

# LAPORAN PRAKTIKUM KONTROL CERDAS

Nama : Nefi Afif Sujatyana  
NIM : 224308093  
Kelas : TKA-7D  
Akun Github : <https://github.com/nefiafif612-cmd>

## 1. Judul Percobaan:

*Canny Edge Detection & Lane Detection with Instance Segmentation*

## 2. Tujuan Percobaan:

- Mahasiswa mampu memahami konsep *Canny Edge Detection* sebagai metode dasar deteksi tepi.
- Mahasiswa dapat menggunakan *Instance Segmentation* untuk deteksi jalur rel kereta (*Lane Detection*).
- Mahasiswa dapat menggunakan dataset *Rail Segmentation* dari Kaggle untuk eksperimen.
- Mahasiswa dapat menggabungkan metode *Canny Edge Detec*

## 3. Landasan Teori:

Kecerdasan buatan atau Artificial Intelligence (AI) adalah sistem yang mempelajari bagaimana membuat komputer dapat berpikir, belajar, dan bertindak seperti manusia (Dosari & Abouellail, 2023). Dalam beberapa tahun terakhir, model YOLO (You Only Look Once) telah menunjukkan kinerja yang sangat baik dalam deteksi objek dan pengenalan pose manusia. YOLO adalah model deteksi satu tahap (single-stage) yang mampu memprediksi posisi bounding box dan kategori objek secara langsung dalam satu propagasi jaringan, sehingga mempercepat proses inferensi dibandingkan metode dua tahap (two-stage) seperti R-CNN dan Faster R-CNN (Giulietti dkk., 2025). Versi terbaru dari model ini, YOLOv8, menghadirkan peningkatan signifikan dalam arsitektur jaringan dan strategi pelatihan, sehingga mampu meningkatkan akurasi dan kecepatan deteksi dalam skenario real-time yang kompleks (Ningrum dkk., 2024). Penerapan YOLOv8 Pose untuk inferensi pose pada gambar, video, dan kamera real-time telah membuka peluang baru

dalam pengembangan sistem pengenalan aktivitas manusia. Kombinasi strategi pengembangan arsitektur, pengolahan fitur multi-skala, dan pengurangan kehilangan fitur akibat occlusion menjadikan YOLOv8 Pose sebagai salah satu model paling unggul untuk tugas HPE. Peningkatan dalam akurasi dan efisiensi inferensi ini memungkinkan YOLOv8 Pose untuk diterapkan dalam berbagai bidang, seperti pengawasan keamanan, pengenalan aktivitas dalam olahraga, interaksi manusia-robot, dan analisis perilaku manusia secara real-time (Giulietti dkk., 2025).

Pengolahan citra melibatkan analisis dan perubahan gambar digital untuk meningkatkan kualitas gambar atau mengekstrak informasi penting dari gambar. Berbagai teknik pengolahan gambar melakukan proses ini, seperti peningkatan gambar, restorasi gambar, segmentasi, dan deteksi tepi. Pengolahan gambar digunakan dalam berbagai industri, termasuk medis, pengawasan keamanan, pengenalan pola, dan otomotif. Salah satu komponen penting pengolahan citra adalah deteksi tepi, yang membantu memahami struktur dan bentuk objek dalam gambar. Algoritma detektor tepi Canny diciptakan oleh John F. Canny pada tahun 1986. Algoritma ini dibuat untuk memenuhi tiga kriteria utama: deteksi tepi yang baik, penempatan yang tepat, dan respons minimal. Beberapa langkah utama diambil dalam proses ini: penggunaan filter Gaussian untuk menghaluskan gambar, menghitung gradien intensitas, penghapusan non-maksimal untuk menghilangkan piksel yang bukan merupakan bagian dari tepi, dan penggunaan ambang hysteresis untuk mengidentifikasi dan menghubungkan tepi yang kuat dan lemah. Keunggulan utama metode ini adalah kemampuan untuk mendeteksi tepi dengan presisi tinggi bahkan pada gambar yang mengandung noise (Salkiawati dkk., 2021).

Prinsip kerja dari computer vision terdiri dari beberapa tahapan yaitu; Images Acquisition, Processing, Analyzing dimana kamera akan menangkap representasi objek lalu di tangkap oleh sensor optik lalu diubah menjadi data digital, dan data-data tersebut akan diproses untuk di analisa Invalid source specified. Salah satu teknik yang menjadi bagian proses dalam computer vision adalah Hough Transform dimana proses ini akan memudahkan sebuah sistem untuk mendeteksi sebuah garis dalam sebuah gambar binari Invalid source

specified. Canny Edge Detection , sebagai metode berbasis gradien, telah lama menjadi andalan dalam mendeteksi tepi pada gambar. Keunggulannya terletak pada ketahanannya terhadap noise, yang dicapai melalui penggunaan Gaussian Blur sebagai langkah pra-pemrosesan (Taqqy dkk., 2024). Metode ini bekerja dengan mengidentifikasi perbedaan intensitas piksel, memungkinkan deteksi tepi yang tajam. Namun, dalam konteks deteksi jalur yang kompleks, metode ini memiliki keterbatasan karena hanya mendeteksi tepi tanpa pemahaman semantik. Di sisi lain, Lane Detection dengan Instance Segmentation menawarkan solusi yang lebih canggih. Dengan memanfaatkan model deep learning untuk segmentasi objek berbasis instan, metode ini mampu membedakan jalur dari lingkungan sekitarnya dengan akurasi yang lebih tinggi. Pendekatan ini memungkinkan identifikasi jalur yang lebih tepat, mengatasi kelemahan metode berbasis tepi klasik seperti Canny, dan memberikan pemahaman kontekstual yang lebih baik terhadap lingkungan jalan.

#### **4. Analisis dan Diskusi**

- **Analisis**

Pada praktikum minggu kelima ini percobaan yang dilakukan adalah membuat Canny Edge Detection & Lane Detection with Instance Segmentation untuk mendeteksi tepi dalam sebuah gambar. Ini melibatkan pengidentifikasian titik-titik dalam gambar di mana kecerahan berubah secara tajam, yang seringkali sesuai dengan batas-batas objek. Dimana pada percobaan praktikum ini, dapat mendeteksi dan mengidentifikasi jalur rel. Tahap awal yang dilakukan adalah mengunduh library “numpy torch torchvision ultralytics matplotlib”. Kemudian, mengunduh dataset kaggle yang berisi kumpulan foto rel kereta api. Tahap selanjutnya adalah membuat program untuk mengimplementasikan Canny Edge Detection . Pertama buat file canny\_edge.py dan tambahkan kode def canny\_edge\_detection(image\_path) untuk mendeteksi tepi dari sebuah gambar menggunakan metode Canny Edge Detection dan kode cv2.imshow("Canny Edge Detection ", edges) | cv2.imwrite("canny\_result.jpg", edges) digunakan untuk membaca dan

menyimpan hasil deteksi tepi ke file untuk digunakan nanti. Pada kasus deteksi jalur rel hasil dari Canny Edge Detection dapat digunakan sebagai input untuk metode deteksi deteksi jalur rel lainnya. Tahap berikutnya adalah membuat program untuk mengimplementasikan Lane Detection dengan Instance Segmentation (YOLOv8-seg). Pertama buat file `lane_detection.py` dan tambahkan kode `model = YOLO("yolov8n-seg.pt")` untuk melakukan segmentasi instan, yang berarti ia dapat mendeteksi dan mensegmentasi objek dalam gambar, menghasilkan masker untuk setiap objek yang terdeteksi. Lalu, kode `def detect_rail_lane(image_path):` digunakan untuk mendefinisikan sebuah fungsi bernama `detect_rail_lane` yang menerima `image_path` (jalur file gambar) sebagai input. Kode `results = model(image_path, show=True) | results[0].save("lane_detection_result.jpg")` juga ditambahkan untuk menjalankan deteksi dan segmentasi instan pada gambar yang diberikan menggunakan model YOLOv8 dan menyimpan hasil deteksi (termasuk masker segmentasi) ke dalam file. Tahap selanjutnya adalah membuat program untuk menggabungkan Canny Edge Detection dengan Lane Detection. Pertama buat file `combined_detection.py` dan tambahkan kode `model = YOLO("yolov8n-seg.pt")` untuk melakukan segmentasi instan, yang berarti ia dapat mendeteksi dan mensegmentasi objek dalam gambar, menghasilkan masker untuk setiap objek yang terdeteksi. Lalu, kode `def combined_detection(image_path):` digunakan untuk mengambil gambar jalur sebagai input dan melakukan deteksi jalur gabungan. `canny_result = canny_edge_detection(image_path)` kode tersebut juga ditambahkan untuk menjalankan Canny Edge Detection. Kemudian, kode `results = model(image_path) | lane_img = results[0].plot()` digunakan untuk menjalankan model YOLOv8-seg pada gambar input untuk mendeteksi jalur dan menghasilkan gambar dengan hasil deteksi jalur dari YOLOv8 seg, termasuk masker segmentasi dan kotak pembatas. Selanjutnya, tambahkan kode `combined = cv2.addWeighted` untuk menggabungkan hasil dari YOLOv8-seg dan Canny Edge Detection menggunakan pembobotan. Ini memungkinkan visualisasi kedua hasil secara bersamaan,

dengan menggabungkan kedua hasil, diharapkan dapat menghasilkan deteksi jalur yang lebih kuat dan akurat.

Kemudian, dilakukan modifikasi program model YOLOv8-seg agar hanya mendeteksi jalur rel dengan mengubah parameter Canny Edge Detection dan membandingkan hasilnya. Langkah awal yang dilakukan untuk memodifikasi adalah ambil data berupa foto rel kereta api sebanyak kurang lebih 1000 foto untuk dijadikan sebagai dataset. Setelah itu, unggah folder foto pada website roboflow, kemudian untuk dilakukan labelling pada seluruh foto agar hanya dapat mendeteksi rel kereta api saja. Lalu, dilakukan augmentasi pada foto tersebut dan dijadikan sebuah dataset. Dataset yang diperoleh akan berupa sebuah folder yang di dalamnya terdapat file “best.pt” berfungsi untuk mencapai hasil deteksi rel kereta api yang paling akurat dan andal. . Pada percobaan ini, dilakukan dengan 2 “best.pt” yang berbeda yaitu sebelum dilakukan modifikasi dan sesudah modifikasi. Langkah selanjutnya adalah memindahkan folder tersebut kedalam sebuah folder yang sama dengan program file. Program yang digunakan pada percobaan modifikasi ini adalah dengan menggabungkan dua teknik pemrosesan citra yaitu deteksi tepi Canny dan deteksi objek (atau segmentasi) menggunakan model YOLO. Tujuannya adalah untuk mendeteksi tepi objek dalam video langsung dari kamera laptop, dengan fokus pada area yang diidentifikasi oleh model YOLO. Kode MODEL\_PATH ditambahkan untuk memuat model YOLO dari jalur file yang ditentukan (ex: MODEL\_PATH = "C:/Users/user/intellegent control-week6/rail-segmentation/best.pt"). Kemudian, ditambahkan fungsi detect\_canny\_edges\_with\_model() memulai loop pemrosesan frame dari kamera. Setiap frame diubah menjadi skala abu-abu untuk deteksi tepi Canny, dan menjadi RGB untuk prediksi YOLO. Hasil prediksi YOLO, baik berupa masker segmentasi atau kotak pembatas, yang digunakan untuk menandai area objek (rail) dalam frame. Overlay transparan juga ditambahkan untuk visualisasi hasil segmentasi atau deteksi objek. Tepi Canny dimodifikasi, dimana tepi di dalam area objek yang terdeteksi diubah menjadi merah, sementara tepi di luar area objek tetap putih. Hasil

penggabungan dari frame asli dan tepi Canny yang dimodifikasi ditampilkan dalam jendela OpenCV. Loop terus berjalan hingga tombol 'q' ditekan, kemudian kamera dilepaskan dan semua jendela OpenCV ditutup

- **Diskusi**

Pada praktikum minggu kelima ini dalam percobaan yang telah dilakukan yaitu Canny Edge Detection & Lane Detection with Instance Segmentation untuk mendeteksi tepi dalam sebuah gambar. Ini melibatkan pengidentifikasian titik-titik dalam gambar di mana kecerahan berubah secara tajam, yang seringkali sesuai dengan batas-batas objek. Dimana pada percobaan praktikum ini, dapat mendeteksi dan mengidentifikasi jalur rel. Canny Edge Detection lebih baik digunakan dalam aplikasi yang membutuhkan deteksi tepi yang cepat dan sederhana, seperti pengolahan citra dasar atau deteksi tepi objek dengan kontras yang jelas. Canny Edge Detection juga lebih baik digunakan ketika sumber daya komputasi terbatas. Sedangkan Instance Segmentation lebih baik digunakan dalam aplikasi yang membutuhkan pemahaman semantik dan deteksi objek yang kompleks, seperti deteksi jalur, kendaraan, atau objek lain dalam lingkungan yang kompleks. Untuk untuk mengoptimalkan kinerja model dalam mengidentifikasi jalan dari lingkungan sekitarnya dibutuhkan peningkatan deteksi jalur dengan menggunakan tuning parameter YOLOv8-seg. Peningkatan ini dilakukan dengan cara menyesuaikan anchor boxes agar lebih sesuai dengan ukuran jalur yang terdeteksi, melakukan data augmentation dengan variasi pencahayaan dan perspektif untuk meningkatkan ketahanan model, mengatur confidence threshold agar model tidak terlalu sensitif terhadap false positives dan memanfaatkan fine-tuning pada layer tertentu untuk meningkatkan spesialisasi model dalam mendeteksi jalur. Metode Instance Segmentation dapat membantu dalam mendeteksi jalur kereta secara real-time dan memberikan input ke sistem kontrol otomatis.

Kombinasi Canny dan Instance Segmentation dapat membantu dalam mendeteksi rel dengan lebih akurat dalam kondisi lingkungan yang berubah-ubah. Selain itu, metode ini dapat digunakan untuk sistem

peringatan dini jika ada hambatan di jalur kereta dengan mendeteksi objek asing yang tidak sesuai dengan pola jalur biasa. Penerapan metode ini dalam sistem navigasi kereta otomatis dapat meningkatkan keandalan dan ketepatan dalam pengenalan jalur, sehingga mendukung sistem pengendalian kereta yang lebih aman dan efisien. Kombinasi antara Canny Edge Detection dan Instance Segmentation memungkinkan sistem untuk mendeteksi jalur secara real-time dengan kecepatan dan akurasi tinggi. Canny Edge Detection dapat digunakan untuk memberikan respons cepat dalam mengenali tepi jalur, sedangkan Instance Segmentation dapat memperbaiki hasil deteksi dengan membedakan jalur dari elemen lain di sekitar jalur seperti kendaraan atau rintangan. Pada sistem navigasi kereta otomatis, hasil deteksi jalur ini dapat digunakan untuk mengatur kecepatan kereta, menghindari tabrakan, dan memastikan kereta tetap berada pada jalur yang benar. Selain itu, sistem dapat dilengkapi dengan algoritma pemrosesan citra tambahan untuk mengenali sinyal atau rambu di sepanjang jalur, sehingga memungkinkan sistem untuk menyesuaikan kecepatan atau melakukan pengereman secara otomatis.




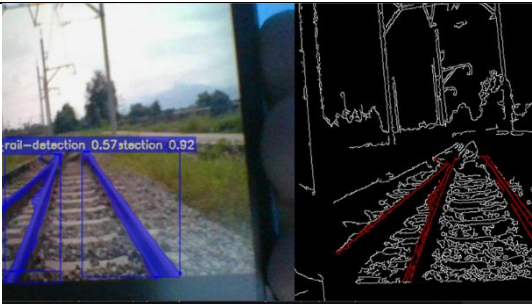
##### **5. *Assignment:***

Dalam praktikum yang telah dilakukan mengenai modifikasi program model YOLOv8-seg agar hanya mendeteksi jalur rel dengan mengubah parameter Canny Edge Detection dan membuat dataset yang baru memiliki tantangan dan kelebihan. Tantangan utama terletak pada sensitivitas parameter ambang batas Canny. Parameter ambang batas (thresholds) pada Canny Edge Detection sangat sensitif. Penyesuaian yang tidak tepat dapat menghasilkan deteksi tepi yang berlebihan (noise) atau kehilangan tepi penting dari jalur rel. Menemukan nilai ambang batas optimal memerlukan eksperimen yang cermat dan pemahaman mendalam tentang karakteristik gambar. Kondisi pencahayaan dan lingkungan yang berubah-ubah (misalnya, bayangan, cuaca buruk, atau kotoran di rel) dapat memengaruhi kinerja Canny Edge Detection secara signifikan. Membuat dataset baru yang akurat dan representatif untuk jalur rel memerlukan waktu dan upaya yang besar. Proses pelabelan data harus dilakukan dengan cermat untuk memastikan kualitas dataset yang tinggi. Meskipun demikian, terdapat kelebihan yaitu Canny Edge Detection dapat meningkatkan akurasi deteksi tepi jalur rel, terutama dalam kondisi di mana marka jalan tidak jelas atau kontrasnya rendah. Ini dapat membantu YOLOv8 seg untuk mengidentifikasi dan memisahkan jalur rel dengan lebih tepat. Deteksi tepi yang dihasilkan oleh Canny dapat memberikan informasi fitur tambahan yang berguna bagi YOLOv8-seg, meningkatkan pemahaman model tentang geometri dan lokasi jalur rel. Pendekatan gabungan ini memiliki

potensi untuk aplikasi khusus dalam pemantauan rel kereta api, pemeliharaan infrastruktur, dan sistem navigasi kereta api otomatis. Dengan menggunakan dataset yang dibuat khusus memungkinkan model untuk diadaptasi dengan lebih baik terhadap karakteristik unik jalur rel, seperti jenis rel, kondisi lingkungan, dan variasi pencahayaan

## 6. Data dan Output Hasil Pengamatan:

Data yang diperoleh pada praktikum minggu ke-6 dengan tabel dibawah ini:

No	Variabel	Hasil Pengamatan
1	Gambar asli rail	
2	Menggunakan program canny_edge.py	
3	Menggunakan program lane_detection.py	
4	Menggunakan program combine_modifikasi.py	



## **7. Kesimpulan:**

Berdasarkan praktikum dan analisis yang telah dilakukan, maka dapat ditarik kesimpulan yaitu:

- Pada praktikum ini berhasil menerapkan metode Canny Edge Detection dan Lane Detection dengan Instance Segmentation (YOLOv8-seg) untuk mendeteksi jalur rel kereta api dengan lebih akurat.
- Canny Edge Detection bekerja dengan baik dalam mendeteksi tepi objek dengan cepat dan sederhana, tetapi kurang efektif dalam memahami konteks gambar secara keseluruhan. Sedangkan Instance Segmentation menggunakan YOLOv8-seg memberikan hasil yang lebih akurat dalam mengenali jalur rel, terutama setelah dilakukan pelabelan dataset dan tuning parameter.
- Kombinasi kedua metode tersebut memberikan hasil yang lebih baik dibandingkan penggunaan metode secara terpisah, terutama dalam kondisi pencahayaan yang beragam dan lingkungan yang kompleks.
- Modifikasi program dengan menggunakan dataset yang lebih spesifik dan tuning parameter pada YOLOv8-seg berhasil meningkatkan keakuratan deteksi jalur rel serta mengurangi kesalahan deteksi objek yang tidak relevan.

## **8. Saran:**

Untuk meningkatkan akurasi dan efisiensi dalam mendeteksi jalur rel, disarankan untuk memperluas dan memvariasikan dataset dengan menambahkan lebih banyak gambar dari berbagai kondisi pencahayaan dan cuaca. Selain itu, performa model YOLOv8-seg dapat ditingkatkan melalui fine tuning, seperti penyesuaian parameter anchor boxes dan confidence threshold. Agar lebih aplikatif di dunia nyata, hasil deteksi jalur rel dapat diintegrasikan dengan sistem navigasi otomatis guna mendukung efisiensi dan keselamatan transportasi kereta. Pengujian model dalam berbagai kondisi nyata, seperti rel yang tertutup oleh objek atau saat hujan, juga penting untuk memastikan keandalan sistem. Di samping itu, penerapan metode machine learning yang lebih maju, seperti segmentasi berbasis transformer atau self-

supervised learning, dapat digunakan untuk meningkatkan akurasi deteksi dan mengoptimalkan kinerja sistem secara keseluruhan.

## **9. Daftar Pustaka:**

- Dosari, F. H. M. A., & Abouellail, S. I. A. D. (2023). Artificial Intelligence (AI) Techniques for Intelligent Control Systems in Mechanical Engineering. *American Journal of Smart Technology and Solutions*, 2(2), 55–64. <https://doi.org/10.54536/ajsts.v2i2.2188>
- Giulietti, N., Todesca, D., Carnevale, M., & Giberti, H. (2025). A Real-Time Human Pose Measurement System for Human-In-The-Loop Dynamic Simulators. *IEEE Access*, 13, <https://doi.org/10.1109/ACCESS.2025.3538332> 24954–24969.
- Ningrum, A. T., Wijay, R., Aziz, M. R. A., Yudha, M., & Rosyani, P. (2024). Face Deteksi Objek pada Gambar dan Video dengan YOLOv8. 2(2).
- Salkiawati, R., Alexander, A. D., & Lubis, H. (2021). Implementasi Canny Edge Detection Pada Aplikasi Pendeteksi Jalur Lalu Lintas. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, <https://doi.org/10.30865/mib.v5i1.2502> 5(1), 164.
- Taqy, M. A., Yahya, C. A., Hafizh, M. A., Nurakmalia, S., & Rosyani, P. (2024). Implementasi dan Analisis Metode Deteksi Tepi Canny Menggunakan OpenCV. 1(4).