

LAPORAN UJIAN TENGAH SEMESTER ROBOTIKA

“Pengendalian Robot Menggunakan Keyboard dan Sensor Jarak”

Ditujukan Untuk Memenuhi Ujian Tengah Semester Robotika



**Universitas
Telkom**

Disusun Oleh :

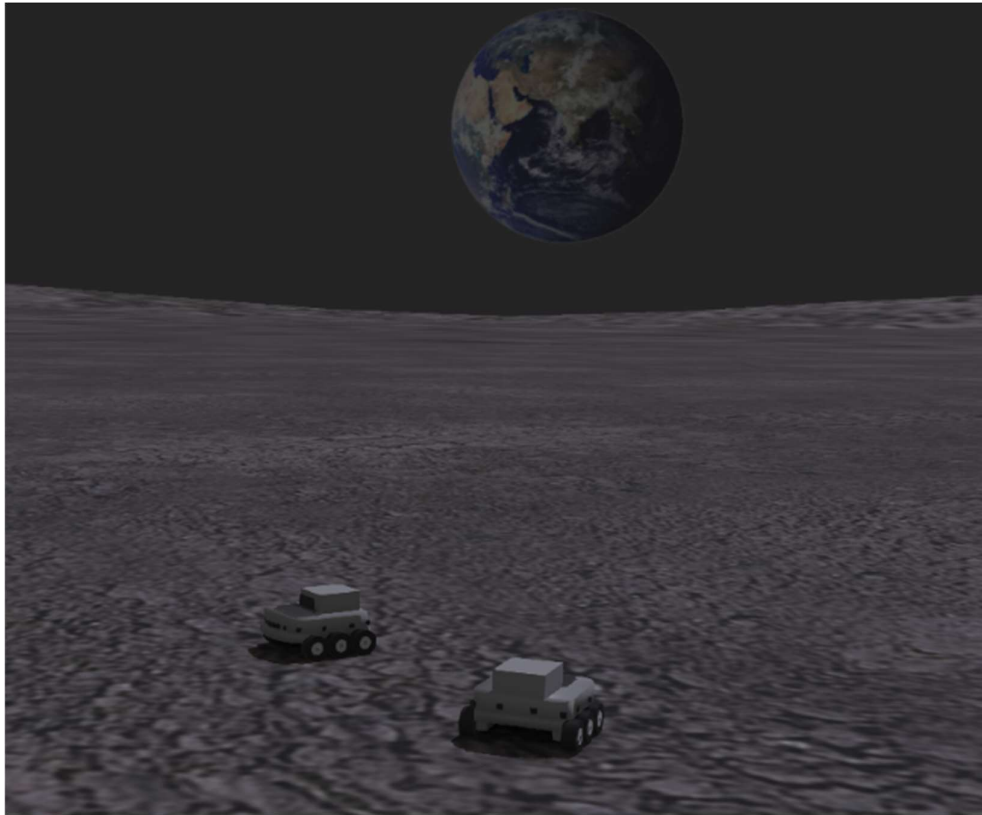
Muhammad Fharist (1103202022)

PROGRAM STUDI TEKNIK KOMPUTER

FAKULTAS TEKNIK ELEKTRO

UNIVERSITAS TELKOM

2023



PENDAHULUAN

Program ini merupakan salah satu demos yang ada di web cyberbotics yang bernama moon.wbt. Program ini mengontrol gerakan robot menggunakan sensor jarak inframerah dan tombol keyboard pada Webots. Robot dapat bergerak maju, mundur, dan berbelok dengan kecepatan yang bervariasi tergantung pada nilai yang diberikan oleh sensor jarak dan tombol keyboard yang ditekan.

Berikut codingan dari program tersebut :

```
#include <stdio.h>
#include <string.h>
#include <webots/distance_sensor.h>
#include <webots/keyboard.h>
#include <webots/motor.h>
#include <webots/robot.h>
```

```

#define TIME_STEP 64

int main() {
    // Inisialisasi Variabel
    WbDeviceTag ir[16];
    WbDeviceTag left_motor, right_motor;
    int manual_control = 0;
    int i, j;
    const char *robot_name;
    double speed[2];
    double sensor_value[16];
    double kUnit = 8.24;
    double matrix[2][16] = {{3, -18, -15, -15, -5, -5, 4, 4, 6, 6, 4, 4, 4, 4, 3, -18},
                             {-18, 3, 4, 4, 4, 4, 6, 6, 4, 4, -5, -5, -15, -15, -18, 3}};

    // Inisialisasi simulasi robot
    wb_robot_init();
    robot_name = wb_robot_get_name();

    // Mengambil device sensor jarak
    if (strcmp(robot_name, "koala", 5) == 0) {
        for (i = 0; i < 16; i++) {
            char sensor_name[16];
            sprintf(sensor_name, "ds%d", i);
            ir[i] = wb_robot_get_device(sensor_name);
        }
    }

    // Mengaktifkan sensor jarak
    for (i = 0; i < 16; i++)
        wb_distance_sensor_enable(ir[i], TIME_STEP);

    // Mengambil device motor dan mengaktifkan motor

```

```

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);

// Mengaktifkan Keyboard
wb_keyboard_enable(TIME_STEP);
printf("You can drive this robot by using the arrow keys of your keyboard\n");

//Melakukan looping terus menerus hingga simulasi berakhir
while (wb_robot_step(TIME_STEP) != -1) {
    for (i = 0; i < 16; i++)
        sensor_value[i] = wb_distance_sensor_get_value(ir[i]);

    for (i = 0; i < 2; i++) {
        speed[i] = 80;
        for (j = 0; j < 16; j++)
            speed[i] += kUnit * matrix[i][j] * sensor_value[j] / 200.0;
        speed[i] /= 50;
    }

    // Mendapatkan input keyboard dan mengatur kecepatan motor jika mode manual
    diaktifkan
    const int key = wb_keyboard_get_key();

    if ((key >= 0) && !manual_control)
        manual_control = 1;

    if (manual_control) {
        switch (key) {
            case WB_KEYBOARD_DOWN:

```

```
    speed[0] = -1;
    speed[1] = -1;
    break;
case WB_KEYBOARD_UP:
    speed[0] = 1;
    speed[1] = 1;
    break;
case WB_KEYBOARD_LEFT:
    speed[0] = -1;
    speed[1] = 1;
    break;
case WB_KEYBOARD_RIGHT:
    speed[0] = 1;
    speed[1] = -1;
    break;
case 0:
    speed[0] = 0;
    speed[1] = 0;
}
}
// Mengatur kecepatan motor
wb_motor_set_velocity(left_motor, speed[0]);
wb_motor_set_velocity(right_motor, speed[1]);
}

wb_robot_cleanup();

return 0;
}
```

DETAIL

- **Library dan Konstanta**

Program menggunakan library `stdio.h`; `string.h`; `webots/distance_sensor.h`; `webots/keyboard.h`; `webots/motor.h`; dan `webots/robot.h`. Selain itu, terdapat konstanta `TIME_STEP` dengan nilai 64 yang menentukan waktu simulasi dalam milidetik.

- **Variabel**

- `ir[16]` : array dari 16 sensor jarak inframerah
- `left_motor` : motor roda kiri
- `right_motor` : motor roda kanan
- `manual_control` : variabel untuk menentukan apakah robot dikendalikan secara manual atau tidak
- `i, j` : variabel untuk iterasi dalam loop
- `robot_name` : nama robot
- `speed[2]` : array yang menyimpan nilai kecepatan motor kiri dan kanan
- `sensor_value[16]` : array yang menyimpan nilai sensor jarak inframerah
- `kUnit` : nilai konstanta yang digunakan dalam perhitungan kecepatan robot
- `matrix[2][16]` : array yang menyimpan matriks yang digunakan dalam perhitungan kecepatan robot

Inisialisasi Robot

Program menginisialisasi robot dengan memanggil fungsi `wb_robot_init()` dan mendapatkan nama robot dengan fungsi `wb_robot_get_name()`. Jika nama robot adalah "koala", program akan mendapatkan 16 device tag dari sensor jarak inframerah dengan menggunakan loop `for`. Selain itu, program mengaktifkan sensor jarak inframerah dengan memanggil fungsi `wb_distance_sensor_enable()`. Program juga mendapatkan device tag dari motor kiri dan kanan dengan menggunakan fungsi `wb_robot_get_device()`, dan mengaktifkan motor dengan memanggil fungsi `wb_motor_set_position()` dan `wb_motor_set_velocity()`.

Loop Utama

Loop utama program akan dijalankan selama simulasi berjalan dan tidak ada kesalahan. Dalam loop, program akan memperoleh nilai sensor jarak inframerah dengan memanggil fungsi `wb_distance_sensor_get_value()`. Selanjutnya, program akan menghitung kecepatan

robot dengan menggunakan matriks matrix dan nilai sensor jarak inframerah. Jika robot dikendalikan secara manual, program akan menentukan kecepatan berdasarkan tombol keyboard yang ditekan. Setelah kecepatan dihitung, program akan mengatur kecepatan motor kiri dan kanan dengan memanggil fungsi `wb_motor_set_velocity()`.

Membersihkan Robot

Setelah simulasi selesai, program akan membersihkan robot dengan memanggil fungsi `wb_robot_cleanup()`.

Kesimpulan

Berdasarkan analisis terhadap kode program di atas, dapat disimpulkan bahwa program tersebut adalah sebuah kontroler untuk robot yang menggunakan sensor jarak inframerah untuk menghindari penghalang. Program tersebut menggunakan matriks sebagai faktor pembobot untuk setiap sensor, sehingga robot dapat menghasilkan kecepatan yang tepat untuk menghindari penghalang. Program tersebut juga memungkinkan pengendalian manual melalui keyboard, yang memungkinkan pengguna untuk mengendalikan kecepatan robot. Program tersebut menggunakan library Webots untuk mengakses sensor, motor, dan keyboard, dan juga mengatur waktu siklus utama dengan `TIME_STEP`. Program ini dapat digunakan sebagai dasar untuk pengembangan kontroler robot yang lebih kompleks.