dan kanan diatur menjadi SPEED.

2.2.4. Kontrol Motor

Nilai kecepatan yang dihitung, yaitu **left_speed** dan **right_speed**, diteruskan ke motor menggunakan fungsi **wb_motor_set_velocity**. Ini mengendalikan gerakan robot.

3. Kesimpulan

Kode yang disediakan mengimplementasikan perilaku pengikut dinding untuk robot dalam lingkungan simulasi Webots. Dengan memanfaatkan sensor jarak dan kamera, robot dapat mendeteksi dan mengikuti dinding, serta menyesuaikan kecepatan dan arahnya berdasarkan pembacaan sensor. Laporan teknis ini memberikan gambaran umum tentang kode dan fungsionalitasnya, sehingga memudahkan pemahaman terhadap perilaku robot dalam simulasi. Modifikasi dan peningkatan lebih lanjut dapat dilakukan untuk menyesuaikan perilaku robot dengan skenario atau persyaratan tertentu.