

Laporan Praktikum Kontrol Cerdas

Nama : Rhehan Adi Prakoso
NIM : 224308094
Kelas : TKA-7D
Akun Github (Tautan) : <https://github.com/RhehanAdi>
Student Lab Assistant : -

1. Judul Percobaan

Week 6 : Canny Edge Detection & Lane Detection with Instance Segmentation.

2. Tujuan Percobaan

Tujuan dari dilakukannya praktikum “*Canny Edge Detection & Lane Detection with Instance Segmentation*” adalah:

- a) Memahami konsep *Canny Edge Detection* sebagai metode dasar deteksi tepi.
- b) Menggunakan *Instance Segmentation* untuk deteksi jalur rel kereta (*Lane Detection*).
- c) Menggunakan dataset *Rail Segmentation* dari *Kaggle* untuk eksperimen.
- d) Menggabungkan metode *Canny Edge Detection* dengan *Instance Segmentation* untuk meningkatkan deteksi jalur.

3. Landasan Teori

- A. ***Canny Edge Detection*** adalah metode deteksi tepi pada citra digital yang dikembangkan untuk mengekstraksi tepi secara akurat dan minim *noise*. Metode ini bekerja melalui beberapa tahapan sistematis. Pertama, citra diredam *noise*-nya menggunakan filter *Gaussian* agar gangguan dari *noise* diminimalkan. Kedua, dilakukan perhitungan gradien intensitas untuk menentukan perubahan intensitas pada tiap piksel, yang menandai kemungkinan keberadaan tepi dan arah orientasinya. Tahap berikutnya adalah *non-maximum suppression*, yaitu menipiskan tepi dengan mempertahankan hanya piksel puncak gradien lokal, sehingga tepi menjadi lebih halus dan kontinu. Terakhir, digunakan *double thresholding* untuk membedakan tepi kuat, tepi lemah, dan *non-tepi*; tepi lemah hanya dipertahankan jika terhubung dengan tepi kuat, sehingga meminimalkan respon palsu akibat *noise*. Dengan

tahapan ini, *Canny Edge Detection* mampu menghasilkan citra tepi yang jelas, kontinu, dan siap untuk analisis lebih lanjut, sehingga metode ini banyak diterapkan dalam pengolahan citra, pengenalan objek, dan sistem otomasi berbasis visi komputer.

- B. ***Instance Segmentation*** adalah teknik dalam pengolahan citra yang bertujuan untuk mengidentifikasi setiap objek secara individual sekaligus menentukan batas tepinya (*mask*) pada citra. Berbeda dengan *semantic segmentation*, yang hanya mengklasifikasikan piksel berdasarkan kelas tanpa membedakan objek yang sama, *instance segmentation* mampu membedakan setiap objek meskipun termasuk kelas yang sama. Proses ini biasanya menggabungkan dua tahapan utama: deteksi objek untuk menentukan lokasi dan kelas objek, serta *masking* untuk menandai piksel yang membentuk objek tersebut. Model *modern* seperti *Mask R-CNN* mengintegrasikan kedua tahapan ini secara *end-to-end*, sehingga dapat memprediksi *bounding box*, kelas, dan *mask* sekaligus. Keunggulan metode ini terlihat pada kemampuannya memberikan informasi objek yang lebih presisi, sehingga banyak digunakan dalam aplikasi seperti robotika, analisis citra medis, otomasi industri, pengawasan, dan kendaraan otonom. Dengan kemampuan membedakan setiap *instance*, teknik ini memungkinkan analisis citra yang lebih detail dan pengambilan keputusan yang lebih akurat dibandingkan metode segmentasi konvensional.
- C. ***Rail Segmentation*** dengan pendekatan *Instance Segmentation* adalah teknik pengolahan citra yang bertujuan untuk mendeteksi dan memisahkan setiap jalur rel kereta secara individual dalam citra atau video. Berbeda dengan segmentasi konvensional yang hanya mengklasifikasikan piksel sebagai rel atau bukan, *instance segmentation* memungkinkan setiap jalur rel memiliki identitas unik, sehingga jalur yang saling berdekatan tetap dapat dibedakan. Prosesnya biasanya melibatkan dua tahapan utama: deteksi objek untuk menemukan lokasi rel dalam citra, dan *masking* untuk menandai piksel yang membentuk jalur rel tersebut. Model *deep learning* modern seperti *Mask R-CNN* atau arsitektur berbasis CNN lainnya sering digunakan karena mampu menangani variasi bentuk rel, pencahayaan, dan kondisi lingkungan yang berbeda secara *end-to-end*. Penerapan *rail segmentation* berbasis *instance segmentation*

sangat penting dalam sistem inspeksi rel otomatis, kendaraan kereta otonom, dan pemeliharaan jalur, karena memberikan informasi yang lebih detail dan presisi dibanding segmentasi tradisional, memungkinkan identifikasi rel individual dan deteksi kerusakan dengan akurasi tinggi.

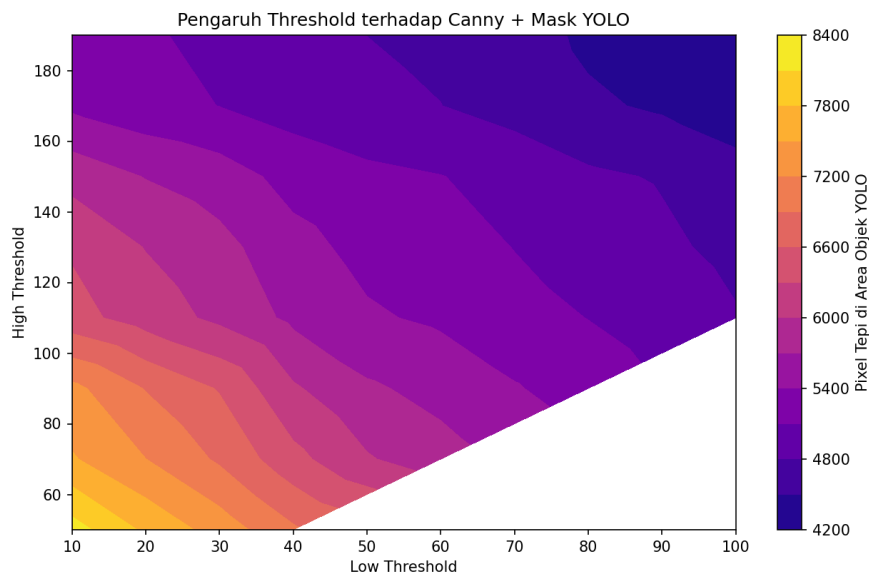
- D. **YOLOv8-seg** adalah versi terbaru dari arsitektur *You Only Look Once* (YOLO) yang dirancang khusus untuk instance segmentation, sehingga mampu mendeteksi objek sekaligus menghasilkan *mask* piksel presisi tinggi untuk setiap *instance*. Berbeda dengan YOLO versi sebelumnya yang hanya fokus pada deteksi *bounding box*, YOLOv8-seg mengintegrasikan modul segmentasi secara *end-to-end*, sehingga prediksi kelas, *bounding box*, dan *mask* dapat dilakukan secara simultan dengan kecepatan tinggi dan akurasi yang baik. Proses pelatihan model ini melibatkan *dataset* yang telah diberi anotasi *mask*, dan menggunakan teknik augmentasi serta *loss function* gabungan yang memperhitungkan kesalahan deteksi dan kesalahan segmentasi. Keunggulan YOLOv8-seg terletak pada kecepatan inferensi yang tinggi, kemampuannya menangani objek berukuran kecil hingga besar, dan skalabilitas untuk berbagai aplikasi. Metode ini banyak digunakan dalam otomasi industri, inspeksi visual, robotika, kendaraan otonom, dan sistem pengawasan, karena memberikan deteksi objek yang cepat serta informasi mask yang detail dan akurat, memungkinkan analisis dan pengambilan keputusan berbasis citra dengan presisi tinggi.

4. Analisis

Pada praktikum ini dilakukan penggabungan metode *Canny Edge Detection* dengan model YOLOv8-seg untuk meningkatkan kemampuan deteksi objek dan presisi tepi pada citra. Tujuan praktikum ini adalah memanfaatkan kekuatan Canny dalam mengekstraksi tepi halus sekaligus kemampuan YOLOv8-seg dalam mendeteksi objek dan menghasilkan *mask* piksel presisi tinggi untuk setiap *instance*. Pendekatan ini diharapkan mampu menghasilkan deteksi objek yang lebih akurat dan tersegmentasi dengan jelas, terutama pada citra dengan objek berukuran kecil atau kondisi kompleks.

Proses percobaan dimulai dengan *preprocessing* citra menggunakan *Canny Edge Detection*. Tahapan ini meliputi reduksi *noise* dengan *filter Gaussian*,

perhitungan *gradien intensitas*, *non-maximum suppression*, dan *double thresholding*. Dua parameter *threshold*, yaitu *lower threshold* dan *upper threshold*, diubah secara bertahap untuk mengamati pengaruhnya terhadap ketajaman dan kelengkapan tepi yang dihasilkan. Hasil Canny kemudian digunakan sebagai masukan tambahan untuk model YOLOv8-seg, sehingga model dapat lebih fokus pada area objek dengan kontur tepi yang jelas. Dari percobaan yang dilakukan diketahui hubungan parameter threshold terhadap hasil deteksi yang digambarkan dalam bentuk grafik yang terlihat pada gambar 1. sebagai berikut.



Berdasarkan gambar 1. terlihat bahwa nilai lower threshold yang terlalu rendah menyebabkan banyak tepi palsu muncul akibat noise, sehingga YOLOv8-seg terkadang menghasilkan mask yang tidak presisi. Sebaliknya, nilai upper threshold yang terlalu tinggi membuat beberapa tepi objek yang lemah tidak terdeteksi, sehingga mask yang dihasilkan menjadi kurang lengkap. Kombinasi threshold yang optimal menghasilkan tepi yang cukup jelas dan kontinu, sehingga YOLOv8-seg mampu memprediksi mask objek dengan lebih akurat dan presisi.

Analisis Hasil

1. Seberapa baik deteksi jalur dengan Instance Segmentation dibandingkan *Canny*?
 - * *Canny Edge Detection* mendeteksi tepi jalur rel dengan cepat, tetapi hanya menghasilkan garis tepi tanpa membedakan objek, sehingga sensitif terhadap noise dan pencahayaan. Sebaliknya, *Instance Segmentation* bisa mengenali jalur rel secara spesifik dengan mask warna, lebih akurat meskipun membutuhkan model terlatih dan perangkat keras lebih kuat.

2. Apakah kombinasi kedua metode dapat meningkatkan akurasi?
 - * Kombinasi *Canny Edge Detection* dan *Instance Segmentation* dapat meningkatkan akurasi deteksi jalur rel. Metode *Canny* menyorot tepi jalur secara detail, sedangkan *Instance Segmentation* mengenali jalur rel secara spesifik. Dengan penggabungan keduanya, hasil deteksi menjadi lebih presisi, karena jalur rel terlihat jelas sekaligus tepi jalur tetap terjaga, sehingga mengurangi kesalahan akibat noise atau variasi pencahayaan.
3. Apa dampak perubahan parameter *Canny (thresholds)* terhadap hasil deteksi
 - * *Threshold* rendah membuat lebih banyak pixel dianggap sebagai tepi, sehingga hasilnya lebih banyak garis, termasuk *noise* atau detail yang tidak penting. Sebaliknya, *threshold* tinggi hanya mendeteksi perubahan intensitas yang kuat, sehingga hasilnya lebih sedikit garis dan tepi utama lebih jelas, tapi detail halus bisa hilang.

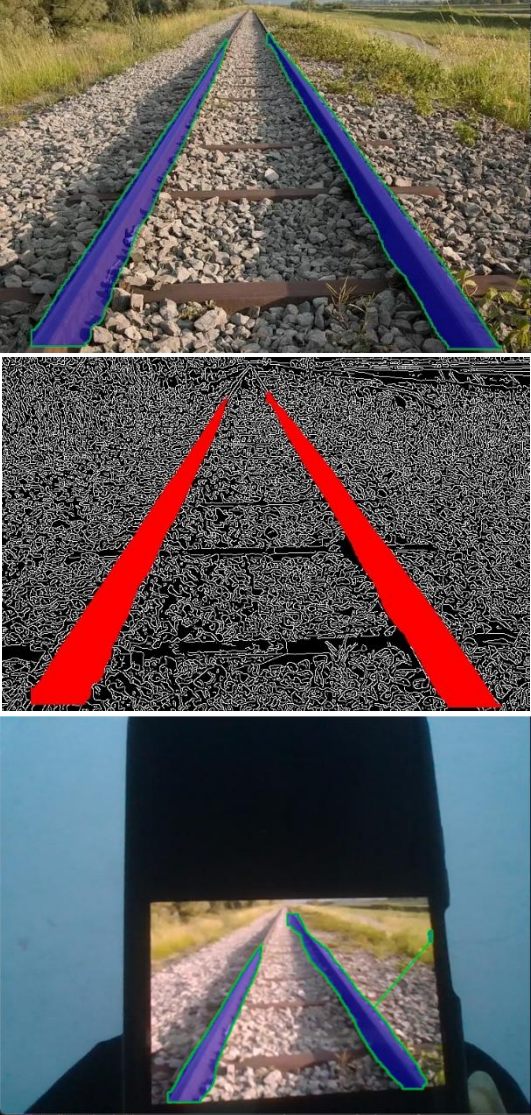
Diskusi

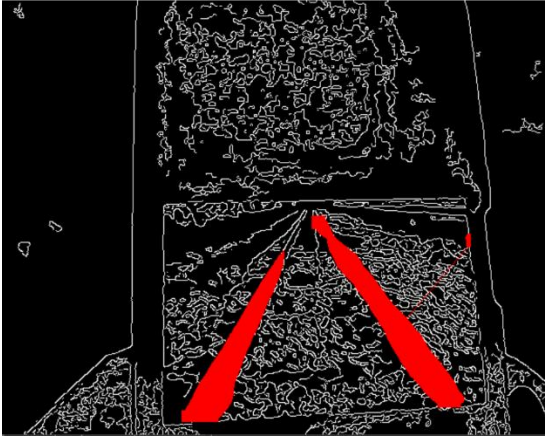


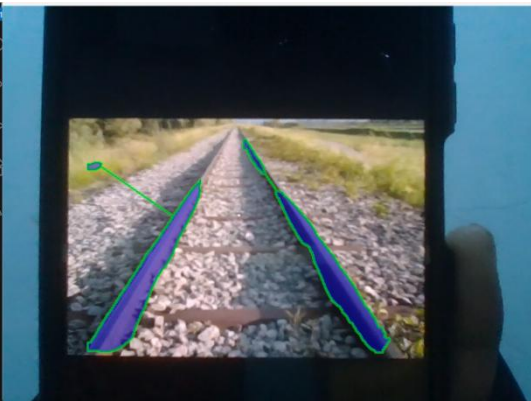
1. Kapan lebih baik menggunakan *Canny Edge Detection* dibanding *Instance Segmentation*?
 - * *Canny Edge Detection* digunakan ketika fokus menemukan tepi objek secara cepat tanpa memerlukan pemahaman semantic serta ringan dan tidak membutuhkan model terlatih, sehingga ideal untuk komputasi terbatas atau saat objeknya jelas dan kontras tinggi. Sebaliknya, jika diperlukan identifikasi objek spesifik atau analisis yang lebih kompleks, seperti menandai jalur rel secara tepat di lingkungan yang beragam, *Instance Segmentation* lebih cocok digunakan.
2. Apakah kombinasi kedua metode dapat meningkatkan akurasi?
 - * Deteksi jalur dengan *YOLOv8-seg* dapat ditingkatkan dengan menyesuaikan *confidence threshold* dan *IoU threshold*, serta menggunakan ukuran input gambar yang lebih besar untuk menangkap detail. Selain itu, augmentasi data dan pelatihan ulang dengan dataset khusus jalur rel membuat model lebih akurat dan robust terhadap variasi kondisi pencahayaan. Post-processing mask, seperti overlay dengan *Canny*, juga dapat memperjelas batas jalur.
3. Bagaimana metode ini dapat diterapkan dalam sistem navigasi kereta otomatis?
 - * *YOLOv8-seg* mengenali jalur rel sebagai objek spesifik, sedangkan *Canny* menyorot tepi jalur dengan detail, sehingga sistem memperoleh informasi posisi dan batas rel secara presisi. Data ini dapat digunakan oleh modul kontrol kereta untuk menentukan arah, kecepatan, dan peringatan potensi bahaya, seperti rel yang rusak atau percabangan jalur.

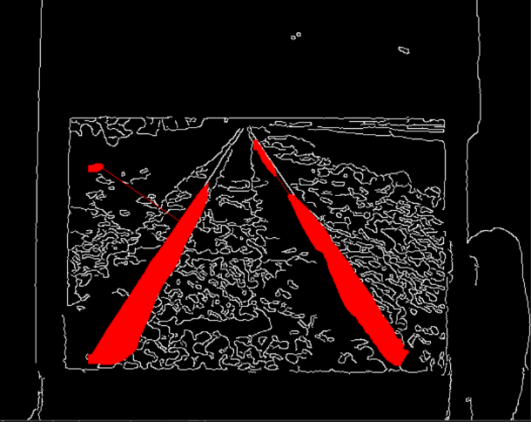
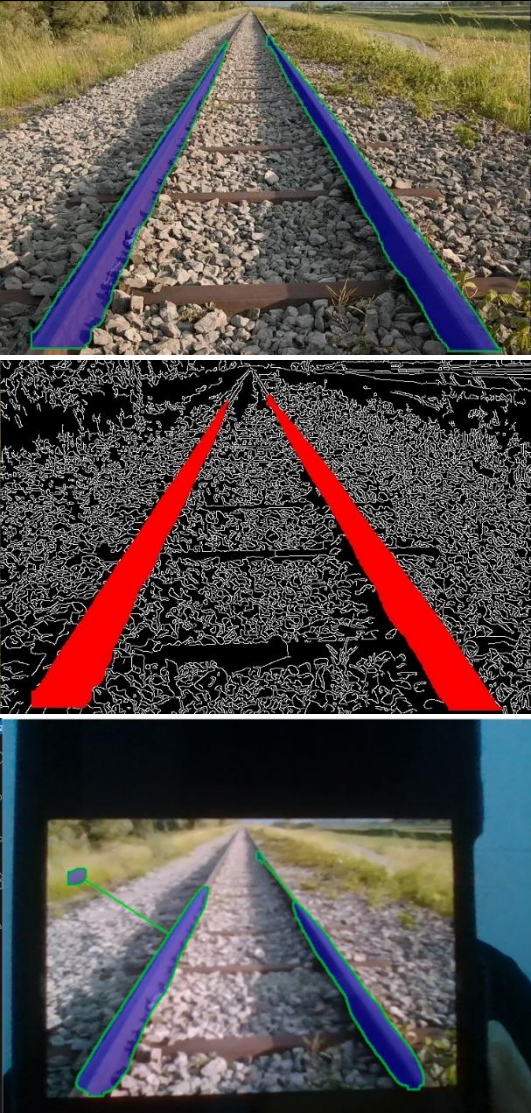
5. Assignment

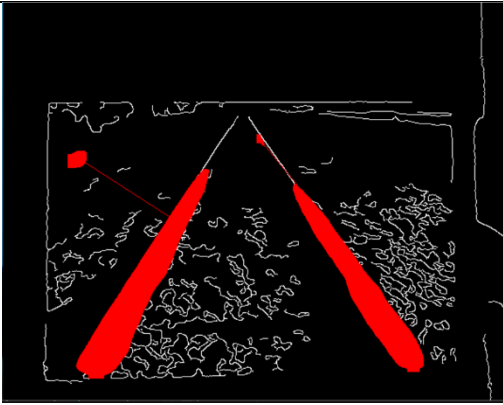
- Studi literatur terkait *canny edge detection*, *lane detection*, dan YOLOV8-Seg.
- Melakukan labelling pada objek rel untuk dijadikan dataset
- Melakukan proses training dari dataset tersebut
- Membuat program untuk *canny edge detection* dan YOLOV8-Seg
- Modifikasi penggunaan threshold untuk mengetahui bagaimana pengaruhnya terhadap hasil deteksi.
- Melakukan run pada program baik menggunakan *webcam*, membaca gambar di satu folder, dan membaca satu gambar.
- Melakukan upload hasil praktikum pada repository Github.

6. Data dan Output Hasil Pengamatan

No	Variabel	Hasil Percobaan	Keterangan
1	Threshold Low=10, High=50		Penggunaan parameter threshold dengan Low=10 dan High = 50 menyebabkan deteksi sangat sensitif, banyak tepi kecil dan <i>noise</i> ikut terdeteksi.

No	Variabel	Hasil Percobaan	Keterangan
			
2	Threshold Low=30 High=120	  	Penggunaan threshold dengan Low=30 dan High = 120 mengurangi sedikit <i>noise</i> walaupun masih banyak tepi yang dideteksi sehingga perlu dioptimalkan penggunaan parameter threshold yang sesuai

No	Variabel	Hasil Percobaan	Keterangan
			
3	Threshold Low=80 High= 200		Penggunaan threshold dengan Low=80 dan High = 200 menyebabkan Deteksi lebih selektif, hanya tepi paling kuat terdeteksi, hal ini menyebabkan detail halus pada objek dihilangkan

No	Variabel	Hasil Percobaan	Keterangan
			

7. Kesimpulan

- Canny Edge Detection* merupakan metode deteksi tepi berbasis gradien yang menggunakan dua nilai ambang batas (*low threshold* dan *high threshold*) untuk memisahkan tepi kuat dan tepi lemah. Proses ini melibatkan tahap *noise reduction*, perhitungan gradien intensitas, *non-maximum suppression*, serta *hysteresis thresholding* untuk memperoleh tepi yang halus dan kontinu.
- Instance Segmentation* adalah metode deteksi objek yang tidak hanya mengenali keberadaan dan posisi objek melalui *bounding box*, tetapi juga memisahkan setiap objek ke dalam area piksel spesifik (*mask*). Dalam konteks *lane detection*, metode ini digunakan untuk mengidentifikasi area jalur rel secara lebih presisi dibandingkan deteksi berbasis bounding box saja.
- Dataset *Rail Segmentation* dari Kaggle merupakan kumpulan citra yang berisi jalur rel kereta dengan berbagai kondisi pencahayaan, latar belakang, dan sudut pandang. Dataset ini digunakan untuk melatih dan menguji model *instance segmentation* agar mampu mengenali serta memisahkan jalur rel dari elemen lingkungan lain di sekitarnya.
- Kombinasi antara *Canny Edge Detection* dan *Instance Segmentation* menghasilkan pendekatan yang lebih komprehensif dalam mendeteksi jalur rel kereta. *Canny Edge Detection* memberikan informasi struktur tepi yang detail, sedangkan *Instance Segmentation* memberikan konteks area objek. Sinergi keduanya memungkinkan deteksi jalur rel yang lebih akurat baik secara bentuk maupun area permukaannya.

8. Saran

- a) Perlu dilakukan pengujian dengan variasi nilai *threshold* yang lebih luas untuk memperoleh batas optimal pada deteksi tepi.
- b) Dapat dikembangkan dengan penggunaan model *instance segmentation* yang lebih kompleks atau berbasis data real-time agar deteksi jalur rel lebih adaptif.
- c) Perlu penambahan tahap *post-processing* untuk mengurangi noise dan memperjelas hasil deteksi gabungan antara Canny dan segmentasi.

9. Daftar Pustaka

- Canny, J., 1986. *A Computational Approach to Edge Detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6), pp.679–698.
- He, K., Gkioxari, G., Dollár, P. and Girshick, R., 2017. *Mask R-CNN*. Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp.2961–2969.
- Redmon, J. and Farhadi, A., 2018. *YOLOv3: An Incremental Improvement*. arXiv:1804.02767 [online]. Available at: <https://arxiv.org/abs/1804.02767> [Accessed 16 October 2025].
- Ultralytics, 2023. *YOLOv8 Documentation*. [online] Available at: <https://docs.ultralytics.com/> [Accessed 16 October 2025].
- Sakaridis, C., Dai, D. and Van Gool, L., 2018. *Semantic Foggy Scene Understanding with Synthetic Data*. International Journal of Computer Vision, 126, pp.973–992.
- Kaggle, 2023. *Rail Segmentation Dataset*. [online] Available at: <https://www.kaggle.com/datasets/rail-segmentation> [Accessed 16 October 2025].