

Analisis Perbandingan Metode Pra Pemrosesan Citra untuk Deteksi Tepi Canny pada Citra Berbagai Kondisi Jalan menggunakan Bahasa Pemrograman Python

Damar Wicaksono^a, Diva Putra Almeyda^b,
Irfan Mikola Muldiyanto Putra^c, Letty Malihatuningrum^d

^{a,b,c,d}*Teknik Elektro, Universitas Tidar, Indonesia*

Corresponding Author:

^a*damar@untidar.ac.id*

ABSTRAK

Deteksi tepi merupakan salah satu teknik penting dalam pengolahan citra digital yang digunakan untuk mengidentifikasi perubahan tiba-tiba dalam intensitas piksel pada suatu objek. Dalam konteks deteksi marka jalan pada jalan raya, deteksi tepi yang akurat memainkan peran krusial dalam meningkatkan keamanan dan navigasi pengendara. Metode Canny Edge Detection telah terbukti efektif dalam mendeteksi tepi dengan akurasi tinggi dalam pengolahan citra digital. Namun, penerapan Canny Edge Detection pada citra jalan dalam berbagai kondisi masih memerlukan penelitian lebih lanjut. Penelitian ini bertujuan untuk mengimplementasikan metode Canny Edge Detection dalam deteksi marka jalan pada citra jalan raya. Tahapan utama penelitian ini meliputi pra pemrosesan citra, di mana noise dihilangkan dan citra dikonversi menjadi citra grayscale untuk mempersiapkan citra sebelum dilakukan deteksi tepi menggunakan metode Canny. Selain itu, akan dilakukan perbandingan dengan beberapa metode pra pemrosesan citra lainnya, seperti median blur dan bilateral blur, untuk mengetahui metode yang paling efektif dalam deteksi tepi pada marka jalan. Berdasarkan hasil penelitian, dapat disimpulkan bahwa penerapan metode Canny Edge Detection dengan pra pemrosesan menggunakan median blur merupakan pendekatan yang efektif dalam deteksi marka jalan pada citra jalan raya. Metode ini mampu menghasilkan deteksi tepi yang akurat dan optimal, yang dapat menjadi dasar untuk pengembangan sistem deteksi marka jalan secara otomatis pada jalan raya. Temuan ini dapat berkontribusi pada peningkatan keamanan dan navigasi pengendara serta pengembangan teknologi deteksi tepi yang lebih akurat dan efektif pada marka jalan.

Kata Kunci : Canny, Pengolahan Citra, Python, OpenCV

ABSTRACT

Edge detection is one of the essential techniques in digital image processing used to identify sudden changes in pixel intensity in an object. In the context of road marking detection on highways, accurate edge detection plays a crucial role in improving motorist safety and navigation. The Canny Edge Detection method has been proven effective in detecting edges with high accuracy in digital image processing. However, applying Canny Edge Detection on road images in various conditions still requires further research. This research aims to implement the Canny Edge Detection method in road marking detection on highway images.

The main stages of this research include image pre-processing, where noise is removed, and the image is converted into a grayscale image to prepare the image before edge detection using the Canny method. In addition, a comparison will be made with several other image pre-processing methods, such as median and bilateral blur, to determine the most effective method for edge detection on road markings. Based on the research results, applying the Canny Edge Detection method with pre-processing using median blur is a practical approach to road marking detection on highway images. This method can produce accurate and optimized edge detection, which can be the basis for developing automatic road marking detection systems on highways. The findings can contribute to the improvement of motorist safety and navigation as well as the development of more accurate and effective edge detection technology on road markings.

Keywords : Canny, Image Processing, Python, OpenCV

PENDAHULUAN

Dewasa ini, minat terhadap bidang Sistem Informasi Geografis (SIG) semakin meningkat seiring dengan peningkatan kualitas internet dan citra satelit. Data jalan raya menjadi penting dalam perencanaan perkotaan yang efisien. Penerapan teknologi pra-pemrosesan citra serta algoritma deteksi tepi, seperti *Canny Edge Detection*, mampu memberikan informasi jalan raya secara optimal, salah satu contoh, ketika bencana melanda, informasi seperti hambatan jalan dapat membantu distribusi bantuan yang efisien (Sobhanmanesh, 2017).

Canny Edge Detection, yang dikembangkan oleh John F. Canny pada tahun 1986, menjadi metode deteksi tepi yang sangat diakui dan akurat dalam pengolahan citra digital (Sadiq, S., & Jaafar, J., 2015). Dalam konteks deteksi tepi pada citra jalan, aplikasi *Canny Edge Detection* bertujuan untuk menemukan metode pra-pemrosesan yang sesuai, menjadi dasar deteksi marka jalan dengan citra yang jelas, serta menghasilkan garis tepi yang halus dan baik. Ini memiliki implikasi besar untuk sistem kendaraan pintar dan pengawasan lalu lintas (Yao, et al., 2019). Keunggulan dari metode *Canny* terletak pada kemampuannya untuk mengurangi *noise* sebelum melakukan perhitungan tepi, sehingga tepi yang dihasilkan menjadi lebih melimpah. Deteksi tepi *Canny* mampu mengidentifikasi tepi yang sebenarnya dengan tingkat error yang minimal dan menghasilkan tepi yang optimal (Zhang et., al., 2020). Dalam penerapan deteksi tepi, tahapan pra pemrosesan citra memiliki peran penting untuk mempersiapkan citra sebelum dilakukan deteksi tepi menggunakan metode *Canny*. Tahapan ini meliputi pengurangan *noise* dan konversi citra berwarna menjadi citra *grayscale* (Arifianto, T., 2017).

Deteksi tepi yang akurat memiliki dampak langsung pada keamanan pengguna jalan (Kumar, et al., 2017). Kesalahan deteksi dapat menyebabkan kesalahan dalam sistem navigasi kendaraan, berpotensi mengakibatkan kecelakaan. Oleh karena itu, penelitian ini berfokus pada deteksi tepi citra jalan dengan metode *Canny Edge Detection*, dengan tujuan meningkatkan keamanan dan navigasi pengendara melalui deteksi tepi yang akurat.

Penelitian ini akan melakukan perbandingan hasil metode *Canny Edge Detection* dengan beberapa metode pra-pemrosesan citra untuk menentukan metode yang paling efektif dalam deteksi tepi pada marka jalan. Diharapkan penelitian ini dapat memberikan kontribusi penting

dalam pengembangan sistem deteksi marka jalan otomatis pada jalan raya, yang pada akhirnya meningkatkan keamanan dan navigasi pengendara.

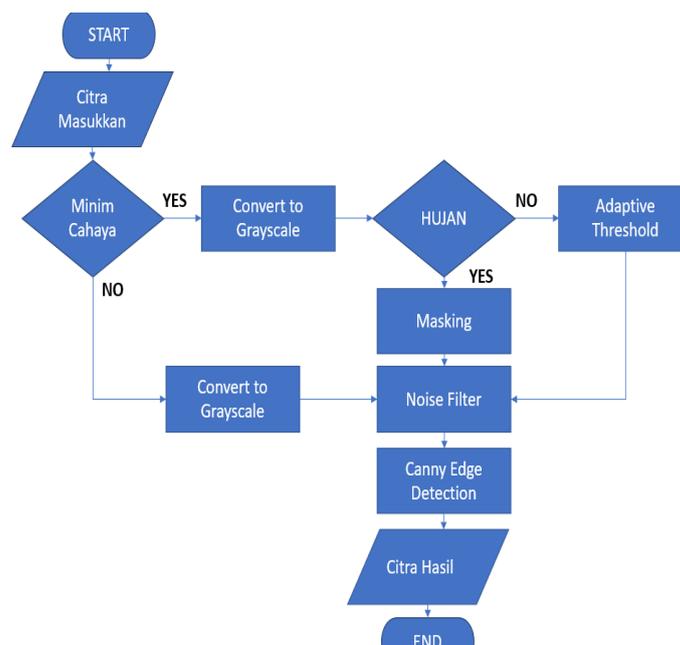
METODE

Subjek

Dalam penelitian ini digunakan data sample citra sejumlah 4 dengan kondisi siang hari cuaca cerah, siang hari cuaca hujan, malam hari cuaca cerah, dan malam hari cuaca hujan. Citra tersebut berekstensi JPG dengan ukuran 1920 x 1080 pixel.

Perancangan Sistem

Sistem kerja penelitian ini digambarkan menggunakan Flowchart yang dapat dilihat pada Gambar 1.



Gambar 1. Flowchart sistem kerja penelitian

Citra Masukkan

Citra yang berjumlah 4 dimasukkan kedalam sistem secara manual menggunakan OpenCV.

Kondisi Citra

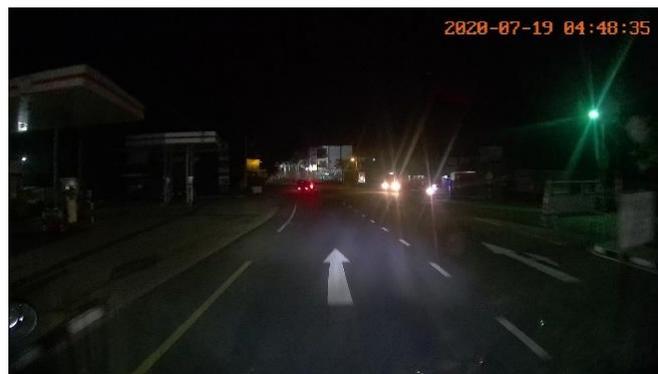
Citra yang digunakan memiliki 4 kondisi siang hari cuaca cerah, siang hari cuaca hujan, malam hari cuaca cerah, dan malam hari cuaca hujan. Ketika cuaca cerah dan penerangan cukup maka pra-pemrosesan citra yang dilakukan hanya pengurangan *noise* dengan *filter* blur, sedangkan ketika cuaca hujan namun memiliki penerangan yang cukup ditambahkan pra-pemrosesan berupa *masking* untuk mengurangi objek yang tidak perlu. Ketika cuaca cerah namun minim pencahayaan (malam), dilakukan pra-pemrosesan tambahan yaitu *Adaptive Threshold* untuk mengatur tingkat pencahayaan citra agar objek yang terdeteksi lebih banyak, sedangkan ketika cuaca hujan dengan penerangan yang kurang, ditambahkan pra-pemrosesan berupa *masking* untuk mengurangi objek yang tidak perlu (Sinaga, A., 2017). Citra sample dapat dilihat pada Gambar 2-5.



Gambar 2. Citra Siang Cerah



Gambar 3. Citra Siang Hujan



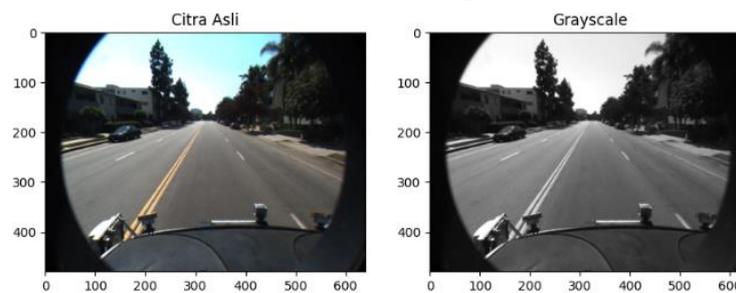
Gambar 4. Citra Malam Cerah



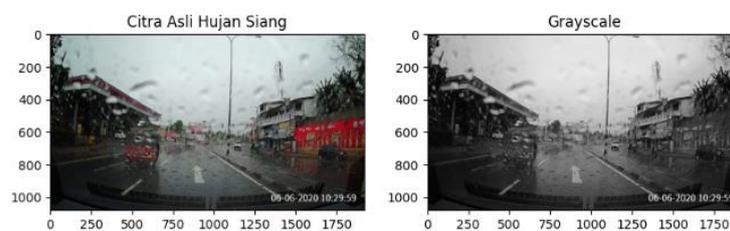
Gambar 5. Citra Malam Hujan

Ubah Citra ke *Grayscale*

Langkah selanjutnya adalah mengubah piksel-piksel citra RGB menjadi citra *Grayscale*. Hal ini dilakukan untuk menyederhanakan model citra yang semula terdiri dari tiga lapisan matriks yaitu R-layer, G-layer, dan B-layer. Jika setiap perhitungan dilakukan pada ketiga lapisan tersebut, maka akan dilakukan tiga perhitungan yang sama, yang cukup merepotkan. Untuk mengatasi hal ini, ketiga lapisan tersebut diubah menjadi satu lapisan matriks *Grayscale*, sehingga menghasilkan citra dengan derajat keabuan. Konversi dari citra berwarna dengan nilai matriks r , g , dan b menjadi citra *Grayscale* dengan nilai s dapat dilakukan dengan mengambil rata-rata dari nilai r , g , dan b . Pada Gambar 6-9, menunjukkan perbandingan citra sebelum diubah ke *Grayscale* dengan citra setelah diubah ke *Grayscale*.



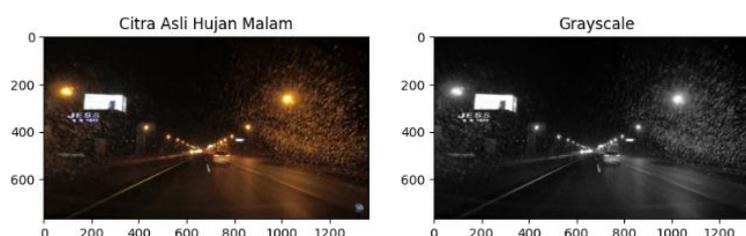
Gambar 6. Citra Siang Cerah *Grayscale*



Gambar 7. Citra Siang Hujan *Grayscale*



Gambar 8. Citra Malam Cerah *Grayscale*



Gambar 9. Citra Malam Hujan *Grayscale*

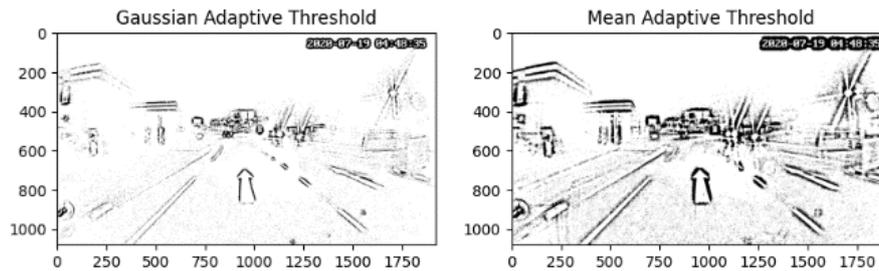
Adaptive Threshold

Thresholding adalah metode segmentasi citra yang memisahkan objek dari latar belakang berdasarkan perbedaan tingkat kecerahan atau kegelapan. Hasilnya adalah citra biner, di mana piksel memiliki nilai intensitas 0 atau 1 (Sulma et., al., 2019).

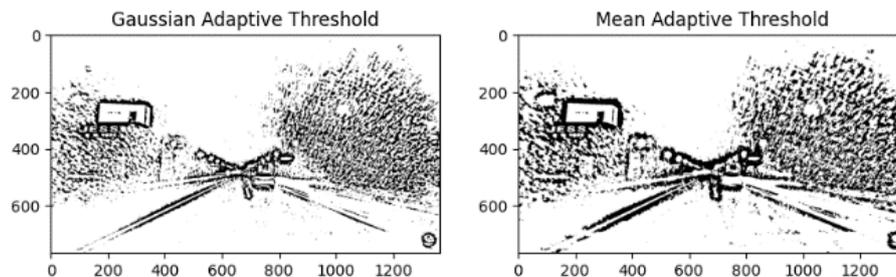
Adaptive Threshold adalah sebuah teknik untuk mengubah citra menjadi citra biner, di mana setiap piksel dalam citra akan diberi nilai hitam atau putih berdasarkan nilai intensitasnya (Saddami et., al., 2018). Metode ini memungkinkan untuk menentukan batas ambang secara adaptif berdasarkan nilai piksel di sekitarnya, sehingga memungkinkan untuk menghasilkan citra biner yang lebih baik dalam mengatasi variasi intensitas piksel pada citra (Supriyatin W., 2020). Dalam penelitian ini, teknik *Adaptive Threshold* digunakan untuk meningkatkan akurasi segmentasi citra dalam memisahkan objek dari latar belakang pada citra yang kompleks. Pada OpenCV penerapan metode ini, memiliki beberapa parameter dan opsi yang dapat diatur, antara lain:

- A. `cv.adaptiveThreshold(src, maxValue, adaptiveMethod, thresholdType, blockSize, C)`
1. `src`: citra *Grayscale* yang akan di-threshold
 2. `maxValue`: nilai piksel maksimum yang akan diberikan ke piksel yang melebihi threshold
 3. `adaptiveMethod`: metode yang digunakan untuk menghitung nilai threshold adaptif. Terdapat dua opsi yang tersedia, yaitu:
 - a. `cv.ADAPTIVE_THRESH_MEAN_C`: nilai threshold adaptif dihitung sebagai rata-rata dari nilai piksel di sekitar blok
 - b. `cv.ADAPTIVE_THRESH_GAUSSIAN_C`: nilai threshold adaptif dihitung sebagai bobot rata-rata dari nilai piksel di sekitar blok dengan bobot Gaussian
 4. `thresholdType`: tipe thresholding yang akan digunakan. Terdapat dua opsi yang tersedia, yaitu:
 - a. `cv.THRESH_BINARY`: nilai piksel di atas nilai threshold akan diberikan nilai maksimum, dan di bawah nilai threshold akan diberikan nilai nol
 - b. `cv.THRESH_BINARY_INV`: nilai piksel di atas nilai threshold akan diberikan nilai nol, dan di bawah nilai threshold akan diberikan nilai maksimum
 5. `blockSize`: ukuran blok yang digunakan untuk menghitung nilai threshold adaptif. Ukuran blok harus ganjil dan lebih besar dari 1.
 6. `C`: konstanta yang dikurangkan dari nilai threshold adaptif. Konstanta ini digunakan untuk mengkompensasi perbedaan intensitas cahaya yang dapat terjadi di seluruh citra.

Perbandingan sebelum proses *Adaptive Threshold* dengan sesudah, dapat dilihat pada Gambar 10 dan Gambar 11.



Gambar 10. Perbandingan Adaptive Method pada Citra Malam Cerah *Grayscale*



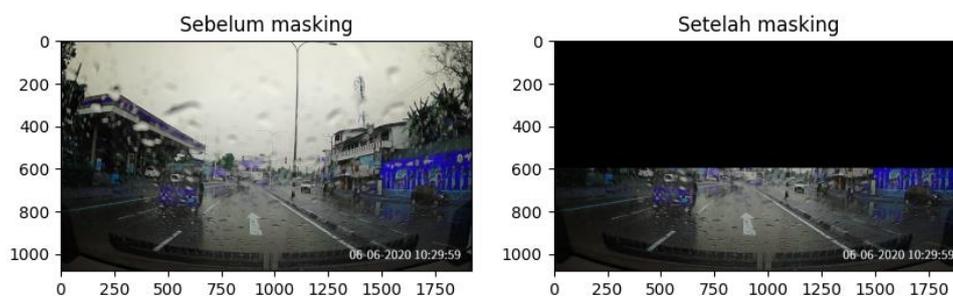
Gambar 11. Perbandingan Adaptive Method pada Citra Malam Hujan *Grayscale*

Masking

Metode *masking* adalah metode dalam pengolahan citra yang digunakan untuk memilih bagian tertentu dari citra yang ingin diproses, sementara bagian lain diabaikan. Hal ini dilakukan dengan membuat sebuah mask atau masker yang berisi nilai-nilai 0 dan 1, dimana nilai 0 menandakan bagian yang diabaikan dan nilai 1 menandakan bagian yang diproses. Masking umumnya digunakan untuk menghilangkan *noise* atau mengambil objek tertentu dalam citra.

Pada OpenCV, kita dapat menggunakan metode *bitwise* untuk melakukan masking. Pertama, kita membuat *mask* dengan nilai 0 dan 1 sesuai dengan bagian yang ingin diproses. Kemudian, kita menggunakan operasi *bitwise* untuk mengambil bagian tertentu dari citra yang sesuai dengan nilai 1 pada *mask*.

Masking dilakukan hanya pada Citra Siang Hujan, karena pada bagian atas citra hanya terdapat bangunan yang nanti akan ikut terdeteksi pada saat proses deteksi tepi.



Gambar 12. *Masking* Citra Siang Hujan

Noise Reduction (Filtering)

Gangguan pada citra digital, seperti kurangnya pencahayaan, keterbatasan resolusi pixel, dan interferensi elektromagnetik, dapat menghasilkan *noise* yang mengurangi kualitas citra,

termanifestasi dalam berbagai bentuk seperti *gaussian*, *speckle*, *salt-and-pepper*, dan *poisson noise* (Wijaya et., al., 2021).

Noise reduction adalah teknik untuk mengurangi *noise* atau gangguan pada citra digital dengan menggunakan *filter* blur (Setyawan, et al., 2022). *Filter* blur bekerja dengan mengaburkan piksel-piksel pada citra sehingga *noise* yang ada di dalam citra menjadi kurang terlihat. *Filter blur* dapat berupa *Gaussian blur*, *Median blur*, *Bilateral filter*, atau *Non-local Means filter*. Setiap *filter* memiliki parameter dan opsi yang berbeda-beda, tergantung pada jenis *noise* yang ingin dihilangkan dan tingkat kehalusan (*smoothness*) yang diinginkan pada citra. *Noise Reduction Filtering Blur* merupakan metode yang sering digunakan dalam berbagai aplikasi seperti pengolahan citra medis, pengenalan wajah, dan pengenalan pola.

1. *Gaussian Blur*

Metode ini menggunakan kernel Gaussian untuk memuluskan citra dengan cara menghilangkan *noise* dan mengurangi detail tajam pada citra. Kernel Gaussian adalah kernel yang menghasilkan bobot tertinggi di tengah kernel dan berkurang secara eksponensial ke arah pinggiran kernel. Semakin besar ukuran kernel (standar deviasi), semakin banyak *noise* yang dihilangkan dan semakin banyak detail yang hilang (Syamsudding, 2019).

2. *Median Blur*

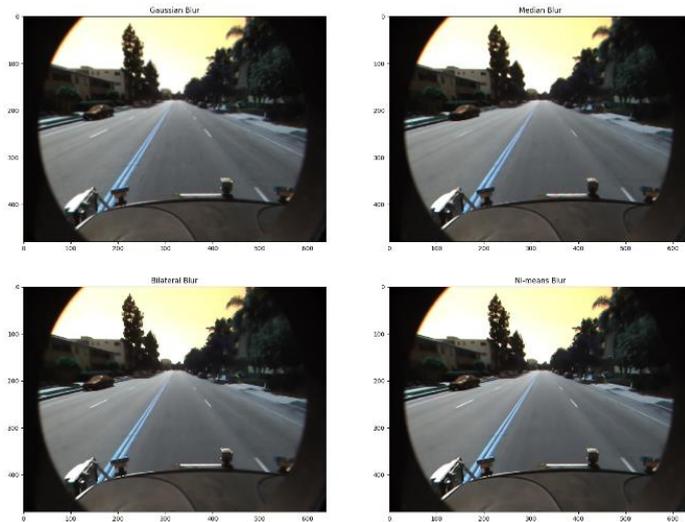
Metode ini menggunakan kernel median untuk memuluskan citra dengan cara menghilangkan *noise* dan mengurangi detail tajam pada citra. Kernel median adalah kernel yang mengambil nilai median dari piksel-piksel di sekitarnya. Metode ini efektif untuk menghilangkan *Noise Salt-and-Pepper*.

3. *Bilateral Filter*

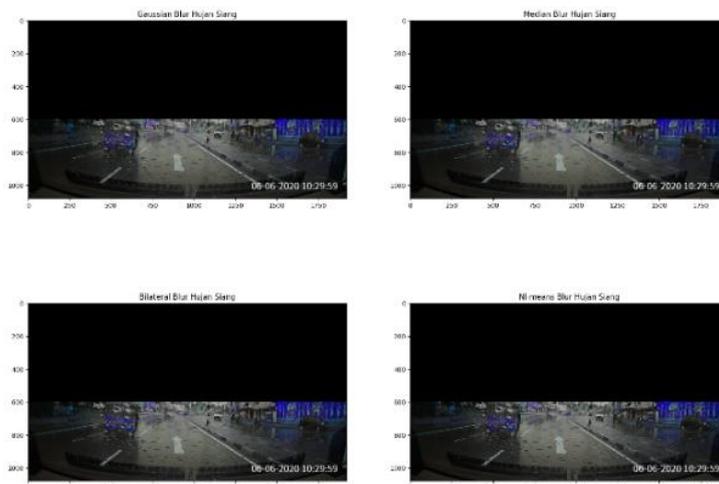
Metode ini menggunakan *filter* bilateral untuk memuluskan citra dengan cara menghilangkan *noise* dan mengurangi detail tajam pada citra. *Filter* bilateral adalah *filter* yang mempertahankan tepi pada citra sambil memuluskan piksel di sekitarnya. *Filter* ini menghitung bobot untuk setiap piksel berdasarkan jarak spasial dan jarak intensitas. Semakin besar jarak spasial dan jarak intensitas, semakin sedikit pengaruh piksel tersebut terhadap piksel yang akan dihilangkan *noise*-nya (Desiani et., al., 2021).

4. *Non-local Means Filter*

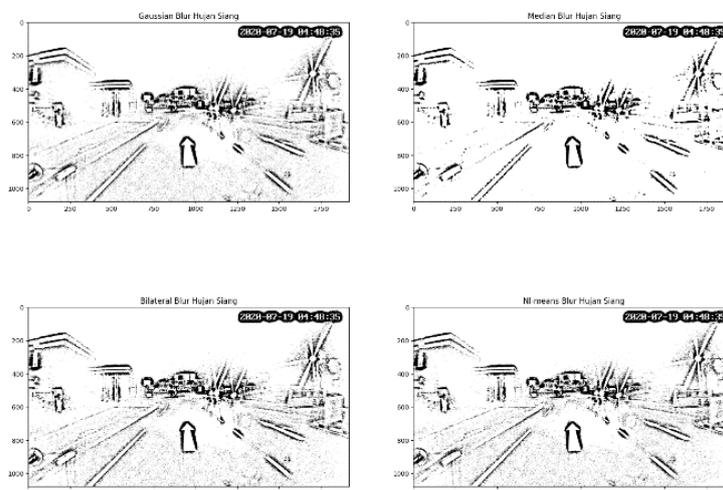
Metode ini menggunakan *filter* non-local means untuk memuluskan citra dengan cara menghilangkan *noise* dan mengurangi detail tajam pada citra. *Filter* non-local means menghitung rata-rata intensitas piksel di sekitarnya berdasarkan kesamaan dengan piksel yang akan dihilangkan *noise*-nya. Metode ini efektif untuk menghilangkan *noise* pada citra berwarna dan tetap mempertahankan detailnya.



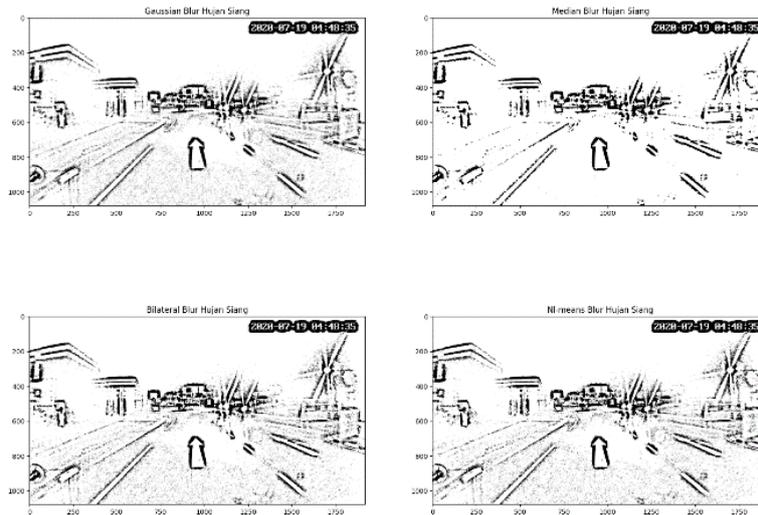
Gambar 13. Filter Blur Citra Siang Cerah



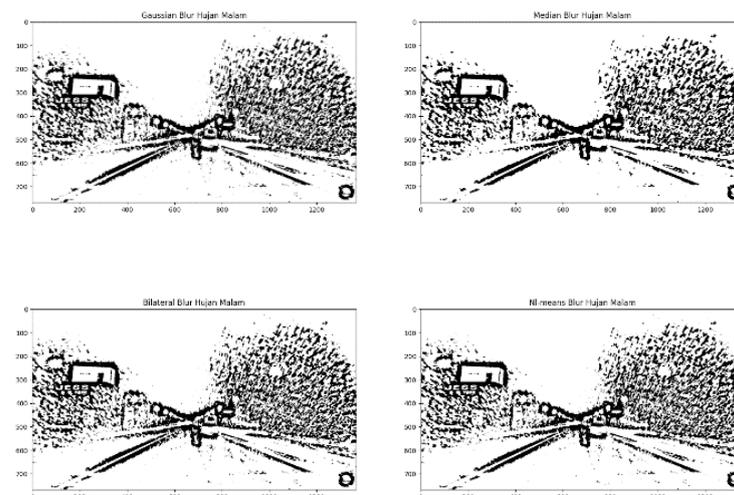
Gambar 14. Filter Blur Citra Siang Hujan



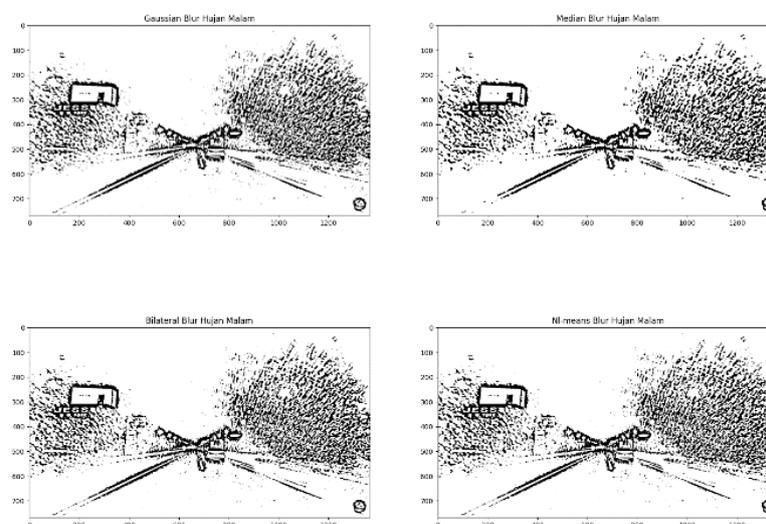
Gambar 15. Filter Blur Citra Malam Cerah dengan Adaptive Method Mean



Gambar 16. Filter Blur Citra Malam Cerah dengan Adaptive Method Gaussian



Gambar 17. Filter Blur Citra Malam Hujan dengan Adaptive Method Mean



Gambar 18. Filter Blur Citra Malam Hujan dengan Adaptive Method Mean

Canny Edge Detection

Proses *Canny Edge Detection* dimulai dengan melakukan *filtering* pada citra asli untuk menghilangkan *noise*. Setelah itu, dilakukan perhitungan gradien pada citra untuk menemukan lokasi perubahan intensitas yang signifikan atau tepi. Kemudian, dilakukan pemilihan tepi dengan menggunakan teknik non-maximum suppression untuk memperoleh tepi yang lebih tepat. Terakhir, dilakukan *thresholding* pada citra untuk memisahkan antara bagian-bagian yang berisi tepi dan yang tidak.

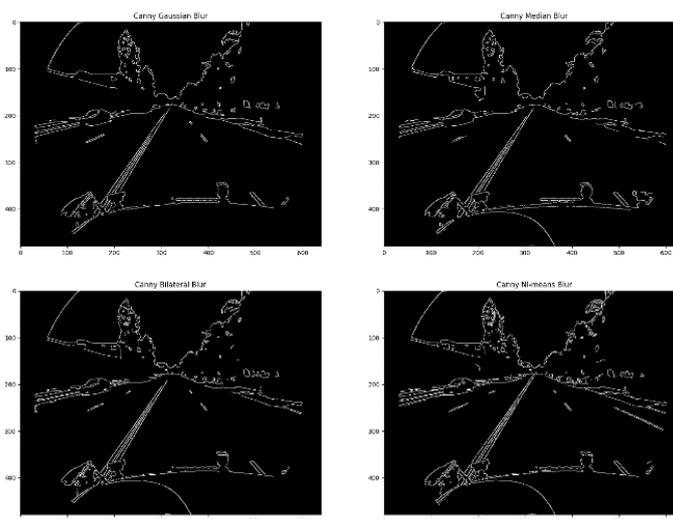
Dalam metode *Canny Edge Detection*, terdapat beberapa parameter yang dapat diatur untuk menghasilkan hasil deteksi tepi yang optimal. Parameter tersebut antara lain nilai *threshold low* dan *high* yang digunakan pada tahap *thresholding*, *kernel size* pada tahap *filtering*, dan *sigma* pada tahap perhitungan gradien. Pemilihan nilai parameter yang tepat sangat penting untuk mendapatkan hasil deteksi tepi yang akurat.

Metode *Canny Edge Detection* pada dasarnya memiliki dua parameter penting yaitu nilai *threshold bawah* (*lower threshold*) dan nilai *threshold atas* (*upper threshold*). *Threshold* adalah nilai batas yang digunakan untuk memisahkan piksel-piksel tepi dari piksel-piksel non-tepi.

HASIL DAN PEMBAHASAN

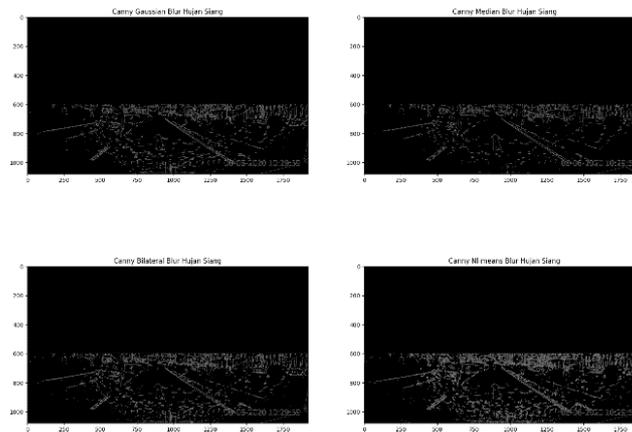
Citra Siang Cerah

Pada Gambar 19, *Canny edge detection* dengan *filter* *NI-means Blur* lebih unggul dalam hal pendeteksian marka jalan karena marka jalan yang terdeteksi lebih panjang dibanding menggunakan *filter* blur yang lain. Selain itu pada marka jalan yang pendek juga lebih jelas terdeteksi bentuknya dibanding pada *filter* blur yang lain.



Gambar 19. Canny Edge Detection Citra Siang Cerah

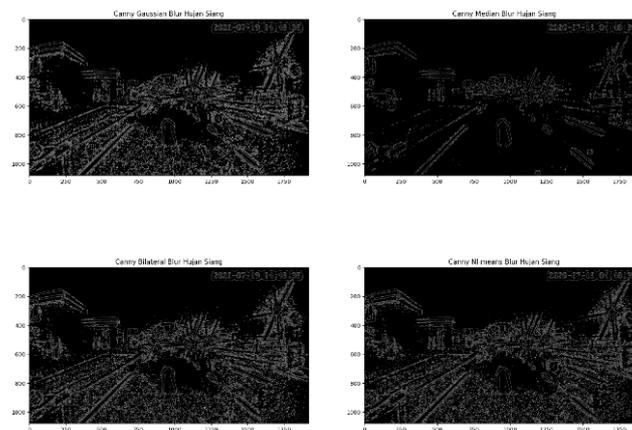
Citra Siang Hujan



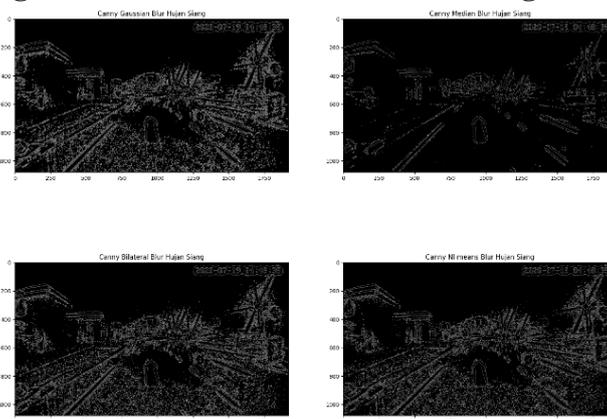
Gambar 20. *Canny* Edge Detection Citra Siang Hujan

Penerapan *Canny* untuk Citra Siang Hujan berfokus pada keseimbangan, yaitu untuk mengurangi *noise* namun juga tetap menjaga kualitas objek deteksi tepi. Pada Gambar 20, ditunjukkan hasil yang paling seimbang yaitu pada *Canny* Edge Detection dengan *filter* blur Median Blur, karena marka jalan tetap dapat dilihat bentuknya juga *noise* titik hujan berkurang signifikan jika dibandingkan dengan *filter* blur yang lain.

Citra Malam Cerah



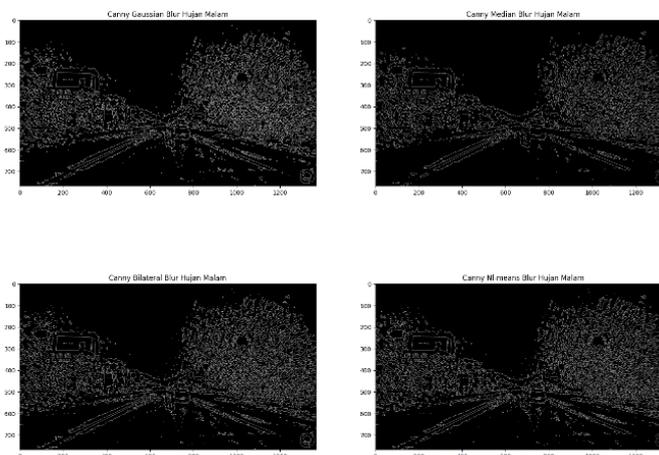
Gambar 21. *Canny* Edge Detection Citra Malam Cerah dengan Adaptive Method Mean



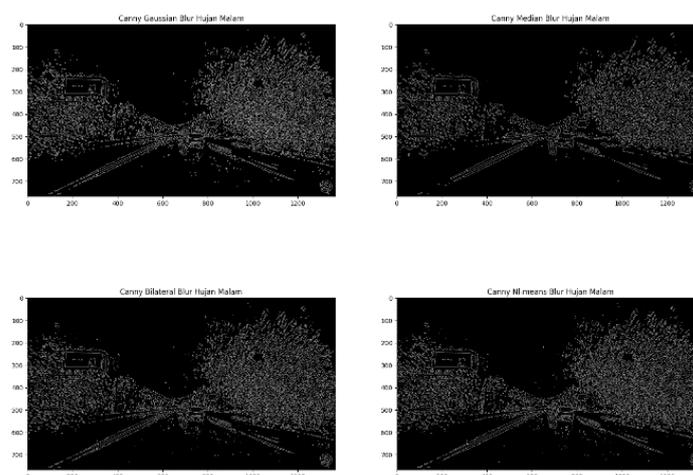
Gambar 22. *Canny* Edge Detection Citra Malam Cerah dengan Adaptive Method Gaussian

Penerapan *Canny* untuk Citra Malam Cerah berfokus pada pengaturan tingkat kecerahan dan keseimbangan untuk mengurangi *noise* akibat *Adaptive Threshold* namun juga tetap menjaga kualitas objek deteksi tepi. Pada Gambar 21 dan Gambar 22, ditunjukkan hasil yang paling baik yaitu pada *Canny Edge Detection* dengan *filter* blur Median Blur dan Adaptive Method Gaussian, karena marka jalan tetap dapat dilihat bentuknya juga *noise* cahaya akibat *Adaptive Threshold* juga berkurang signifikan dibandingkan yang lain.

Citra Malam Hujan



Gambar 23. *Canny Edge Detection* Citra Malam Hujan dengan Adaptive Method Mean



Gambar 24. *Canny Edge Detection* Citra Malam Hujan dengan Adaptive Method Gaussian

Penerapan *Canny* untuk Citra Malam Hujan berfokus pada pengaturan tingkat kecerahan dan keseimbangan untuk mengurangi *noise* akibat *Adaptive Threshold* namun juga tetap menjaga kualitas objek deteksi tepi. Pada Gambar 23 dan Gambar 24, ditunjukkan hasil yang paling baik pada *Canny Edge Detection* dengan *filter* Bilateral Blur dan Adaptive Method Gaussian. Akan tetapi kurang memuaskan karena *noise* titik hujan yang tidak terlalu signifikan berkurang selain itu bentuk marka jalan masih dapat dikatakan kurang jelas jika dibandingkan citra-citra pada kondisi sebelumnya.

SIMPULAN

Berdasarkan hasil dan pembahasan analisa penerapan deteksi tepi Canny dengan berbagai tahap pra-pemrosesan citra untuk berbagai kondisi, dapat disimpulkan bahwa:

- a. Jika citra yang digunakan memiliki *noise* yang minim, dapat digunakan *filter* NI-mean blur, karena hasil deteksi tepi lebih detail dengan kontras tinggi
- b. Jika citra memiliki *noise* yang tinggi dan ingin fokus pada pengurangan *noise*, gunakan *filter* Median Blur karena dihasilkan citra dengan *noise* rendah yang signifikan dibandingkan *filter* blur yang lain
- c. Untuk citra yang minim intensitas cahaya, diterapkan metode Adaptive Threshold agar objek-objek yang sebelumnya tidak dapat terdeteksi menjadi dapat terdeteksi

DAFTAR PUSTAKA

- Arifianto, T. (2017). Segmentasi Aksara pada Tulisan Aksara Jawa Menggunakan Adaptive Threshold. *SMATIKA Jurnal*, 7(1), 01-05
- Kumar, S. N., Sukanya, B., Mohan, B., Prathibha, G., (2017). Extraction of Roads from Satellite Images Based on Edge Detection. *IJEDR*, 5(2), 187-190.
- Saddami, K., Away, Y., Munadi, K., & Arnia, F. (2017). Perbandingan Metode Penekanan Noise Lokal dan Non-Lokal untuk Binerisasi Dokumen Jawi Kuno. Seminar Nasional dan Expo Teknik Elektro 2018, 107-112.
- Sadiq, S., & Jaafar, J. (2015). Road edge detection using Hough transform and Canny operator. *International Journal of Computer Science and Mobile Computing*, 4(5), 684-688.
- Setyawan, G. C., & Nawansari, M. P. (2022). Gaussian and Median Screening Performance In Image Softening. *Journal of Information Technology*, 2(2), 1-4.
- Sinaga, A. (2017). Implementasi Teknik Threshoding Pada Segmentasi Citra Digital. *Jurnal Mantik Penusa*, 1(2), 48.
- Sinurat, S., & Siagian, E. R. (2021). Peningkatan Kualitas Citra Dengan Gaussian Filter Terhadap Citra Hasil Deteksi Robert. *Pelita Informatika: Informasi dan Informatika*, 9(3), 225-231.
- Sobhanmanesh, F., & Gholami, A. (2017). An overview of edge detection techniques for road extraction from satellite images. *Journal of Remote Sensing & GIS*, 6(3), 1-12.
- Sulma, S., Rahmi, K., Febrianti, N., Sitorus, J. (2019). Deteksi Tumpahan Minyak Menggunakan Metode Adaptive Threshold Dan Analisis Tekstur Pada Data SAR. *Majalah Ilmiah Globe*, 21(1), 45-52
- Supriyatin, W. (2020). Perbandingan Metode Sobel, Prewitt, Robert dan Canny pada Deteksi Tepi Objek Bergerak. *ILKOM Jurnal Ilmiah*, 12(2), 112–120.
- Syamsudding, S. (2019). Aplikasi Peningkatan Kualitas Citra Menggunakan Metode Median *Filtering* Untuk Menghilangkan *Noise*. *Prosiding Seminar Ilmiah Sistem Informasi dan Teknologi Informasi*, 8(1), 227-236.
- Wijaya, P. H., Wulaningrum, R., & Halilintar, Risa. (2021). Perbaikan Citra Dengan Menggunakan Metode Gaussian Dan Mean *Filter*. *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)*, 5(2), 100–105.

- Yao, J., Chen, W., & Liu, Y. (2019). Real-time road marking detection and tracking with deep learning. *IEEE Transactions on Intelligent Transportation Systems*, 20(5), 1643-1653.
- Zhang, S., Xiang, S., & Chen, L. (2020). A novel road detection method based on improved Canny operator and region growing. *IEEE Access*, 8, 76033-76044.