

Analisis Week 11

Nur Ihsan Ibrahim Abdul Fattah

1103210191

TK45G09

Analisis Lidar :

Kode ini adalah program untuk mengendalikan robot menggunakan sensor lidar dan sensor jarak (ultrasonic) untuk navigasi dan penghindaran tabrakan. Program ini berfungsi untuk mengatur kecepatan motor robot berdasarkan data yang diperoleh dari sensor jarak, serta menggunakan informasi dari lidar untuk memperoleh gambaran tentang lingkungan sekitar. Pada awal program, robot dan perangkat yang dibutuhkan seperti lidar, sensor ultrasonic, dan motor diinisialisasi. Lidar digunakan untuk mengukur jarak di sekitar robot, sementara sensor jarak ultrasonic (di bagian kiri dan kanan robot) digunakan untuk mendeteksi rintangan di depan sensor. Setelah itu, program mengaktifkan perangkat tersebut dan siap menerima data dari masing-masing sensor. Fungsi `extract_lidar_data` membaca data dari lidar dan menampilkan informasi jarak dari objek yang terdeteksi. Fungsi `read_distance_sensors` mengumpulkan nilai jarak dari sensor ultrasonic kiri dan kanan dan menampilkannya. Berdasarkan data sensor ini, fungsi `compute_speeds` menghitung kecepatan roda robot. Kecepatan motor disesuaikan dengan data dari sensor untuk menghindari tabrakan atau menjaga robot tetap bergerak dengan kecepatan yang sesuai. Di dalam loop utama, robot terus-menerus membaca data dari sensor lidar dan ultrasonic, kemudian menghitung kecepatan roda dan mengatur kecepatan motor sesuai perhitungan. Setelah simulasi selesai, memori robot dibersihkan. Dengan struktur seperti ini, robot dapat melakukan navigasi secara mandiri sambil menghindari rintangan, meskipun kode ini masih dapat dikembangkan lebih lanjut untuk meningkatkan algoritma penghindaran tabrakan atau menambah fungsionalitas lainnya.

Analisis Google colab :

- **Deteksi Garis:** Kode pertama mendeteksi garis menggunakan algoritma Canny Edge Detection dan Hough Transform. Gambar diubah menjadi grayscale untuk menyederhanakan pemrosesan, kemudian tepi gambar diekstraksi menggunakan algoritma Canny. Setelah itu, Hough Transform mendeteksi garis berdasarkan representasi koordinat polar (ρ dan θ). Hasilnya adalah gambar asli dengan garis yang ditampilkan dalam warna hijau, cocok untuk aplikasi seperti navigasi robot atau pemetaan jalan.
- **Template Matching:** Bagian kedua mendeteksi objek spesifik (dalam contoh disebut "Vas Bunga") dengan metode template matching. Gambar utama dan template dibandingkan menggunakan korelasi normalisasi (`cv2.TM_CCORR_NORMED`). Lokasi dengan nilai kecocokan tertinggi diberi kotak hijau dan label. Teknik ini cocok untuk mendeteksi objek tertentu, meskipun akurasi bergantung pada kesesuaian ukuran dan orientasi template.
- **Gaussian Pyramid:** Bagian ketiga membuat Gaussian Pyramid, yang secara bertahap mengurangi resolusi gambar menggunakan filter Gaussian. Teknik ini berguna untuk

analisis skala multi-resolusi, seperti kompresi gambar atau deteksi fitur yang muncul pada berbagai skala.

- **Deteksi Lingkaran:** Kode ini mendeteksi lingkaran dalam gambar menggunakan Hough Transform untuk lingkaran. Parameter seperti radius minimum dan maksimum serta sensitivitas pengaturan mendukung fleksibilitas deteksi. Lingkaran yang terdeteksi digambar ulang pada gambar asli dengan lingkaran hijau di tepi dan titik merah di pusatnya.
- **Ekstraksi Warna Dominan:** Algoritma K-Means Clustering digunakan untuk mengidentifikasi lima warna dominan dalam gambar. Setiap warna dominan ditampilkan sebagai kotak warna. Pendekatan ini berguna dalam aplikasi seperti desain visual, palet warna, atau analisis estetika gambar.
- **Deteksi Kontur:** Bagian terakhir mendeteksi kontur besar dalam gambar. Setelah mengurangi noise dengan Gaussian Blur dan menerapkan adaptive thresholding untuk menangani pencahayaan tidak merata, kontur ditemukan menggunakan algoritma Canny dan `cv2.findContours`. Hanya kontur besar yang digambar pada gambar asli untuk fokus pada fitur utama.