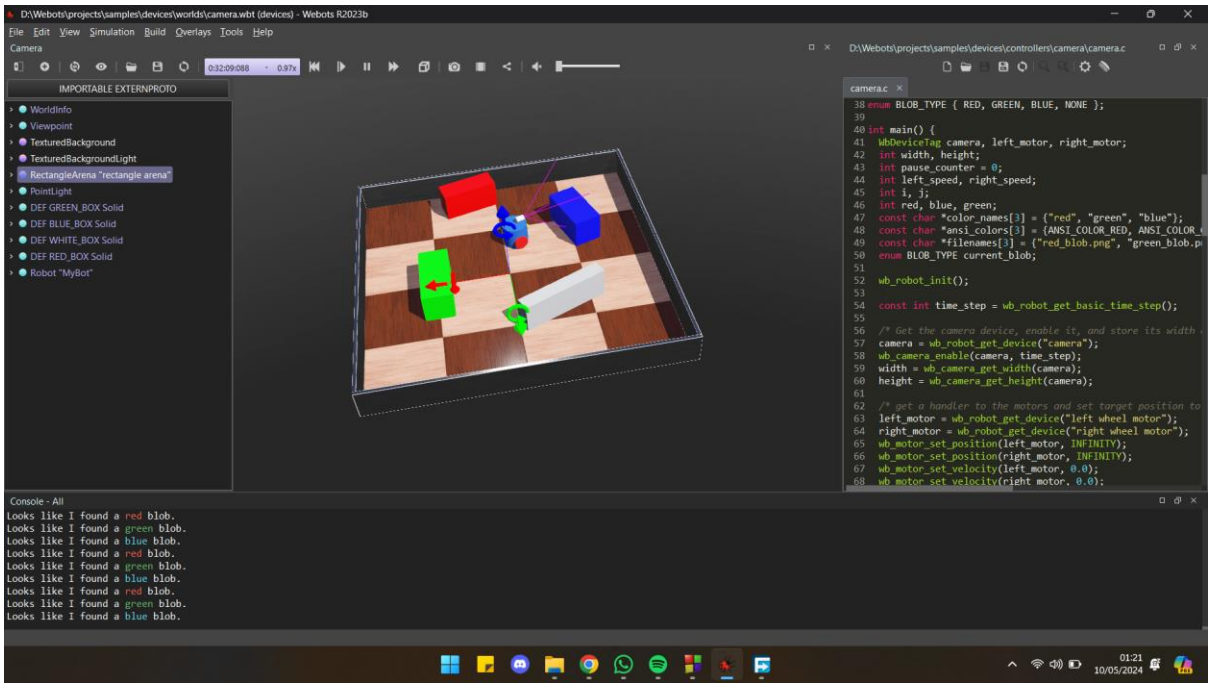
Nama : Muhammad Farrel Ahadi Tama
NIM : 1103210177
Kelas : TK-45-01

Camera robot untuk mendeteksi blob warna (Merah, Hijau, dan Biru)



Gambar diatas Camera Robot untuk Mendeteksi Blob Warna sebuah sistem yang menggunakan kamera untuk mengambil gambar dari lingkungan sekitarnya dan kemudian menganalisis gambar tersebut untuk mendeteksi dan membedakan objek berdasarkan warnanya. Sistem ini dapat digunakan untuk mendeteksi blob (gumpalan) warna tertentu, seperti merah, hijau, dan biru.

```
#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>
```

Pada bagian ini untuk mengakses fungsi-fungsi webots dan beberapa fungsi bahasa c

```
#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"
```

Pada bagian ini untuk mendefinisikan ANSI konstanta yang ada dalam kodingan

```
#define SPEED 4
enum BLOB_TYPE { RED, GREEN, BLUE, NONE };
```

// Di sini, data kamera, motor kiri dan kanan, lebar dan tinggi gambar, penghitung jeda, kecepatan motor, variabel untuk iterasi, serta variabel untuk menyimpan data warna dan blob yang terdeteksi lalu dideklarasikan pada perangkat.

```
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;
```

//Fungsi wb_robot_init() dipanggil untuk menginisialisasi lingkungan simulasi. Lebar dan tinggi citra diperoleh dari kamera. Motor kiri dan kanan diinisialisasi dan diatur ke kecepatan nol dengan menggunakan fungsi-fungsi yang disediakan oleh Webots.

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

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

  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);

  /* Main loop */
  while (wb_robot_step(time_step) != -1) {
```

// Loop utama di sini. Ini akan terus berjalan selama simulasi berlangsung.

```
    /* Get the new camera values */
    const unsigned char *image = wb_camera_get_image(camera);
```

// Gambar dari kamera diperoleh di sini menggunakan fungsi

```
    if (pause_counter > 0)
```

// Jika masih d atas 0, kecepatan motor diatur ke nol.

```
      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;
   }
```

// Jika kamera tidak tersedia, kecepatan motor akan diatur ke nol

```
   else if (!image) {
    left_speed = 0;
    right_speed = 0;
   } else {  // pause_counter == 0
```

//Di sini, nilai-nilai warna merah, hijau, dan biru dihitung untuk bagian tengah citra kamera.

```
    red = 0;
    green = 0;
    blue = 0;

    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);
     }
    }
```

//Jenis blob yang terdeteksi ditentukan berdasarkan perbandingan nilai-nilai warna merah, hijau, dan biru.

```
    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;
```

//Jika blob terdeteksi, kecepatan motor diatur ke nol, pesan di cetak ke konsol dengan warna yang sesuai dengan jenis blob yang terdeteksi, dan gambar kamera disimpan ke file.

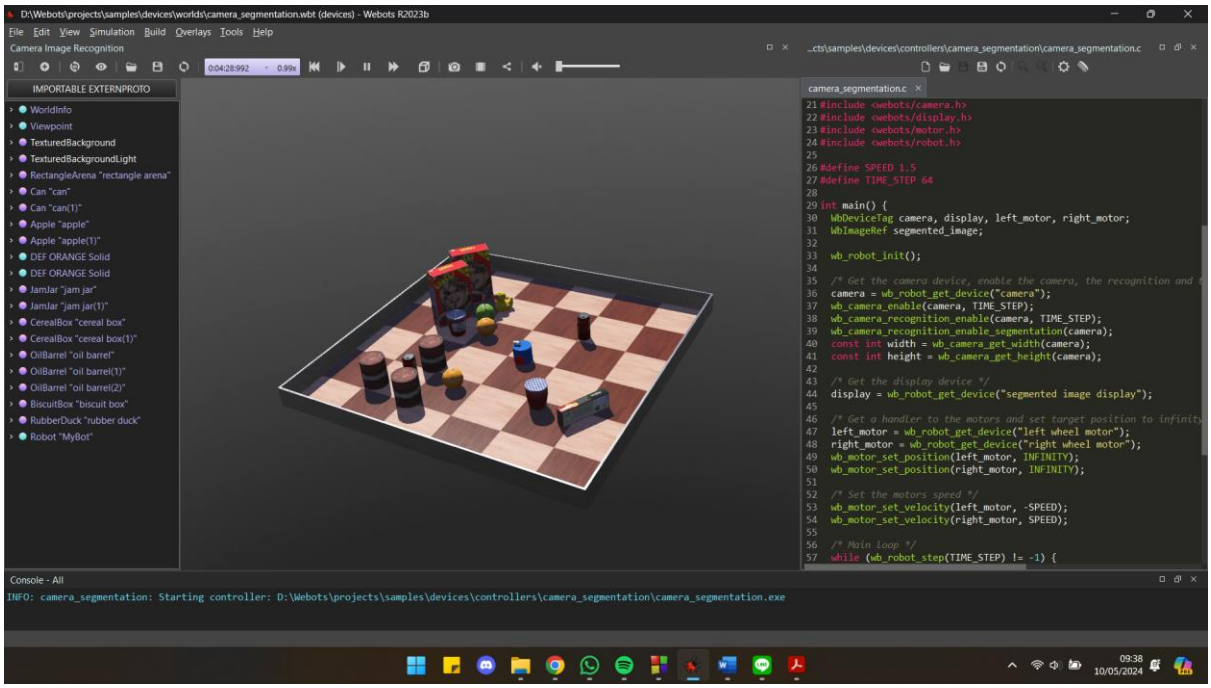
```
    else {
     left_speed = 0;
     right_speed = 0;
     printf("Looks like I found a %s%s%s blob.\n", ansi_colors[current_blob],
color_names[current_blob], ANSI_COLOR_RESET);
     // compute the file path in the user directory
     char *filepath;
#ifdef _WIN32
```

```
      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;
    }
   }

//Kecepatan motor diatur sesuai dengan nilai yang telah dihitung dalam loop.
   wb_motor_set_velocity(left_motor, left_speed);
   wb_motor_set_velocity(right_motor, right_speed);
 }

//Membersihkan sumber daya yang digunakan oleh simulasi, dan mengembalikan dari
awal
 wb_robot_cleanup();
 return 0;
}
```

Kamera robot untuk segmentasi pengenalan dalam lingkungan simulasi Webots



Kamera robot untuk segmentasi pengenalan dalam lingkungan simulasi Webots merupakan sebuah komponen virtual yang dapat disematkan pada robot simulasi di lingkungan

```
//Mengimpor header file yang diperlukan
#include <webots/camera.h>
#include <webots/display.h>
#include <webots/motor.h>
#include <webots/robot.h>

//Untuk mengatur kecepatan robot dan interval waktu simulasi
#define SPEED 1.5
#define TIME_STEP 64

//Variabel yang dideklarasikan kamera, display, dan motor kiri dan kanan.
int main() {
  WbDeviceTag camera, display, left_motor, right_motor;
  WbImageRef segmented_image;



  wb_robot_init(); //untuk menginisialisasi lingkungan simulasi.

//Perangkat kamera diaktifkan dan kemampuan pengenalan dan segmentasi diaktifkan juga.
  camera = wb_robot_get_device("camera");
  wb_camera_enable(camera, TIME_STEP);
  wb_camera_recognition_enable(camera, TIME_STEP);
  wb_camera_recognition_enable_segmentation(camera);
```

```
//Lebar dan tinggi gambar kamera diperoleh untuk penggunaan selanjutnya.
  const int width = wb_camera_get_width(camera); //Lebar Kamera
  const int height = wb_camera_get_height(camera); //Tinggi Kamera

  /* Get the display device */
  display = wb_robot_get_device("segmented image display"); //Perangkat display

  /* Get a handler to the motors and set target position to infinity (speed control). */
  left_motor = wb_robot_get_device("left wheel motor");//Motor kiri
  right_motor = wb_robot_get_device("right wheel motor");//Motor kanan
//Posisi target motor diatur ke INFINITY untuk kontrol kecepatan. Kecepatan motor
ditetapkan ke kecepatan yang telah ditentukan.
  wb_motor_set_position(left_motor, INFINITY);
  wb_motor_set_position(right_motor, INFINITY);
  /* Set the motors speed */
  wb_motor_set_velocity(left_motor, -SPEED);
  wb_motor_set_velocity(right_motor, SPEED);

  //Loop Utama
  while (wb_robot_step(TIME_STEP) != -1) {
    if (wb_camera_recognition_is_segmentation_enabled(camera) &&
wb_camera_recognition_get_sampling_period(camera) > 0) {//memeriksa apakah
segmentasi pengenalan diaktifkan untuk kamera dan periode sampling lebih besar dari
0.
      /* Get the segmented image and display it in the Display */
      const unsigned char *data = wb_camera_recognition_get_segmentation_image(camera);
      if (data) {//mendapatkan gambar segmentasi dan menampilkannya di layar.
        segmented_image = wb_display_image_new(display, width, height, data,
WB_IMAGE_BGRA);
        wb_display_image_paste(display, segmented_image, 0, 0, false);
        wb_display_image_delete(display, segmented_image); //Setelah ditampilkan, gambar
yang dihasilkan dihapus dari display.
      }
    }
  }
//Membersihkan sumber daya yang digunakan oleh simulasi, dan mengembalikan dari
awal
  wb_robot_cleanup();

  return 0;
}
```