

Nama : Arif Al Imran

NIM : 1103210193

## **Laporan Hasil Analisis Simulasi**

### **Ekstraksi Garis dengan Hough Transform**

Hough Transform adalah metode yang digunakan untuk mendeteksi garis pada gambar berbasis tepi. Pada simulasi ini, gambar input diubah menjadi grayscale menggunakan `cv2.cvtColor`, dan deteksi tepi dilakukan dengan algoritma Canny (`cv2.Canny`). Parameter seperti threshold minimum dan maksimum pada algoritma Canny memengaruhi sensitivitas deteksi. Setelah itu, fungsi `cv2.HoughLines` digunakan untuk mendeteksi garis dalam ruang parameter. Setiap garis yang terdeteksi direpresentasikan oleh parameter  $\rho$  (jarak dari asal koordinat) dan  $\theta$  (sudut). Untuk menampilkan hasil, setiap garis diterjemahkan ke dalam koordinat kartesian dan digambar menggunakan `cv2.line`. Dengan parameter akurasi sudut dan jarak yang disesuaikan, simulasi ini menunjukkan hasil deteksi yang cukup akurat pada gambar dengan kontras yang baik.

### **Template Matching untuk Deteksi Objek**

Template Matching adalah metode pencocokan pola untuk menemukan subgambar tertentu dalam gambar utama. Dalam simulasi ini, template objek yang dicari diubah menjadi grayscale menggunakan `cv2.cvtColor` untuk keseragaman proses. Fungsi `cv2.matchTemplate` dengan metode korelasi, seperti `cv2.TM_CCOEFF`, digunakan untuk mencocokkan template dengan gambar utama. Hasilnya adalah peta nilai kesesuaian yang dianalisis untuk menemukan koordinat dengan skor maksimum menggunakan `cv2.minMaxLoc`. Deteksi ini sangat bergantung pada ukuran dan orientasi template, sehingga perubahan dalam skala atau rotasi dapat memengaruhi akurasi. Hasil deteksi divisualisasikan dengan menggambar persegi panjang di sekitar lokasi template menggunakan `cv2.rectangle`.

### **Pembuatan Pyramid Gambar**

Pyramid gambar digunakan untuk menganalisis gambar pada berbagai resolusi. Simulasi ini menggunakan `cv2.pyrDown` untuk mengecilkan gambar dan `cv2.pyrUp` untuk memperbesar gambar. Setiap langkah dilengkapi dengan konversi gambar ke grayscale agar proses lebih cepat. Penggunaan pyramid memungkinkan deteksi fitur pada skala yang berbeda, yang berguna dalam aplikasi seperti deteksi objek multi-skala. Hasilnya divisualisasikan dengan menyusun setiap level pyramid dalam satu tampilan menggunakan fungsi `numpy.hstack`, mempermudah analisis perbedaan detail di tiap tingkat resolusi.

### **Deteksi Lingkaran dengan Hough Transform**

Deteksi lingkaran menggunakan Hough Transform dilakukan dengan fungsi `cv2.HoughCircles`. Data input yang digunakan adalah gambar hasil deteksi tepi dari algoritma Canny. Parameter seperti akurasi akumulasi, jarak antar lingkaran, dan kisaran radius diatur untuk mendukung deteksi yang optimal. Setiap lingkaran yang terdeteksi divisualisasikan dengan menggambar lingkaran dan titik pusatnya menggunakan fungsi `cv2.circle`. Analisis ini menunjukkan bahwa keberhasilan deteksi lingkaran sangat bergantung pada kualitas gambar dan parameter yang dipilih.

### **Ekstraksi Warna Dominan pada Gambar**

Ekstraksi warna dominan dilakukan dengan algoritma pengelompokan K-Means (cv2.kmeans). Gambar input dikonversi ke ruang warna RGB dan direstrukturisasi menjadi array dua dimensi untuk memudahkan proses clustering. Dengan menetapkan jumlah cluster sesuai jumlah warna dominan yang diinginkan, algoritma ini mengelompokkan piksel berdasarkan kesamaan warna. Hasil dari simulasi ini divisualisasikan dengan menciptakan patch warna yang merepresentasikan setiap cluster, membantu pengguna memahami distribusi warna utama dalam gambar.

### **Deteksi Kontur pada Gambar**

Kontur adalah representasi batas objek dalam gambar. Dalam simulasi ini, gambar diubah menjadi biner dengan metode thresholding menggunakan cv2.threshold. Fungsi cv2.findContours digunakan untuk mendeteksi semua kontur dalam gambar. Dengan parameter seperti mode pengambilan kontur (cv2.RETR\_EXTERNAL untuk kontur terluar) dan metode aproksimasi (cv2.CHAIN\_APPROX\_SIMPLE untuk efisiensi data), kontur digambar ulang pada gambar asli menggunakan cv2.drawContours. Pendekatan ini sangat berguna untuk menganalisis bentuk dan ukuran objek dalam gambar.

### **Analisis Tugas Simulasi Webots: Lidar Data Extraction and Obstacle Detection**

Dalam tugas simulasi ini, data dari sensor lidar diambil untuk mendeteksi keberadaan dan posisi objek. Menggunakan pustaka C, data lidar diekstraksi dalam bentuk array jarak. Array ini diolah untuk mendeteksi perubahan signifikan yang menunjukkan adanya rintangan. Proses ini dimulai dengan pembacaan data lidar secara langsung menggunakan API Webots. Selanjutnya, thresholding diterapkan untuk membedakan objek dari latar belakang berdasarkan jarak tertentu.

Simulasi menggunakan matplotlib untuk memvisualisasikan data lidar dalam bentuk plot polar, membantu pengguna memetakan posisi rintangan secara intuitif. Fungsi deteksi jarak minimum digunakan untuk mengidentifikasi rintangan terdekat, sedangkan algoritma clustering sederhana diterapkan untuk mengelompokkan data yang berdekatan, memungkinkan estimasi ukuran dan bentuk rintangan. Setiap hasil dianalisis untuk mengukur efektivitas parameter seperti resolusi sudut lidar dan ambang jarak. Dengan visualisasi tambahan menggunakan plt.scatter, simulasi memberikan wawasan yang jelas tentang distribusi rintangan di sekitar robot.