

DETEKSI TEPI PADA CITRA OBJEK BENDA MENGGUNAKAN ALGORITMA SOBEL DAN PREWITT DENGAN PYTHON

M Riski Qisthiano^{1*}, Ayu Okta Pratiwi²

¹Institut Teknologi dan Bisnis Nasional (ITBN); Banyuasin, Sumatera Selatan

²Institusi Teknologi Dan Sains Nahdlatul Ulama Sriwijaya; Palembang, Sumatra Selatan

Received: 8 Maret 2025
Accepted: 27 Maret 2025
Published: 14 April 2025

Keywords:

Deteksi Tepi,
Algoritma Sobel,
Algoritma Prewitt,
Pengolahan Citra,
Python.

Correspondent **Email:**
riskiqisthiano@itbn.ac.id

Abstrak. Deteksi tepi merupakan salah satu teknik penting dalam pengolahan citra digital untuk mengekstraksi informasi bentuk dan struktur suatu objek. Penelitian ini bertujuan untuk menganalisis kinerja algoritma Sobel dan Prewitt dalam mendeteksi tepi pada citra objek benda. Permasalahan yang dihadapi adalah bagaimana efektivitas kedua algoritma dalam menghasilkan deteksi tepi yang optimal pada berbagai jenis citra. Metode yang digunakan dalam penelitian ini meliputi pemrosesan citra dengan Python, penerapan algoritma Sobel dan Prewitt, serta evaluasi hasil berdasarkan kualitas deteksi tepi yang dihasilkan. Hasil penelitian menunjukkan bahwa algoritma Sobel menghasilkan tepi yang lebih tajam dengan kontras tinggi, sedangkan algoritma Prewitt lebih sensitif terhadap perubahan intensitas tetapi memiliki respon yang lebih lemah dibandingkan Sobel. Berdasarkan hasil eksperimen, algoritma Sobel lebih disarankan untuk citra dengan kontras tinggi, sementara algoritma Prewitt lebih sesuai untuk citra dengan gradasi halus. Penelitian ini diharapkan dapat memberikan wawasan bagi pengembangan lebih lanjut dalam analisis citra digital, terutama dalam aplikasi deteksi objek dan pengolahan gambar berbasis kecerdasan buatan.

Abstract. Edge detection is one of the important techniques in digital image processing to extract information about the shape and structure of an object. This study aims to analyze the performance of the Sobel and Prewitt algorithms in detecting edges in object images. The problem faced is how effective the two algorithms are in producing optimal edge detection on various types of images. The methods used in this study include image processing with Python, the application of the Sobel and Prewitt algorithms, and evaluation of the results based on the quality of the resulting edge detection. The results show that the Sobel algorithm produces sharper edges with high contrast, while the Prewitt algorithm is more sensitive to changes in intensity but has a weaker response than Sobel. Based on the experimental results, the Sobel algorithm is more recommended for images with high contrast, while the Prewitt algorithm is more suitable for images with smooth gradations. This study is expected to provide insight for further development in digital image analysis, especially in the application of object detection and image processing based on artificial intelligence.

1. PENDAHULUAN

Di era digital saat ini telah berkembang teknologi – teknologi yang digunakan masyarakat, terutama kecerdasan buatan[1].

Deteksi tepi merupakan salah satu tahap fundamental dalam pengolahan citra digital yang bertujuan untuk mengidentifikasi batas atau kontur objek dalam sebuah gambar. Teknik

ini sangat penting karena memungkinkan ekstraksi fitur-fitur utama seperti bentuk, kontur, dan struktur objek, yang merupakan dasar bagi aplikasi-aplikasi lanjut seperti segmentasi citra, pengenalan objek, dan analisis pola[2][3].

Gambar berfungsi sebagai media komunikasi untuk menyampaikan pesan tertentu dari pembuatnya. Dalam bidang teknik, gambar berperan sebagai bahasa yang menyampaikan informasi, menjadikannya elemen penting dalam proses perancangan dan perencanaan[4]. Gambar teknik adalah representasi visual dari suatu objek desain atau sistem yang dibuat menggunakan standar simbol dan notasi khusus, sehingga dapat dipahami oleh perancang dan pelaksana. Oleh karena itu, gambar diharapkan mampu menyampaikan data dan informasi dengan jelas serta akurat. Informasi yang disampaikan dalam gambar tidak dapat disampaikan secara lisan, melainkan harus menggunakan simbol, notasi, garis, dan tulisan yang bersifat tegas sesuai dengan aturan keteknikan yang telah disepakati bersama[5].

Berbagai metode telah dikembangkan untuk melakukan deteksi tepi, di antaranya algoritma Sobel dan Prewitt yang dikenal karena kesederhanaan serta efektivitasnya. Algoritma Sobel bekerja dengan menghitung gradien intensitas piksel melalui operasi konvolusi menggunakan kernel tertentu, sehingga dapat menonjolkan area dengan perubahan intensitas tinggi. Sedangkan, algoritma Prewitt menggunakan kernel yang berbeda dan memberikan sensitivitas yang bervariasi dalam mendeteksi gradien[6].

Sejumlah penelitian telah membandingkan kedua algoritma tersebut. Misalnya, sebuah studi oleh Rumah Coding menunjukkan bahwa Sobel cenderung menghasilkan tepi yang lebih tajam pada citra dengan kontras tinggi, sedangkan Prewitt lebih responsif terhadap perubahan gradasi halus. Lebih lanjut, survei menyeluruh mengenai teknik deteksi tepi yang dilakukan oleh MDPI memberikan gambaran komprehensif tentang berbagai metode deteksi tepi, termasuk Sobel dan Prewitt.

Penelitian lain yang diterbitkan di ScienceDirect mengkaji perbandingan kinerja berbagai teknik deteksi tepi, serta menyoroti kelebihan dan keterbatasan masing-masing metode dalam konteks pengolahan citra digital.

Demikian pula, analisis komparatif yang dipaparkan dalam ResearchGate memberikan wawasan mendalam mengenai keakuratan dan efisiensi algoritma dalam berbagai kondisi pencahayaan dan noise. Sebuah ulasan terkini di MDPI juga menegaskan pentingnya pemilihan algoritma yang tepat berdasarkan karakteristik citra yang diolah.

Dengan semakin berkembangnya teknologi, implementasi deteksi tepi menggunakan bahasa pemrograman Python melalui pustaka OpenCV menjadi pilihan populer karena kemudahan dalam pengembangan dan fleksibilitasnya. Oleh karena itu, penelitian ini difokuskan pada analisis kinerja algoritma Sobel dan Prewitt dalam mendeteksi tepi pada citra objek benda, dengan harapan dapat memberikan rekomendasi yang bermanfaat bagi pengembangan aplikasi pengolahan citra di masa depan.

2. TINJAUAN PUSTAKA

2.1. *Python*

Python adalah bahasa pemrograman yang sangat direkomendasikan untuk analisis data. Dalam beberapa tahun terakhir, dukungan pustaka Python yang semakin berkembang, terutama Pandas, telah menjadikannya alternatif yang kuat untuk tugas analisis data. Pandas memungkinkan pengguna untuk mengolah, menganalisis, dan memvisualisasikan data dengan mudah, sehingga mempermudah proses analisis data[7]. Selain itu, Python juga merupakan bahasa pemrograman multiplatform yang banyak digunakan dalam aplikasi teknologi saat ini dan mendatang, khususnya dalam bidang Internet of Things (IoT). Fleksibilitas dan kemudahan penggunaannya membuat Python ideal untuk pengembangan prototipe dan analisis data dalam proyek IoT[8].

2.2. *OpenCV*

OpenCV merupakan pustaka open-source yang dikembangkan secara khusus untuk mendukung berbagai proses pengolahan citra digital dan computer vision[9]. Pustaka ini menyediakan beragam algoritma pemrosesan citra, seperti transformasi geometris, filtering, serta berbagai teknik lainnya yang telah dioptimalkan agar bekerja secara efisien dan cepat. Dibandingkan dengan pustaka lain

seperti PIL atau SciPy dalam Python, OpenCV memiliki keunggulan dalam mendukung pemrosesan real-time serta dilengkapi modul yang dioptimalkan untuk berbagai jenis perangkat keras. Selain itu, OpenCV juga menawarkan fitur deteksi objek menggunakan metode computer vision, menjadikannya pilihan utama dalam pengembangan aplikasi berbasis visi komputer[10].

2.3. Algoritma Sobel

Algoritma Sobel adalah metode deteksi tepi dalam pengolahan citra digital yang memanfaatkan operasi konvolusi untuk menghitung gradien intensitas piksel dalam arah horizontal dan vertikal. Dengan menggunakan dua kernel berukuran 3x3, algoritma ini menyoroti area dengan perubahan intensitas tinggi, yang biasanya menunjukkan keberadaan tepi dalam citra. Keunggulan utama dari metode Sobel adalah kemampuannya dalam mengurangi noise sebelum melakukan perhitungan deteksi tepi, sehingga menghasilkan deteksi tepi yang lebih halus dan akurat[11].

2.4. Algoritma Prewitt

Algoritma Prewitt mirip dengan Sobel, namun menggunakan kernel yang berbeda untuk menghitung gradien dalam arah horizontal dan vertikal. Kernel Prewitt juga berukuran 3x3 dan dirancang untuk menekankan perubahan intensitas di sepanjang sumbu tertentu. Setelah konvolusi dengan kedua kernel, magnitudo gradien dihitung dengan cara yang sama seperti pada metode Sobel. Meskipun metode Prewitt lebih sederhana dan memiliki perhitungan yang lebih cepat, metode ini cenderung lebih sensitif terhadap noise dibandingkan dengan metode Sobel[6].

2.5. Deteksi Tepi

Deteksi tepi merupakan salah satu teknik segmentasi gambar yang bertujuan untuk mengidentifikasi garis tepi suatu objek dalam citra, yang berfungsi untuk menandai bagian-bagian penting dari gambar tersebut. Selain itu, proses ini juga berperan dalam meningkatkan ketajaman detail pada citra yang mengalami keaburan akibat kesalahan atau efek dari proses akuisisi gambar[12]. Dalam penerapannya, metode deteksi tepi (*edge*

detection) mengubah area tertentu dalam citra menjadi dua nilai intensitas, yaitu rendah dan tinggi, yang biasanya direpresentasikan dalam bentuk biner, seperti nol dan satu[13].

3. Metodologi Penelitian

Penelitian ini bertujuan untuk mendeteksi tepi pada citra objek benda menggunakan algoritma Sobel dan Prewitt dengan bahasa pemrograman Python. Metode penelitian terdiri dari beberapa tahap utama, yaitu pengumpulan data, preprocessing citra, penerapan algoritma deteksi tepi, dan analisis hasil.

3.1. Pengumpulan Data

Tahap awal dalam penelitian ini adalah pengumpulan data citra objek benda yang akan digunakan dalam eksperimen. Data citra diambil dari berbagai sumber, seperti dataset yang tersedia di platform open-source seperti Kaggle dan GitHub, serta dari hasil pemotretan langsung menggunakan kamera digital. Format citra yang digunakan dalam penelitian ini adalah JPEG dan PNG, yang umum digunakan dalam pemrosesan citra digital. Pemilihan dataset dilakukan dengan mempertimbangkan variasi objek dan pencahayaan untuk menguji efektivitas algoritma deteksi tepi.

3.2. Preprocessing Data

Setelah data citra dikumpulkan, dilakukan tahap preprocessing untuk memastikan bahwa citra siap digunakan dalam proses deteksi tepi. Preprocessing bertujuan untuk meningkatkan kualitas citra dengan menghilangkan noise dan memastikan bahwa citra dalam format yang sesuai untuk pemrosesan lebih lanjut. Berikut adalah tahapan preprocessing yang dilakukan:

1. Konversi ke Grayscale

Citra berwarna dikonversi menjadi citra grayscale agar hanya mempertahankan informasi intensitas cahaya. Langkah ini dilakukan untuk menyederhanakan analisis citra, karena informasi warna tidak diperlukan dalam deteksi tepi.

2. Normalisasi Intensitas Piksel

Setelah konversi ke grayscale, dilakukan normalisasi untuk memastikan bahwa nilai piksel berada dalam rentang yang optimal. Hal ini bertujuan untuk menghindari

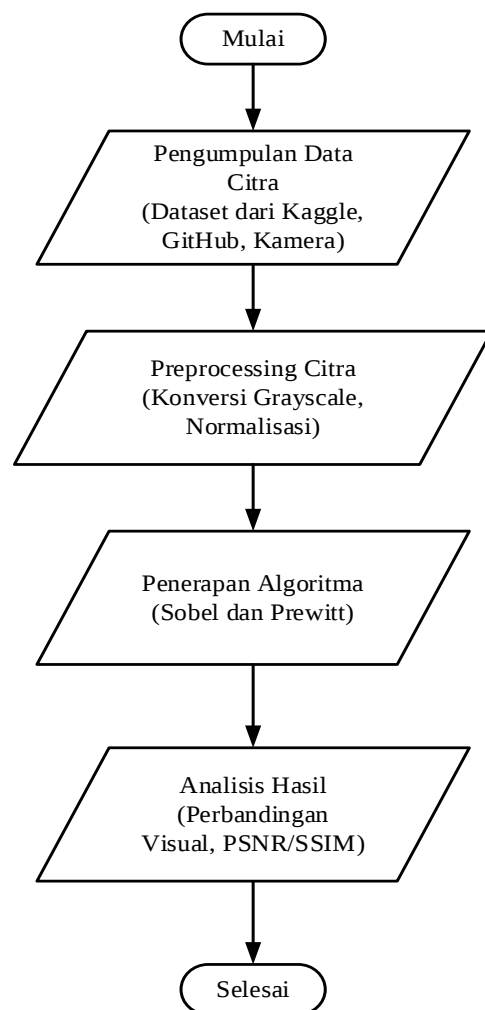
- perbedaan skala intensitas yang dapat mempengaruhi hasil deteksi tepi.
3. Penerapan Filter Pereduksi Noise
Noise dalam citra dapat mengganggu proses deteksi tepi, sehingga dilakukan proses penyaringan menggunakan filter seperti Gaussian blur. Teknik ini membantu mengurangi gangguan dari detail kecil yang tidak relevan tanpa menghilangkan tepi utama objek.
 4. Peningkatan Kontras Citra
Untuk meningkatkan kejelasan tepi objek, dilakukan peningkatan kontras menggunakan teknik seperti histogram equalization. Metode ini membantu dalam meratakan distribusi intensitas cahaya sehingga struktur objek lebih jelas terlihat.
 5. Deteksi dan Eliminasi Noise Tambahan
Jika masih terdapat noise setelah tahap sebelumnya, diterapkan filter tambahan seperti median filter untuk menghilangkan gangguan tanpa mengaburkan tepi objek

3.3. Penerapan Algoritma Deteksi Tepi

Pada tahap ini, algoritma deteksi tepi Sobel dan Prewitt diterapkan untuk mengidentifikasi batas-batas objek dalam citra. Algoritma Sobel bekerja dengan menghitung gradien intensitas piksel dalam arah horizontal dan vertikal menggunakan operator diferensial pertama. Hasil gradien kemudian dikombinasikan untuk membentuk peta tepi. Sementara itu, algoritma Prewitt memiliki konsep serupa tetapi menggunakan kernel yang berbeda, menghasilkan deteksi tepi yang lebih halus dibandingkan Sobel. Kedua algoritma diimplementasikan menggunakan pustaka OpenCV dalam Python untuk memastikan efisiensi dan akurasi pemrosesan citra.

3.4 Flowchart Sistem

Flowchart sistem menggambarkan alur kerja proses deteksi tepi menggunakan algoritma Sobel dan Prewitt. Diagram ini menunjukkan tahapan utama dalam sistem, mulai dari input citra hingga hasil deteksi tepi.



Gambar 1. Alur Sistem

1. Pengumpulan Data Citra
Tahap awal dalam sistem ini adalah pengumpulan data citra yang akan digunakan dalam proses deteksi tepi. Sumber data dapat berasal dari dataset yang telah tersedia di platform seperti Kaggle atau GitHub, serta citra yang diambil langsung menggunakan kamera. Pemilihan dataset yang berkualitas akan berpengaruh terhadap hasil deteksi tepi yang dihasilkan.
2. Preprocessing Citra
Setelah data citra dikumpulkan, dilakukan tahap preprocessing untuk memastikan citra siap digunakan dalam analisis lebih lanjut. Proses ini mencakup konversi citra berwarna menjadi grayscale, yang bertujuan untuk menyederhanakan informasi dengan hanya mempertahankan intensitas cahaya. Selain itu, dilakukan

normalisasi intensitas piksel untuk memastikan nilai piksel berada dalam rentang yang optimal, sehingga dapat meningkatkan keakuratan deteksi tepi.

3. Penerapan Algoritma (Sobel dan Prewitt)

Pada tahap ini, algoritma deteksi tepi Sobel dan Prewitt diterapkan pada citra hasil preprocessing. Algoritma Sobel bekerja dengan menghitung gradien intensitas piksel dalam arah horizontal dan vertikal untuk mendeteksi tepi dengan lebih halus. Sementara itu, algoritma Prewitt juga menggunakan metode gradien, tetapi lebih sederhana dibandingkan Sobel. Kedua algoritma ini dibandingkan untuk menilai efektivitas masing-masing dalam mendeteksi tepi objek pada citra.

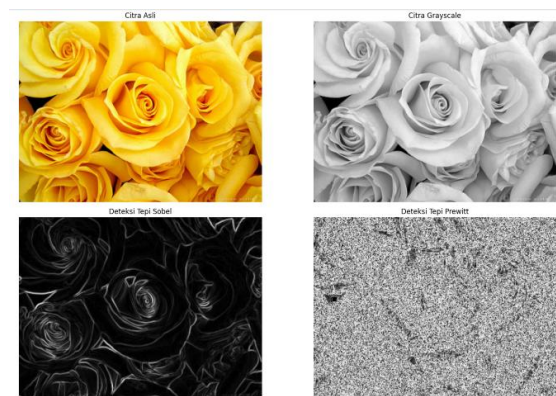
4. Analisis Hasil

Setelah algoritma diterapkan, dilakukan analisis hasil deteksi tepi. Analisis dilakukan secara visual dengan membandingkan kualitas tepi yang dihasilkan oleh masing-masing algoritma. Selain itu, metrik seperti Peak Signal-to-Noise Ratio (PSNR) dan Structural Similarity Index (SSIM) dapat digunakan untuk mengukur seberapa baik hasil deteksi dibandingkan dengan citra asli. Analisis ini bertujuan untuk menentukan algoritma mana yang memberikan hasil terbaik dalam deteksi tepi pada citra yang diuji

4. HASIL DAN PEMBAHASAN

4.1. Hasil Deteksi Tepi

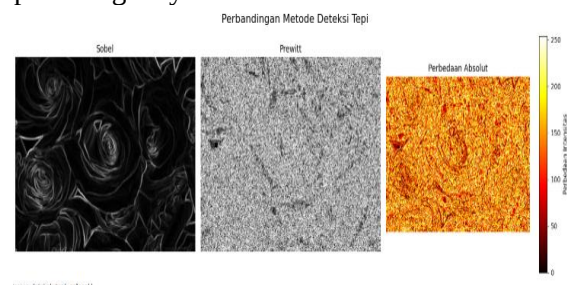
Penelitian ini bertujuan untuk mendeteksi tepi pada citra objek benda menggunakan algoritma Sobel dan Prewitt. Setelah melalui tahap preprocessing, yaitu konversi grayscale dan normalisasi, citra kemudian diproses menggunakan kedua algoritma tersebut. Hasil deteksi tepi dari masing-masing algoritma dibandingkan untuk melihat efektivitas serta kualitas hasil yang dihasilkan.



Gambar 2. Hasil Deteksi

4.2. Perbandingan Algoritma Sobel dan Prewitt

Algoritma Sobel dan Prewitt memiliki pendekatan yang serupa dalam mendeteksi tepi, namun terdapat perbedaan dalam metode penghitungan gradiennya. Sobel menggunakan koefisien bobot yang lebih besar, sehingga menghasilkan deteksi tepi yang lebih tajam dibandingkan Prewitt. Sementara itu, Prewitt lebih sensitif terhadap perubahan intensitas piksel tetapi lebih sederhana dalam perhitungannya.



Gambar 3. Hasil Perbandingan Deteksi

Gambar ini menunjukkan perbandingan hasil deteksi tepi menggunakan metode Sobel dan Prewitt, serta perbedaan absolut antara keduanya.

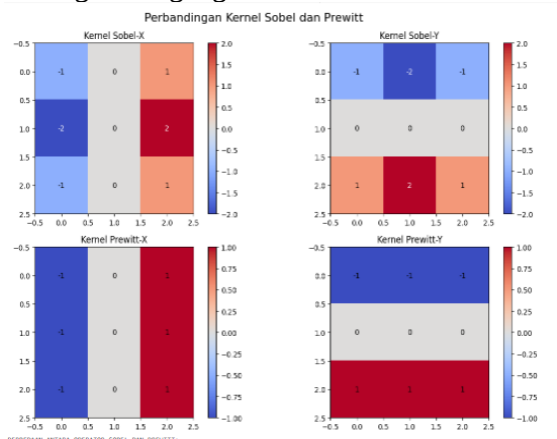
1. Hasil Sobel (kiri) menunjukkan tepi yang lebih tajam dan jelas dengan kontras tinggi. Sobel lebih sensitif terhadap perubahan intensitas dan memberikan hasil yang lebih halus.
2. Hasil Prewitt (tengah) tampak lebih berisik dengan detail yang kurang jelas dibandingkan Sobel. Ini karena Prewitt menggunakan bobot konvolusi yang lebih sederhana.
3. Perbedaan Absolut (kanan) menggambarkan perbedaan antara hasil Sobel dan Prewitt. Warna terang

menunjukkan perbedaan yang lebih besar, sedangkan warna gelap menunjukkan kesamaan hasil antara kedua metode.

Untuk mengevaluasi performa kedua algoritma, dilakukan analisis berdasarkan aspek visual dan pengukuran objektif seperti Peak Signal-to-Noise Ratio (PSNR) dan Structural Similarity Index Measure (SSIM).

4.3. Evaluasi Hasil

Hasil analisis menunjukkan bahwa algoritma Sobel memberikan hasil deteksi tepi yang lebih jelas dengan kontras yang lebih tinggi dibandingkan Prewitt. Namun, Prewitt lebih cepat dalam komputasi karena bobot konvolusinya lebih ringan. Dari hasil perhitungan PSNR dan SSIM, diperoleh nilai yang menunjukkan kualitas deteksi tepi dari masing-masing algoritma.



Gambar 4. Hasil Perbandingan Kernel Sobel dan Prewitt

Gambar ini menunjukkan perbandingan kernel Sobel dan Prewitt dalam mendeteksi tepi pada sumbu X dan Y.

1. Kernel Sobel-X dan Sobel-Y memiliki bobot yang lebih tinggi pada piksel tengah (nilai 2 dan -2), yang membuatnya lebih sensitif terhadap perubahan intensitas. Hal ini menghasilkan deteksi tepi yang lebih tajam dan lebih halus.
2. Kernel Prewitt-X dan Prewitt-Y menggunakan bobot yang lebih sederhana dengan nilai yang sama di semua baris atau kolom, sehingga

menghasilkan deteksi tepi yang lebih kasar dibandingkan Sobel.

4.4 Hasil Analisis

Berdasarkan hasil eksperimen yang dilakukan, perbandingan metode deteksi tepi menggunakan operator Sobel dan Prewitt menunjukkan beberapa perbedaan signifikan dalam hal karakteristik kernel, sensitivitas terhadap noise, hasil visual, dan penerapannya.

1. Koefisien Kernel

Operator Sobel memiliki bobot 2 pada piksel tengah dalam kernel, yang berfungsi untuk memperkuat perubahan intensitas di sekitar titik pusat. Sebaliknya, operator Prewitt menggunakan bobot yang seragam dengan nilai 1, sehingga distribusi bobotnya lebih merata. Hal ini membuat Sobel lebih fokus pada detail lokal dibandingkan Prewitt.

2. Sensitivitas terhadap Noise

Karena pembobotan yang lebih besar pada piksel tengah, operator Sobel lebih tahan terhadap noise dibandingkan Prewitt. Prewitt, dengan bobot yang lebih seimbang di seluruh kernel, lebih rentan terhadap gangguan noise, terutama pada gambar dengan banyak detail kecil atau kontras rendah.

3. Sudut Deteksi Tepi

Operator Sobel lebih unggul dalam mendeteksi tepi dengan sudut diagonal karena perbedaan bobot antara sumbu horizontal dan vertikal. Sementara itu, operator Prewitt lebih efektif dalam mendeteksi tepi yang dominan secara horizontal dan vertikal karena distribusi bobot yang lebih seragam.

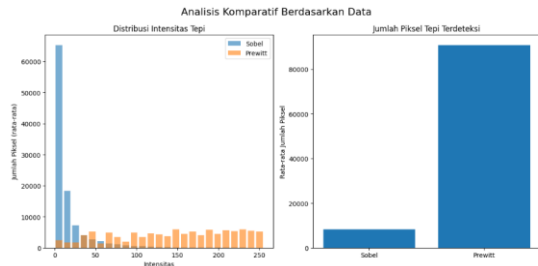
4. Hasil Visual

Dari hasil eksperimen yang divisualisasikan pada gambar, operator Sobel menghasilkan tepi yang lebih tajam dan lebih tebal dibandingkan dengan Prewitt. Di sisi lain, Prewitt memberikan hasil yang lebih halus, tetapi dalam beberapa kasus, detail halus mungkin kurang tajam dibandingkan Sobel.

5. Aplikasi Umum

Operator Sobel lebih sering digunakan dalam aplikasi yang memerlukan deteksi tepi dengan detail yang lebih

kompleks, seperti dalam pengolahan citra medis dan visi komputer. Operator Prewitt lebih cocok digunakan dalam gambar dengan tepi yang memiliki kontras tinggi dan kurang kompleks, seperti pengolahan citra industri atau sistem deteksi sederhana.



Gambar 5. Analisis Komparatif Berdasarkan Data

Hasil analisis menunjukkan bahwa operator Sobel lebih selektif dalam mendeteksi tepi dengan intensitas tinggi, menghasilkan distribusi intensitas yang lebih terpusat di nilai rendah. Sebaliknya, operator Prewitt lebih menyebarkan deteksi tepinya di berbagai tingkat intensitas, menunjukkan sensitivitas yang lebih tinggi terhadap perubahan kecil dalam gambar. Hal ini terlihat pada distribusi intensitas piksel tepi, di mana Sobel memiliki puncak tinggi di intensitas rendah, sedangkan Prewitt lebih merata di seluruh spektrum intensitas.

Selain itu, jumlah piksel tepi yang terdeteksi oleh Prewitt jauh lebih banyak dibandingkan Sobel, yang berarti Prewitt lebih sensitif dalam menangkap perubahan kontras. Namun, sensitivitas tinggi ini juga dapat menyebabkan deteksi yang lebih rentan terhadap noise. Sementara itu, Sobel cenderung memberikan hasil yang lebih tajam dan lebih sedikit noise karena bobot yang lebih besar pada piksel pusat. Oleh karena itu, pemilihan operator bergantung pada kebutuhan spesifik aplikasi, apakah lebih mengutamakan keakuratan tepi utama (Sobel) atau sensitivitas deteksi yang lebih luas (Prewitt).

5. KESIMPULAN

- a. Hasil analisis menunjukkan bahwa operator Sobel lebih selektif dalam mendeteksi tepi dengan intensitas tinggi, menghasilkan garis tepi yang lebih tajam dengan jumlah piksel tepi yang lebih sedikit. Sebaliknya, operator Prewitt lebih sensitif

terhadap perubahan kontras, sehingga mendeteksi lebih banyak tepi tetapi juga lebih rentan terhadap noise. Dari segi distribusi intensitas, Sobel cenderung menghasilkan puncak yang lebih tajam, sedangkan Prewitt menunjukkan distribusi yang lebih merata di berbagai tingkat intensitas.

- b. Dalam hal kelebihan dan kekurangan, Sobel lebih unggul dalam mempertajam tepi utama dan mengurangi efek noise, namun kurang sensitif terhadap perubahan kecil dalam gambar. Sementara itu, Prewitt lebih responsif terhadap variasi intensitas gambar, tetapi cenderung menangkap lebih banyak noise dibandingkan Sobel. Untuk pengembangan lebih lanjut, kombinasi metode Sobel dan Prewitt dapat dieksplorasi guna memperoleh hasil deteksi yang lebih optimal. Selain itu, penerapan teknik pemrosesan citra yang lebih lanjut, seperti filter adaptif atau pendekatan berbasis machine learning, dapat meningkatkan akurasi dan efektivitas deteksi tepi sesuai dengan karakteristik gambar yang diolah.

DAFTAR PUSTAKA

- [1] R. G. Abdullah and M. Kurniawan, "Sistem Deteksi Penyakit Pada Otak Dengan Pendekatan Klasifikasi CNN Dan Preprocessing Image Generator," *JITET (Jurnal Inform. dan Tek. Elektro Ter.)*, vol. 12, no. 3, 2024.
- [2] P. Laia, "Penerapan Metode Prewitt, Canny Dan Sobel Pada Proses Deteksi Tepi Citra," *J. Media Inform. Budidarma*, vol. 2, no. 1, pp. 45–50, 2018, doi: 10.30865/mib.v2i1.996.
- [3] I. M. Sitanggang and P. Marpaung, "Analisis Dan Perbandingan Metode Sobel Dan Canny Pada Deteksi Tepi Citra Daun Sirih Merah," *JIKOMSI [Jurnal Ilmu Komput. dan Sist. Informasi]*, vol. 3, no. 3, 2021, doi: 10.22303/csrid.13.1.2021.13-23.
- [4] E. Rahmawati and G. D. S. Rahayu, "Implementasi Model Discovery Learning Berbasis Media Gambar Dalam Meningkatkan Kecerdasan Ekologis Siswa Sekolah Dasar," *J. Elem. Educ.*, vol. 04, no. 02, p. 242, 2021.

- [5] A. O. Pratiwi and M. R. Qisthiano, "Implementasi Teknologi AI Untuk Sistem Identifikasi Spesies Hewan Berbasis Website Dengan Pendekatan Machine Learning," *J. Rekayasa Sist. Inf. dan Teknol.*, vol. 2, no. 2, pp. 830–837, 2024.
- [6] Sukatmi, "Perbandingan Deteksi Tepi Citra Digital dengan Metode Prewitt , Sobel dan Canny," *KOPERTIP J. Ilm. Manaj. Inform. dan Komput.*, vol. 01, no. 01, pp. 1–4, 2017.
- [7] R. Priambodo and T. M. Kadarina, "Pelacakan Lokasi Pasien berbasis Internet of Things untuk Sistem Pendukung Layanan Kesehatan Ibu dan Anak," *INOVTEK Polbeng - Seri Inform.*, vol. 5, no. 2, 2020, doi: 10.35314/isi.v5i2.1509.
- [8] M. R. Qisthiano, I. Ruswita, and P. Armilia, "Implementasi Metode SVM dalam Analisis Sentimen Mengenai Vaksin dengan Menggunakan Python 3," vol. 13, no. 1, pp. 1–7, 2023.
- [9] L. Riani, A. Zahra, F. Kurniawan, I. L. Valenza, and F. Z. Darmansah, "SiCitra : Aplikasi Berbasis Web untuk Pemrosesan Citra Digital Menggunakan OpenCV," *J. Inform. UPGRIS*, vol. 10, no. 2, 2024.
- [10] T. C. A. S. Zulkhaidi, E. Maria, and Y. Yulianto, "Pengenalan Pola Bentuk Wajah dengan OpenCV," *J. Rekayasa Teknol. Inf.*, vol. 3, no. 2, 2020.
- [11] A. S. Anas and A. A. Rizal, "Deteksi Tepi dalam Pengolahan Citra Digital," *Semin. Nas. TIK dan Ilmu Sos.*, pp. 1–6, 2017.
- [12] A. Wijaya and H. Franata, "Peningkatan Hasil Segmentasi Deteksi Tepi Menggunakan Morphology Pada Pengolahan Citra," *Jukomika - (Jurnal Ilmu Komput. Dan Inform.*, vol. 3, pp. 2655–755, 2020, [Online]. Available: <https://jurnal.ikhafi.or.id/index.php/jukomika/>
- [13] N. B. Tsani and H. Harliana, "Implementasi Deteksi Tepi Canny Dengan Transformasi Powerlaw Dalam Mendeteksi Stadium Kanker Serviks," *J. Ilm. Intech Inf. Technol. J. UMUS*, vol. 1, no. 01, pp. 22–33, 2019, doi: 10.46772/intech.v1i01.35.