

## TUGAS ROBOTIKA MINGGU 10

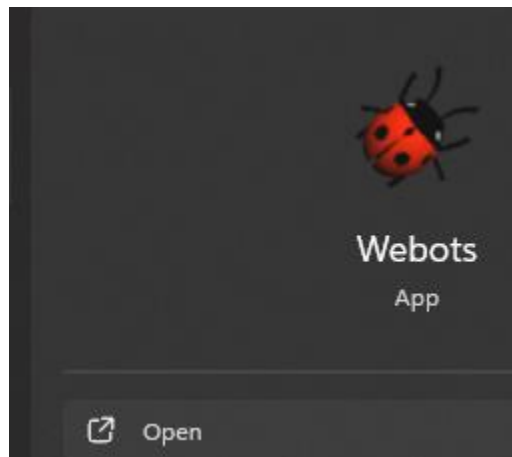
Nama : Bagas Fabian Maulana

NIM : 1103184180

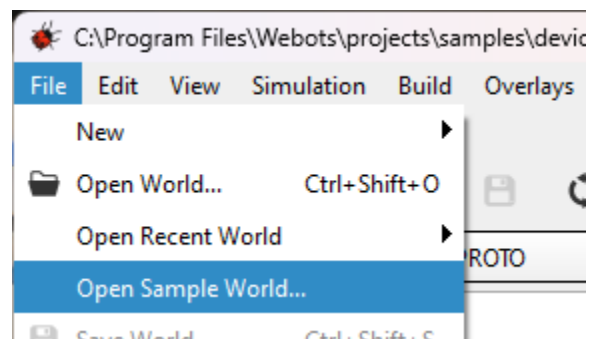
Demo Kamera deteksi warna dengan Webots

Langkah-langkah :

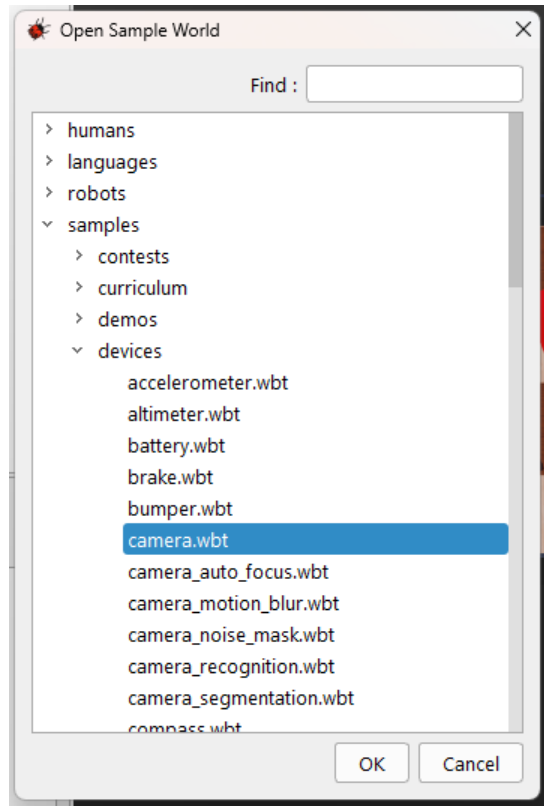
1. Buka webots dari start menu.



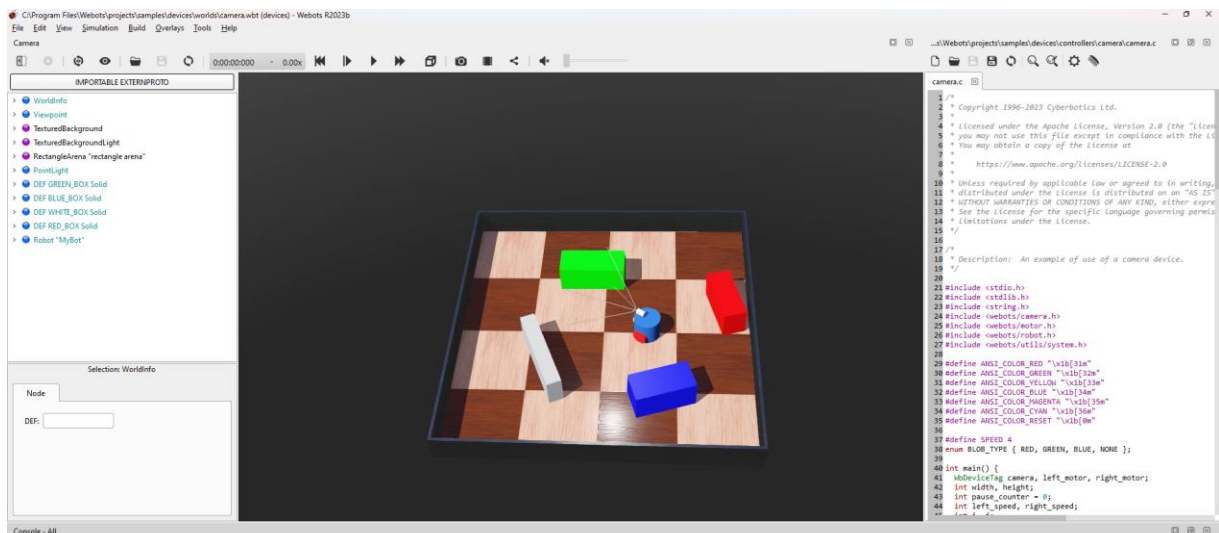
2. Klik File>Open sample world



3. Klik samples>devices>camera.wbt. Lalu klik OK.



4. Maka akan terbuka sample world dengan robot yang terintegrasi dengan kamera dan 4 buah balok dengan warna silver, hijau, biru, dan merah.



5. Maka akan otomatis terbuka text editor pada panel sebelah kanan webots. Disitu berisi source code yang digunakan untuk pengoperasian robot dengan kameranya. Berikut adalah penjelasan dari source code yang disediakan

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <webots/camera.h>
#include <webots/motor.h>
#include <webots/robot.h>
#include <webots/utils/system.h>
```

Kode diatas digunakan untuk inisialisasi library yang akan digunakan. Library mencakup library camera, motor, robot, system dari webots.

```
#define ANSI_COLOR_RED "\x1b[31m"
#define ANSI_COLOR_GREEN "\x1b[32m"
#define ANSI_COLOR_YELLOW "\x1b[33m"
#define ANSI_COLOR_BLUE "\x1b[34m"
#define ANSI_COLOR_MAGENTA "\x1b[35m"
#define ANSI_COLOR_CYAN "\x1b[36m"
#define ANSI_COLOR_RESET "\x1b[0m"
```

Kode diatas untuk mendefinisikan warna yang ditemukan oleh kamera nantinya

```
int main() {
    WbDeviceTag camera, left_motor, right_motor;
    int width, height;
    int pause_counter = 0;
    int left_speed, right_speed;
    int i, j;
    int red, blue, green;
    const char *color_names[3] = {"red", "green", "blue"};
    const char *ansi_colors[3] = {ANSI_COLOR_RED, ANSI_COLOR_GREEN,
ANSI_COLOR_BLUE};
    const char *filenames[3] = {"red_blob.png", "green_blob.png", "blue_blob.png"};
    enum BLOB_TYPE current_blob;
```

```
wb_robot_init();  
const int time_step = wb_robot_get_basic_time_step();
```

kode diatas adalah inisialisai variable yang akan digunakan untuk program nanti.

```
camera = wb_robot_get_device("camera");  
wb_camera_enable(camera, time_step);  
width = wb_camera_get_width(camera);  
height = wb_camera_get_height(camera);
```

kode diatas berguna untuk menghidupkan kamera dan mendapatkan info dari ukuran dan tinggi sensor kamera

```
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);
```

kode diatas adalah inisialisasi robot beserta motornya. Di set agar berputar terus-terusan.

```
while (wb_robot_step(time_step) != -1) {  
    const unsigned char *image = wb_camera_get_image(camera);  
  
    if (pause_counter > 0)  
        pause_counter--;  
  
    if (pause_counter > 640 / time_step) {  
        left_speed = 0;  
        right_speed = 0;  
    }  
    else if (pause_counter > 0) {  
        left_speed = -SPEED;  
        right_speed = SPEED;  
    }  
}
```

```

else if (!image) { // image may be NULL if Robot.synchronization is FALSE
    left_speed = 0;
    right_speed = 0;
} else { // pause_counter == 0
    /* Reset the sums */
    red = 0;
    green = 0;
    blue = 0;

```

kode diatas berguna untuk menentukan kapan robot akan berhenti berputar. Perulangan ini juga memastikan agar kamera tidak memutar dibalok yang sama terus.

```

for (i = width / 3; i < 2 * width / 3; i++) {
    for (j = height / 2; j < 3 * height / 4; j++) {
        red += wb_camera_image_get_red(image, width, i, j);
        blue += wb_camera_image_get_blue(image, width, i, j);
        green += wb_camera_image_get_green(image, width, i, j);
    }
}
/* If a component is much more represented than the other ones,
 * a blob is detected
 */
if ((red > 3 * green) && (red > 3 * blue))
    current_blob = RED;
else if ((green > 3 * red) && (green > 3 * blue))
    current_blob = GREEN;
else if ((blue > 3 * red) && (blue > 3 * green))
    current_blob = BLUE;
else
    current_blob = NONE;
if (current_blob == NONE) {
    left_speed = -SPEED;
    right_speed = SPEED;
}
else {
    left_speed = 0;
    right_speed = 0;

```

```

        printf("Looks like I found a %s%s%s blob.\n", ansi_colors[current_blob],
color_names[current_blob], ANSI_COLOR_RESET);
        char *filepath;
#ifdef _WIN32
        const char *user_directory =
wbu_system_short_path(wbu_system_getenv("USERPROFILE"));
        filepath = (char *)malloc(strlen(user_directory) + 16);
        strcpy(filepath, user_directory);
        strcat(filepath, "\\");
#else
        const char *user_directory = wbu_system_getenv("HOME");
        filepath = (char *)malloc(strlen(user_directory) + 16);
        strcpy(filepath, user_directory);
        strcat(filepath, "/");
#endif
        strcat(filepath, filenames[current_blob]);
        wb_camera_save_image(camera, filepath, 100);
        free(filepath);
        pause_counter = 1280 / time_step;
    }
}

/* Set the motor speeds. */
wb_motor_set_velocity(left_motor, left_speed);
wb_motor_set_velocity(right_motor, right_speed);
}

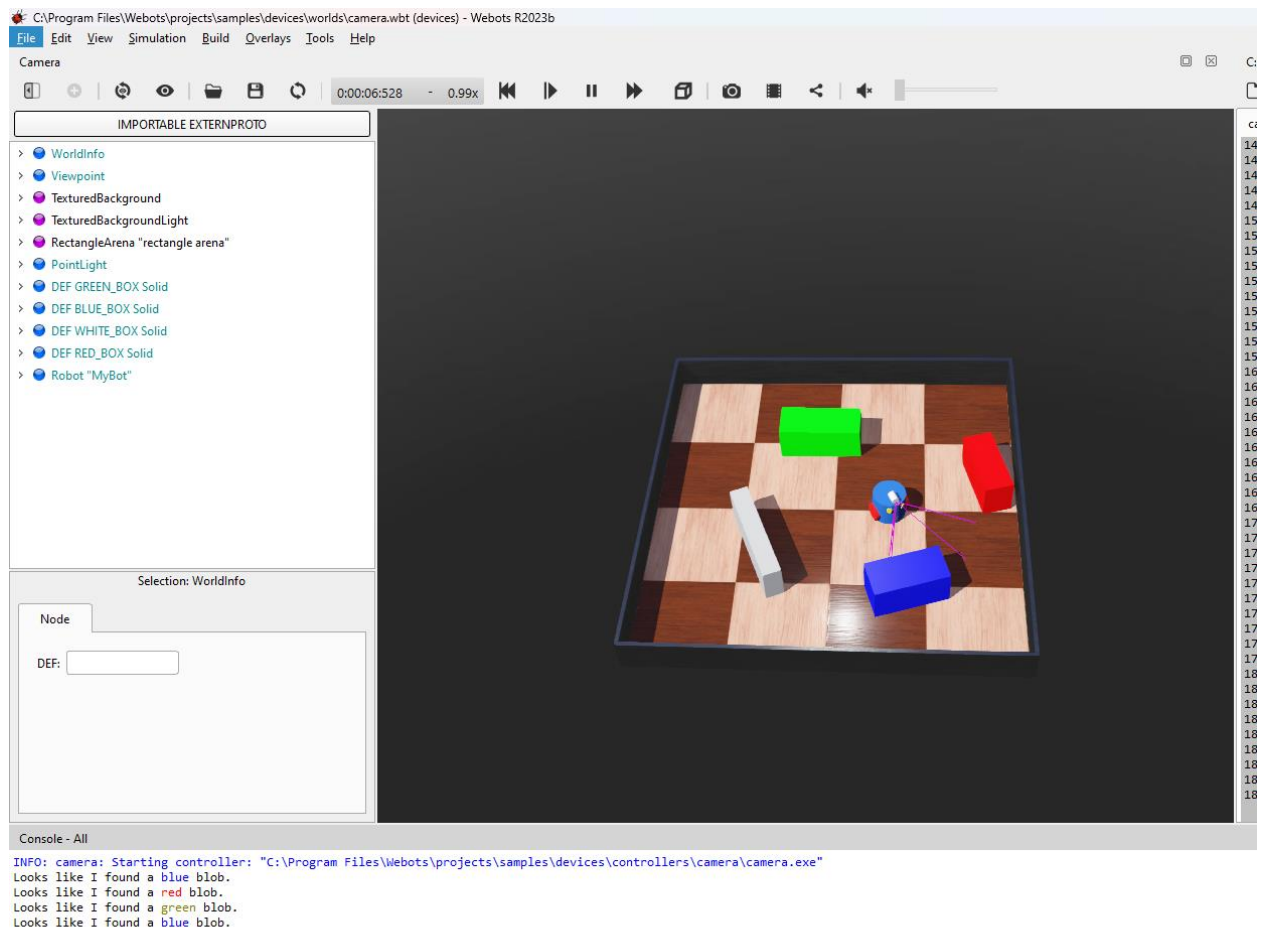
wb_robot_cleanup();

return 0;
}

```

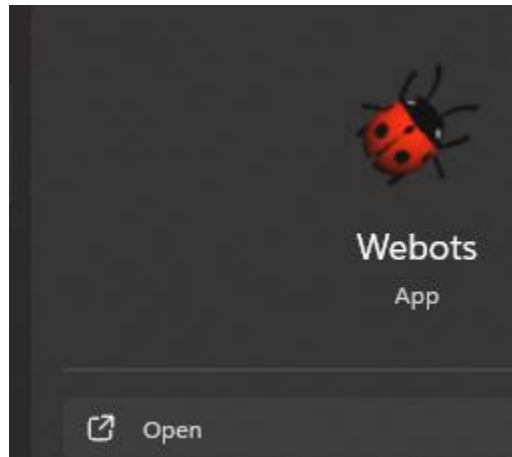
Kode ini adalah fungsi utama dari pendeteksian warna yang dilakukan oleh kamera. Kamera akan mendeteksi balok yang ada didepannya dan merubah variable `current_blob` sesuai dengan yang dilihat oleh kamera. Dan juga terdapat kode jika tidak ditemukan balok, maka robot akan terus berputar. Ketika balok ditemukan, robot akan berhenti, mengambil gambar, memberi output warna yang ditemukan, dan akan melanjutkan putaran Kembali.

6. Ketika dijalankan simulasinya, kita bisa lihat di console apa yang dilihat oleh kamera

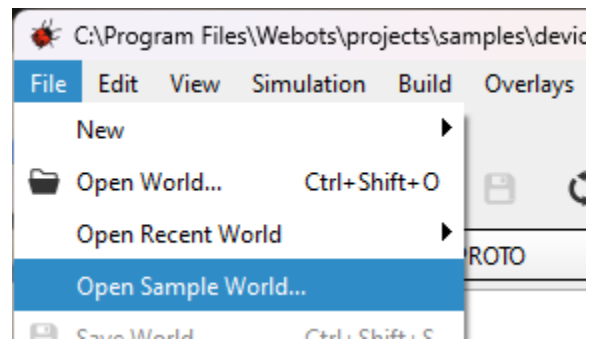


## Demo kamera autofocus

1. Buka webots dari start menu.

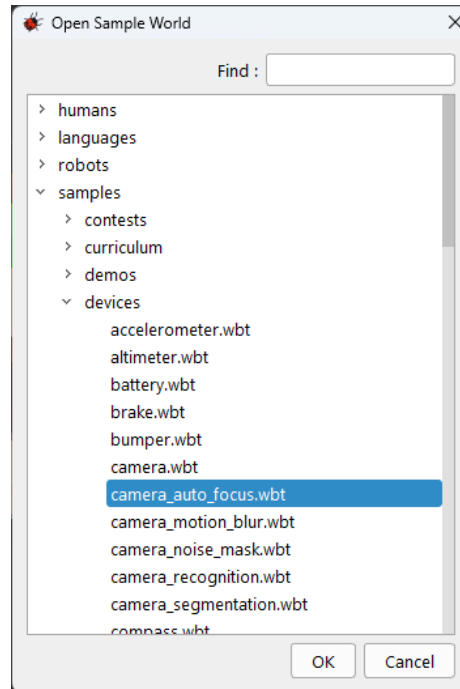


2. Klik File>Open sample world

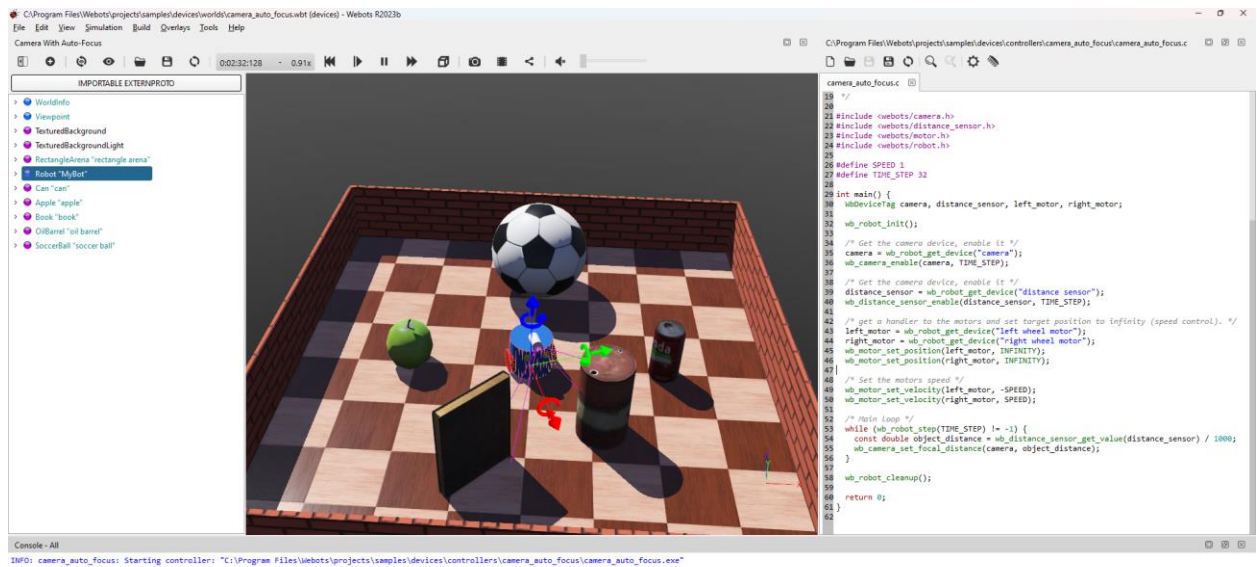




3. Klik samples>devices>camera\_autofocus.wbt. Lalu klik OK.



4. Maka akan terbuka sample world dengan robot yang terintegrasi dengan kamera dan 5 buah object.



5. Maka akan otomatis terbuka text editor pada panel sebelah kanan webots. Disitu berisi source code yang digunakan untuk pengoperasian robot dengan kameranya. Berikut adalah penjelasan dari source code yang disediakan

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

```
#define SPEED 1
#define TIME_STEP 32
```

Kode diatas melakukan inisialisasi library dan melakukan definisi untuk variable speed dan time\_step 32

```
int main() {
  WbDeviceTag camera, distance_sensor, left_motor, right_motor;
  wb_robot_init();
  camera = wb_robot_get_device("camera");
  wb_camera_enable(camera, TIME_STEP);
  distance_sensor = wb_robot_get_device("distance sensor");
  wb_distance_sensor_enable(distance_sensor, TIME_STEP);
```

kode diatas dilakukan untuk inisialisasi kamera dan motor dari robot. dan mengambil data sensor jarak yang ada di kamera

```
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, -SPEED);
wb_motor_set_velocity(right_motor, SPEED);
```

Kode ini berguna untuk menggerakkan motor robot agar terus berputar dengan kecepatan konstan.

```

while (wb_robot_step(TIME_STEP) != -1) {
    const double object_distance = wb_distance_sensor_get_value(distance_sensor)
/ 1000;
    wb_camera_set_focal_distance(camera, object_distance);
}

wb_robot_cleanup();
return 0;
}

```

Kode ini berguna untuk menghitung jarak antara kamera dengan benda yang didepan kamera dari sensor jarak kamera, dan mengatur jarak fokal kamera berdasarkan hasil perhitungan jarak antara benda dengan kamera.

6. Bila menjalankan simulasi, kita bisa melihat robot akan berjalan terus menerus.

