



Analisis Kerusakan pada Beton dengan Citra Digital Menggunakan Metode *Edge Canny Detection*

M. Naufal Syahputra^{1*}, Achmad Fauzi², Melda Pita Uli Sitompul³

¹⁻³STMIK Kaputama, Indonesia

Email: smnaufal833@gmail.com^{1*}, ahmad.azy@nusamandiri.ac.id, meldasitompul19@gmail.com

Jl. Veteran No.4A, Tangsi, Kec. Binjai Kota, Kota Binjai, Sumatera Utara, Indonesia 20714

*Penulis korespondensi

Abstract. *This study aims to design and implement a damage analysis system for concrete surfaces by utilizing digital image processing based on the Canny edge detection method. The developed system allows users to upload images of concrete surfaces, which are then processed through several stages: conversion to grayscale, transformation to binary images, and crack edge detection using the Canny operator. This process aims to automatically detect crack patterns on the concrete surface. The detection results, represented as edge lines, are used to calculate the percentage of the damaged area. Based on this percentage value, the system automatically classifies the damage level into light, moderate, or severe categories. System testing shows that the Canny method can accurately identify crack patterns, with sufficient detection levels to be used in monitoring the condition of concrete surfaces. The analysis results are then presented in both visual and numerical forms, providing valuable information for assessing the structural condition of concrete. Thus, this system can serve as an efficient and effective tool for early detection of structural damage in concrete infrastructure, ultimately supporting better maintenance and repair efforts.*

Keywords: *Damage Detection; Dempster-Shafer; Expert Systems; Joint Health; Osteoarthritis.*

Abstrak. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem analisis kerusakan pada permukaan beton dengan memanfaatkan pengolahan citra digital berbasis metode deteksi tepi Canny. Sistem yang dibangun memungkinkan pengguna untuk mengunggah citra permukaan beton, kemudian memrosesnya melalui beberapa tahap, yaitu konversi ke grayscale, transformasi ke citra biner, dan pendeteksian tepi retakan menggunakan operator Canny. Proses ini bertujuan untuk mendeteksi pola retakan yang terdapat pada permukaan beton secara otomatis. Hasil deteksi berupa garis tepi digunakan untuk menghitung persentase area yang mengalami kerusakan. Berdasarkan nilai persentase tersebut, sistem akan secara otomatis mengklasifikasikan tingkat kerusakan menjadi kategori ringan, sedang, atau berat. Pengujian sistem menunjukkan bahwa metode Canny mampu mengidentifikasi pola retakan secara akurat, dengan tingkat deteksi yang memadai untuk digunakan dalam pemantauan kondisi permukaan beton. Hasil analisis ini kemudian disajikan dalam bentuk visual dan numerik yang memberikan informasi yang bermanfaat untuk penilaian kondisi struktural beton. Dengan demikian, sistem ini dapat menjadi alat bantu yang efisien dan efektif dalam pemantauan dini kerusakan struktural pada infrastruktur beton, yang pada akhirnya dapat mendukung upaya pemeliharaan dan perawatan yang lebih baik.

Kata kunci: Dempster-Shafer; Deteksi Kerusakan; Kesehatan Sendi; Osteoarthritis; Sistem Pakar.

1. LATAR BELAKANG

Beton adalah suatu elemen dalam konstruksi yang merupakan struktur sederhana yang dibentuk oleh campuran semen, air, agregat halus, agregat kasar yang berupa batu pecah atau kerikil, udara serta bahan campuran lainnya. Beton merupakan material yang sering digunakan pada struktur bangunan. Salah satu alasannya adalah kemudahan dalam pengerjaan yang

didukung dengan kekuatan yang tinggi. Dalam hal kekuatan, beton kuat menahan tekan tetapi lemah dalam menahan gaya tarik. Oleh sebab itu, dalam banyak pekerjaan sipil, beton dikombinasikan dengan baja yang kuat menahan gaya tarik atau yang sering disebut beton bertulang.

Keretakan pada beton bisa disebabkan oleh berbagai faktor seperti beban berlebih, perubahan suhu, dan faktor lingkungan lainnya. Kedalaman keretakan adalah parameter kunci yang perlu diukur untuk menilai tingkat kerusakan beton dan mengambil langkah-langkah perbaikan yang diperlukan. Oleh karena itu, analisa yang akurat terhadap kedalaman keretakan sangat penting dalam pemeliharaan dan perbaikan struktur beton.

Salah satu teknik pengolahan citra yang sangat populer dan efektif adalah metode prediksi tepi Canny. Metode ini dikembangkan oleh John F. Canny pada tahun 1986 dan telah menjadi standar dalam deteksi tepi karena kemampuannya dalam menghasilkan tepi yang jelas dengan tingkat kebisingan yang rendah (Mulyana & Pratama, 2023). Metode ini bekerja dengan mencari area dalam gambar di mana terjadi perubahan intensitas yang tajam, yang biasanya menunjukkan keberadaan tepi objek. Dengan menggunakan metode ini, pola dan bentuk daun Bidara dapat dikenali dengan lebih baik.

Penelitian relevan sebelumnya telah menunjukkan bahwa metode pengolahan citra digital dapat sangat efektif dalam identifikasi tanaman. Misalnya, penelitian oleh Endru Suphato dan Henny Wiyanto (2022) dengan judul “Penilaian Kerusakan Beton Melalui Pemrosesan Gambar Dengan Canny Edge Detection Method”. Pemrosesan gambar dilakukan dengan merancang aplikasi dengan bantuan MATLAB menggunakan Graphical User Interface (GUI). Langkah selanjutnya adalah membuat fungsi untuk membaca dan memproses gambar kerusakan, menampilkan jenis kerusakan, menentukan tingkat kondisi setiap jenis kerusakan, menghitung tingkat kondisi per-elemen struktur. Penggunaan aplikasi penilaian dilakukan dengan menginput gambar kerusakan berupa foto hasil pengamatan lapangan dan menentukan jenis kerusakan. Aplikasi akan secara otomatis menghasilkan tingkat kondisi kerusakan beton.

2. KAJIAN TEORITIS

Berdasarkan uraian latar belakang masalah di atas maka identifikasi masalah dalam penelitian ini sebagai berikut : (1) Bagaimana Bagaimana metode tepi canny dapat menganalisa citra kerusakan beton? (2) Apa yang menjadi parameter daun bidara untuk mengidentifikasi citra kerusakan beton? (3) Bagaimana akurasi dan efesiensi hasil dari pengenalan pola dan citra kerusakan beton dengan metode tepi canny yang digunakan?

Untuk mencegah mengembangkannya permasalahan maka penulis membatasi masalah untuk lebih terperinci dan jelas sehingga pemecahan terarah dan berhasil. Sebagai berikut: (1) Penelitian ini hanya fokus pada pengenalan pola kerusakan beton. (2) Metode yang digunakan dalam penelitian ini adalah deteksi tepi canny. (3) Software Matlab yang digunakan adalah Matlab R2014b.

3. METODE PENELITIAN

Pengenalan Pola

Pengenalan pola adalah bidang dalam ilmu komputer dan kecerdasan buatan yang berkaitan dengan pengidentifikasian dan klasifikasi objek atau pola berdasarkan fitur-fitur yang diekstraksi dari data (Al Fatta, 2009). Pada dasarnya, pengenalan pola bertujuan untuk membedakan antara berbagai kelas objek dengan menggunakan informasi yang terkandung dalam data tersebut. Proses pengenalan pola menurut (Andono & Sutojo, 2018) dimulai dengan pengumpulan data, yang dapat berupa citra, suara, teks, atau data sensor lainnya. Data ini kemudian diproses untuk menghilangkan noise dan mengoptimalkan kualitasnya. Langkah selanjutnya adalah ekstraksi fitur, di mana karakteristik penting dari data diidentifikasi. Fitur ini bisa berupa informasi geometris, tekstur, warna, atau pola spesifik yang relevan dengan objek yang sedang dianalisis. Setelah fitur diekstraksi, langkah berikutnya adalah klasifikasi. Dalam tahap ini, model algoritma digunakan untuk menentukan kelas atau kategori objek berdasarkan fitur yang telah diperoleh. Berbagai algoritma dapat diterapkan, termasuk K-Nearest Neighbors (KNN), Support Vector Machines (SVM), dan model berbasis pembelajaran mendalam, seperti Convolutional Neural Networks (CNN). Proses klasifikasi melibatkan pelatihan model menggunakan data yang telah dilabeli, sehingga model dapat belajar mengenali pola yang ada.

Pengenalan pola adalah proses mengelompokkan data numerik dan simbolik termasuk citra secara otomatis oleh komputer, tujuan dari pengelompokan adalah untuk mengenali suatu objek di dalam citra (Gunawan, 2024). Manusia bisa mengenali objek-objek disekitarnya karena otak manusia belajar mengklasifikasi objek-objek di alam sehingga mampu membedakan suatu objek dengan objek lainnya. Kemampuan sistem visual manusia inilah yang akan ditiru oleh mesin. Komputer akan menerima masukan berupa citra objek yang akan diidentifikasi kemudian memproses citra tersebut dan memberikan keluaran berupa informasi atau deskripsi objek di dalam citra. Gambar 2.1 menunjukkan diagram kotak sederhana dari proses pengenalan pola.



Beton

Beton adalah suatu material yang terdiri dari campuran semen, air, agregat (kasar dan halus) dan bahan tambahan bila diperlukan. Beton yang banyak dipakai pada saat ini yaitu beton normal. Beton normal ialah beton yang mempunyai berat isi 2200–2500 kg/m³ dengan menggunakan agregat alam yang dipecah atau tanpa dipecah. Beton normal dengan kualitas yang baik yaitu beton yang mampu menahan kuat desak/hancur yang diberi beban berupa tekanan dengan dipengaruhi oleh bahan-bahan pembentuk, kemudahan pengerjaan (*workability*), faktor air semen (FAS) dan zat tambahan (*admixture*) bila diperlukan (Alam, dkk). Beton merupakan bahan dari campuran antara Portland cement, agregat halus (pasir), agregat kasar (kerikil), air dengan tambahan adanya rongga-rongga udara. Campuran bahan-bahan pembentuk beton harus ditetapkan sedemikian rupa, sehingga menghasilkan beton basah yang mudah dikerjakan, memenuhi kekuatan tekan rencana setelah mengeras dan cukup ekonomis (Sutikno, 2023).

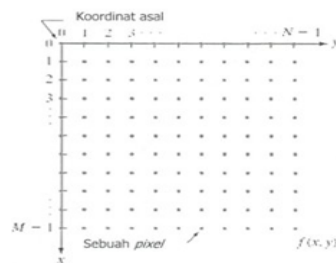
Pengenalan Pola

Pengenalan pola adalah bidang dalam ilmu komputer dan kecerdasan buatan yang berkaitan dengan pengidentifikasian dan klasifikasi objek atau pola berdasarkan fitur-fitur yang diekstraksi dari data (Al Fatta, 2009). Pada dasarnya, pengenalan pola bertujuan untuk membedakan antara berbagai kelas objek dengan menggunakan informasi yang terkandung dalam data tersebut. Proses pengenalan pola menurut (Andono & Sutojo, 2018) dimulai dengan pengumpulan data, yang dapat berupa citra, suara, teks, atau data sensor lainnya. Data ini kemudian diproses untuk menghilangkan noise dan mengoptimalkan kualitasnya. Langkah selanjutnya adalah ekstraksi fitur, di mana karakteristik penting dari data diidentifikasi. Fitur ini bisa berupa informasi geometris, tekstur, warna, atau pola spesifik yang relevan dengan objek yang sedang dianalisis.

Setelah fitur diekstraksi, langkah berikutnya adalah klasifikasi. Dalam tahap ini, model algoritma digunakan untuk menentukan kelas atau kategori objek berdasarkan fitur yang telah diperoleh. Berbagai algoritma dapat diterapkan, termasuk *K-Nearest Neighbors* (KNN), *Support Vector Machines* (SVM), dan model berbasis pembelajaran mendalam, seperti *Convolutional Neural Networks* (CNN). Proses klasifikasi melibatkan pelatihan model menggunakan data yang telah dilabeli, sehingga model dapat belajar mengenali pola yang ada. Evaluasi model merupakan langkah krusial dalam pengenalan pola. Menggunakan metrik seperti akurasi, presisi, recall, dan F1-score, kinerja model diuji dengan menggunakan dataset yang tidak digunakan selama pelatihan. Hal ini membantu dalam menilai seberapa baik model dapat menggeneralisasi pada data baru dan mengidentifikasi potensi masalah seperti *overfitting* atau *underfitting*.

Citra Digital

Menurut (Dijaya & Setiawan, 2023) pengolahan citra digital adalah disiplin ilmu yang mencakup berbagai teknik dan metode untuk memanipulasi gambar dalam format digital. Proses ini tidak hanya melibatkan manipulasi sederhana seperti pemotongan atau perubahan ukuran gambar, tetapi juga teknik-teknik yang lebih kompleks yang digunakan untuk meningkatkan kualitas gambar, mengekstrak informasi penting, atau mempersiapkan gambar untuk pemrosesan lebih lanjut. Pengolahan citra digital mengintegrasikan prinsip-prinsip matematika, komputasi, dan teknik untuk mengubah gambar mentah menjadi bentuk yang lebih berguna dan informatif. Citra atau Gambar (Image) adalah sebuah informasi yang memiliki bentuk visual. Suatu citra yang diperoleh dari hasil kamera dan dapat diolah menggunakan komputer atau laptop merupakan citra digital, dapat juga diartikan bahwa citra adalah kumpulan dari piksel-piksel yang disusun dalam dua dimensi. Piksel merupakan elemen terkecil yang menyusun citra pada suatu titik tertentu mengandung nilai yang mewakili kecerahan dari sebuah warna, Umumnya Citra digital berbentuk persegi panjang yang memiliki lebar dan tingi tertentu. Ukuran ini biasanya dinyatakan dalam banyaknya piksel sehingga ukuran citra selalu bernilai bulat (Agusti & Nababan, 2022). Dalam komputer, setiap piksel diwakili oleh dua buah bilangan bulat (integer) untuk menunjukkan lokasinya dalam bidang citra dan sebuah nilai dalam bilangan bulat untuk menunjukkan cahaya atau terang-gelapnya piksel tersebut. Untuk menunjukkan lokasi pada suatu piksel, koordinat (0,0) digunakan sebagai posisi kiri atas dalam bidang citra, dan koordinat (m-1, n-1) digunakan sebagai posisi kanan bawah dalam citra berukuran (m x n) piksel. dapat dilihat pada gambar 2.1.



Gambar 1. Koordinat Citra Digital.

Algoritma Canny Edge Detection

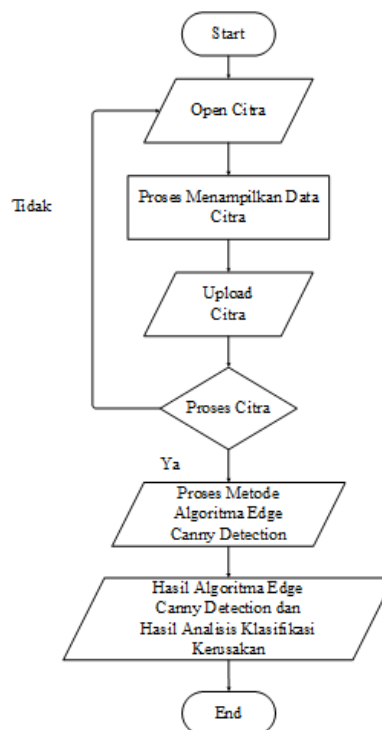
Algoritma Canny Edge Detection adalah teknik dalam pengolahan citra untuk mendeteksi tepi (edge) pada gambar. Algoritma ini mengabungkan beberapa langkah seperti penghalusan (smoothing), perhitungan gradien, non-maximum suppression, dan histerisis untuk menghasilkan tepi yang akurat dan tajam pada citra. Hal ini membantu mengidentifikasi batas objek dan memisahkan dari latar belakang dalam citra. Operator yang menggunakan 2

buah karnel 16×16 (K_x dan K_y) dan magnitude gradient (G) dengan persamaan 1, 2, 3. Adapun langkah-langkah perhitungan Algoritma Canny Edge Detection dibawah ini: (1) Saat gambar gambar dihaluskan, turunan I_x dan I_y wrt x dan y dihitung. Ini dapat diimplementasikan dengan menggabungkan I dengan karnel Canny K_x dan K_y , dengan persamaan (2.1) dan (2.2) : (2) Kemudian, besaran G dan kemiringan Θ dari gradient dihitung dengan persamaan (2.3) : (3) Menentukan nilai Thresholding menggunakan persamaan (2.4)

4. HASIL DAN PEMBAHASAN

Analisa Sistem

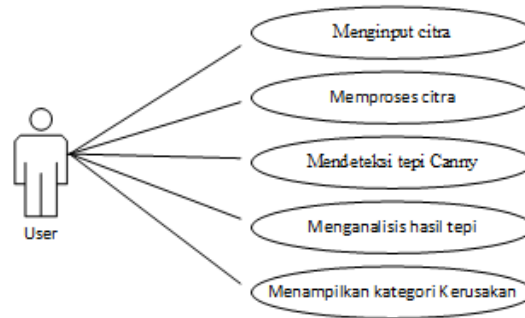
Analisis sistem adalah proses mempelajari dan memahami suatu sistem untuk mengidentifikasi masalah, kebutuhan, dan peluang perbaikan. Tujuannya adalah untuk mengumpulkan informasi yang diperlukan untuk merancang, mengembangkan, atau meningkatkan sistem yang ada. (1) Flowchart, *Flowchart* yang menggambarkan langkah-langkah yang dilakukan oleh sistem dalam penilaian kerusakan beton pada bangunan menggunakan deteksi tepi Canny dalam proses prancangannya. Berikut adalah *flowchart* dalam pembuatan penilaian kerusakan beton dengan menggunakan metode deteksi tepi *Canny*.



Gambar 2. *Flowchart* Klasifikasi Kerusakan Beton.

(2) Use Case Diagram, Rancangan *use case diagram* yang berfungsi untuk interaksi antara pemain atau aktor atau pengguna sistem yang akan dihubungkan sebagai pengguna sistem

lainnya. Ada beberapa perancangan use case diagram dalam sistem pengenalan pola citra daun bidara adalah sebagai berikut:



Gambar 3. Use Case Diagram.

Penerapan Algoritma Adge

Berdasarkan pengamatan yang telah dilakukan, deteksi garis tepi pada kerusakan pada Beton bisa dilihat pada keretakannya. Melihat dari keretakan beton jalan di analisis lebih lanjut dan dijelaskan pada gambar 4:



Gambar 4. Citra Kerusakan Beton.

Gambar diatas menunjukkan bahwa untuk menganalisis memerlukan letak pada keretakan dan mengukur retakan dengan menentukan titik koordinat (sumbu X dan Y) pada citra digital. Dan dapat diproses yang di deteksi tepi Canny dalam pemrograman pengolahan citra seperti MATLAB.

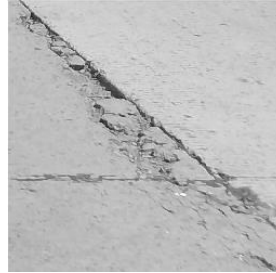
Perhitungan Algoritma Canny

Dalam analisis citra digital terdiri dari rangkaian tahapan yang saling mendukung untung menghasilkan tepi yang akurat. Proses perhitungan dimulai dari Smoothing, Finding Gradient dan Thresholding. Uraian tiap tahap akan dijelaskan pada sub-bab berikut:

Smoothing

Tahap smoothing merupakan proses awal dalam metode deteksi tepi Canny yang bertujuan untuk mengurangi gangguan (noise) pada citra digital. Proses ini dilakukan dengan menggunakan teknik konvolusi, di mana nilai intensitas setiap piksel diperbarui berdasarkan rata-rata bobot dari piksel-piksel di sekitarnya.

Sebelum proses ini berjalan, citra terlebih dahulu dikonversi ke dalam bentuk grayscale agar setiap piksel hanya memiliki satu nilai intensitas. Pengubahan ini penting untuk mempermudah pengolahan citra.. Ilustrasi hasil konversi gambar beton ke grayscale yang memperlihatkan bentuk awal permukaan beton dalam skala keabuan sebagai dasar deteksi tepi. Dapat dilihat pada gambar 5.



Gambar 5. Citra Beton Grayscale

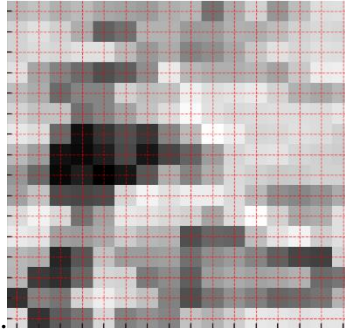
Nilai-nilai tersebut mewakili intensitas keabuan dari masing-masing piksel. Jumlah elemen bilangan pada matriks tergantung pada ukuran dan resolusi gambar yang digunakan. Sebagai contoh, dalam proses perhitungan awal, citra diambil sebagian dari baris dan kolom tertentu. Dalam perhitungan ini berukuran 18×18 piksel untuk dianalisis lebih lanjut. Matrik ini menjadi dasar dalam menentukan pola retakan berdasarkan perubahan nilai intensitas antar piksel. Hasil konvolusi selanjutnya ditampilkan dalam bentuk bilangan matriks, seperti pada gambar 6. berikut:

189	190	192	191	190	190	193	196	194	193	192	193	193	192	188	185	189	189
189	191	192	191	189	188	189	191	194	193	192	192	193	192	190	188	186	191
189	190	191	190	189	188	189	190	192	191	190	190	192	193	192	191	183	192
187	187	188	188	188	189	191	192	188	187	187	188	190	191	192	192	190	183
183	182	181	181	182	184	186	187	184	184	185	187	189	191	191	192	194	181
182	180	177	176	177	179	179	179	181	183	186	189	191	192	191	191	192	187
184	182	180	180	182	183	182	180	182	185	190	195	196	195	193	192	194	190
187	186	186	188	191	192	190	188	183	188	194	199	200	198	195	193	193	195
189	190	188	185	192	200	196	186	171	173	179	187	191	191	190	191	196	183
186	188	188	186	188	189	180	169	152	130	150	154	161	171	181	189	182	175
185	187	188	190	191	188	180	173	162	158	154	155	160	169	181	189	185	182
187	186	188	191	192	191	188	188	187	187	186	187	187	187	187	187	187	188
190	188	187	188	188	187	187	189	191	194	198	199	198	195	190	187	184	188
192	191	190	191	190	188	186	187	190	192	193	193	193	194	193	192	188	191
191	192	193	193	194	194	192	190	192	194	193	191	190	192	192	191	187	187
189	191	192	191	192	195	194	190	188	191	193	192	190	189	185	181	184	185
190	190	191	192	192	192	192	192	191	193	194	193	191	188	186	185	185	186
189	189	190	191	191	191	191	191	192	193	193	192	190	188	187	186	190	190

Gambar 6. Bilangan Matriks.

Finding Gradient

Pada tahap ini, citra terlebih dahulu dibagi menjadi blok berukuran 16×16 piksel untuk mempermudah analisis. Setiap blok kemudian dikonversi menjadi bentuk numerik berdasarkan citra intensitas keabuan (grayscale), sehingga memungkinkan perhitungan ini selisih antar piksel untuk mendeteksi perubahan kontras. Proses ini membantu sistem mengidentifikasi area yang mengalami perbedaan nilai terang-gelap mencolok, yang sering kali menunjukkan keberadaan tepi atau retakan.



Gambar 7. Sampel Citra 16 x 16 Pixel.

189	190	192	191	190	190	193	196	194	193	192	193	193	192	188	185
189	191	192	191	189	188	189	191	194	193	192	192	193	192	190	188
189	190	191	190	189	188	189	190	192	191	190	190	192	193	192	191
187	187	188	188	188	189	191	192	188	187	187	188	190	191	192	192
183	182	181	181	182	184	186	187	184	184	185	187	189	191	191	192
182	180	177	176	177	179	179	179	181	183	186	189	191	192	191	191
184	182	180	180	182	183	182	180	182	185	190	195	196	195	193	192
187	186	186	188	191	192	190	188	183	188	194	199	200	198	195	193
189	190	188	185	192	200	196	186	171	173	179	187	191	191	190	191
186	188	188	186	188	189	180	169	152	150	150	154	161	171	181	189
185	187	188	190	191	188	180	173	162	158	154	155	160	169	181	189
187	186	188	191	192	191	188	188	187	187	186	187	187	187	187	187
190	188	187	188	188	187	187	189	191	194	198	199	198	195	190	187
192	191	190	191	190	188	186	187	190	192	193	193	193	194	193	192
191	192	193	193	194	194	192	190	192	194	193	191	190	192	192	191
189	191	192	191	192	195	194	190	188	191	193	192	190	189	185	181

Gambar 8 Nilai Pixel 16x16

Menentukan operator Canny harus menggunakan matriks dari Canny Edge Detection K_x dan K_y untuk melakukan deteksi tepi. Dengan rumus sebagai berikut:

$$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad K_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Nilai-nilai piksel akan diproses dengan metode Canny Edge Detection untuk mendeteksi tepi pada citra dan nilai tersebut diproses sesuai dengan ketentuan kernel Canny Edge Detection horizontal (K_x) dan kernel Canny Edge Detection vertikal (K_y).

Menentukan nilai gradien dari citra grayscale yang telah dioperasikan dengan nilai matriks menjadi pixel citra. Perhitungan Canny Edge Detection dapat dilihat pada bawah ini:

189	190	192
189	191	192
189	190	191

Gambar 8. Pixel Citra Perhitungan Pertama.

$$K_x = (189) (-1) + (189) (-2) + (189) (-1) + (190) (0) + (191) (0) + (190) (0) + (192) (1) + (192) (2) + (191) (1) = 11$$

$$K_y = (189) (1) + (190) (2) + (192) (1) + (189) (0) + (191) (0) + (192) (0) + (189) (-1) + (190) (-2) + (191) (-1) = 1$$

$$K [f(x,y)] = \sqrt{-11^2 + 1^2} = 11,045$$

Nilai pixel yang diperoleh dari perhitungan pixel yang telah dioperasikan. Terletak pada baris 1 dan kolom 1 adalah 11,045.

Perhitungan ke-2 dapat di hitung pada gambar 9:

190	192	191
191	192	191
190	191	190

Gambar 9. Pixel Citra Perhitungan ke-2.

$$K_x = (190) (-1) + (191) (-2) + (190) (-1) + (191) (0) + (192) (0) + (191) (0) + (191) (1) + (191) (2) + (190) (1) = 1$$

$$K_y = (190) (1) + (192) (2) + (191) (1) + (191) (0) + (192) (0) + (191) (0) + (190) (-1) + (191) (-2) + (190) (-1) = 1$$

$$G [f(x,y)] = \sqrt{1^2 + 1^2} = 1,414$$

Nilai pixel yang diperoleh dari perhitungan pixel yang telah dioperasikan. Terletak pada baris 1 dan kolom 2 adalah 1,414.

192	191	190
192	191	189
191	190	189

Gambar 10. Pixel Citra Perhitungan Ke-3.

$$K_x = (192) (-1) + (192) (-2) + (191) (-1) + (191) (0) + (191) (0) + (190) (0) + (190) (1) + (189) (2) + (189) (1) = -10$$

$$K_y = (135) (1) + (110) (2) + (123) (1) + (108) (0) + (121) (0) + (123) (0) + (131) (-1) + (106) (-2) + (119) (-1) = 4$$

$$K [f(x,y)] = \sqrt{-10^2 + 4^2} = 10,770$$

Nilai pixel yang diperoleh dari perhitungan pixel yang telah dioperasikan. Terletak pada baris 1 dan kolom 3 adalah 10,770.

Perhitungan ke- 4 dapat di hitung pada gambar 11:

191	190	190
191	189	188
190	189	188

Gambar 11. Pixel Citra Perhitungan Ke-4.

$$K_x = (191) (-1) + (191) (-2) + (190) (-1) + (190) (0) + (189) (0) + (189) (0) + (190) (1) + (188) (2) + (188) (1) = -9$$

$$K_y = (191) (1) + (190) (2) + (190) (1) + (191) (0) + (189) (0) + (189) (0) + (190) (-1) + (188) (-2) + (188) (-1) = 5$$

$$K [f(x,y)] = \sqrt{-9^2 + 5^2} = 10,295$$

Nilai pixel yang diperoleh dari perhitungan pixel yang telah dioperasikan. Terletak pada baris 1 dan kolom 4 adalah 10,295

Perhitungan ke-5 dapat di hitung pada gambar 12:

190	190	193
189	188	189
189	188	189

Gambar 12. Pixel Citra Perhitungan Ke-5.

$$K_x = (190) (-1) + (189) (-2) + (189) (-1) + (190) (0) + (188) (0) + (188) (0) + (193) (1) + (189) (2) + (189) (1) = 3$$

$$K_y = (190) (1) + (190) (2) + (193) (1) + (189) (0) + (188) (0) + (188) (0) + (189) (-1) + (188) (-2) + (188) (-1) = 9$$

$$K [f(x,y)] = \sqrt{3^2 + 9^2} = 9,486$$

Nilai pixel yang diperoleh dari perhitungan pixel yang telah dioperasikan. Terletak pada baris 1 dan kolom 5 adalah 9,486

Perhitungan ke-6 dapat di hitung pada gambar 13:

190	193	196
188	189	191
188	189	190

Gambar 13. Pixel Citra Perhitungan Ke-6.

$$K_x = (190) (-1) + (188) (-2) + (188) (-1) + (193) (0) + (189) (0) + (189) (0) + (196) (1) + (191) (2) + (190) (1) = 14$$

$$K_y = (190) (1) + (193) (2) + (196) (1) + (188) (0) + (189) (0) + (191) (0) + (188) (-1) + (189) (-2) + (190) (-1) = 16$$

$$K [f(x,y)] = \sqrt{14^2 + 16^2} = 21,260$$

Nilai pixel yang diperoleh dari perhitungan pixel yang telah dioperasikan. Terletak pada baris 1 dan kolom 6 adalah 21,260.

Perhitungan ke-7 dapat di hitung pada gambar 14:

193	196	194
189	191	194
189	190	192

Gambar 14. Pixel Citra Perhitungan Ke-7.

$$K_x = (193) (-1) + (189) (-2) + (189) (-1) + (196) (0) + (191) (0) + (190) (0) + (194) (1) + (194) (2) + (192) (1) = 4$$

$$K_y = (193) (1) + (196) (2) + (194) (1) + (189) (0) + (191) (0) + (194) (0) + (189) (-1) + (190) (-2) + (194) (-1) = 8$$

$$K [f(x,y)] = \sqrt{4^2 + 8^2} = 22,803$$

Nilai pixel yang diperoleh dari perhitungan pixel yang telah dioperasikan. Terletak pada baris 1 dan kolom 7 adalah 22,803.

Perhitungan ke- 8 dapat di hitung pada gambar 15:

196	194	193
191	194	193
190	192	191

Gambar 15. Pixel Citra Perhitungan Ke-8.

$$K_x = (193) (-1) + (191) (-2) + (190) (-1) + (194) (0) + (194) (0) + (192) (0) + (193) (1) + (193) (2) + (191) (1) = 2$$

$$K_y = (193) (1) + (194) (2) + (193) (1) + (191) (0) + (194)(0) + (193) (0) + (190) (-1) + (193) (-2) + (191) (-1) = 12$$

$$K [f(x,y)] = \sqrt{2 + 12^2} = 12,165$$

Nilai pixel yang diperoleh dari perhitungan pixel yang telah dioperasikan. Terletak pada baris 1 dan kolom 8 adalah 12,165.

Perhitungan ke- 9 dapat di hitung pada gambar 16:

194	193	192
194	193	192
192	191	190

Gambar 16. Pixel Citra Perhitungan Ke-9.

$$K_x = (194) (-1) + (194) (-2) + (192) (-1) + (193) (0) + (193) (0) + (191) (0) + (192) (1) + (192) (2) + (190) (1) = -8$$

$$K_y = (194) (1) + (193) (2) + (192) (1) + (184) (0) + (193) (0) + (192) (0) + (192) (-1) + (191) (-2) + (190) (-1) = 8$$

$$K [f(x,y)] = \sqrt{-3^2 + 9^2} = 11,313$$

Nilai pixel yang diperoleh dari perhitungan pixel yang telah dioperasikan. Terletak pada baris 1 dan kolom 9 adalah 11,313.

Perhitungan ke- 10 dapat di hitung pada gambar 17:

193	192	193
193	192	192
191	190	190

Gambar 17. Pixel Citra Perhitungan Ke-10.

$$K_x = (193) (-1) + (193) (-2) + (191) (-1) + (192) (0) + (192) (0) + (190) (0) + (193) (1) + (192) (2) + (190) (1) = 9$$

$$K_y = (193) (1) + (192) (2) + (193) (1) + (193) (0) + (192) (0) + (192) (0) + (191) (-1) + (192) (-2) + (190) (-1) = 9$$

$$K [f(x,y)] = \sqrt{9^2 + 9^2} = 12,727.$$

Nilai pixel yang diperoleh dari perhitungan pixel yang telah dioperasikan. Terletak pada baris 1 dan kolom 10 adalah 12,727.

Perhitungan ke- 11 dapat di hitung pada gambar 18:

192	193	193
192	192	193
190	190	192

Gambar 18. Pixel Citra Perhitungan Ke-11.

$$K_x = (192) (-1) + (192) (-2) + (190) (-1) + (193) (0) + (192) (0) + (190) (0) + (193) (1) + (193) (2) + (192) (1) = 5$$

$$K_y = (192) (1) + (193) (2) + (193) (1) + (192) (0) + (192) (0) + (193) (0) + (190) (-1) + (190) (-2) + (192) (-1) = 9$$

$$K [f(x,y)] = \sqrt{5^2 + 9^2} = 10,295$$

Nilai pixel yang diperoleh dari perhitungan pixel yang telah dioperasikan. Terletak pada baris 1 dan kolom 11 adalah 10,295.

Perhitungan ke- 12 dapat di hitung pada gambar 19:

193	193	192
192	193	192
190	192	193

Gambar 19. Pixel Citra Perhitungan Ke-12.

$$K_x = (193) (-1) + (192) (-2) + (190) (-1) + (193)(0) + (193) (0) + (192) (0) + (192) (1) + (92) (2) + (193) (1) = 2$$

$$K_y = (193) (1) + (193) (2) + (192) (1) + (192) (0) + (193) (0) + (193) (0) + (190) (-1) + (192) (-2) + (193) (-1) = 4$$

$$K [f(x,y)] = \sqrt{2^2 + 4^2} = 4,472.$$

Nilai pixel yang diperoleh dari perhitungan pixel yang telah dioperasikan. Terletak pada baris 1 dan kolom 12 adalah 89.442

Perhitungan ke- 13 dapat di hitung pada gambar 20:

193	192	188
193	192	190
192	193	192

Gambar 20. Pixel Citra Perhitungan Ke-13.

$$K_x = (193) (-1) + (193) (-2) + (192) (-1) + (192) (0) + (192) (0) + (193) (0) + (188) (1) + (193) (2) + (192) (1) = -11$$

$$K_y = (193) (1) + (192) (2) + (188) (1) + (193) (0) + (192) (0) + (190) (0) + (192) (-1) + (193) (-2) + (193) (-1) = -5$$

$$K [f(x,y)] = \sqrt{-11^2 + (-5^2)} = 12,083.$$

Nilai pixel yang diperoleh dari perhitungan pixel yang telah dioperasikan. Terletak pada baris 1 dan kolom 13 adalah 12,083.

Perhitungan ke- 14 dapat di hitung pada gambar 21:

192	188	185
192	190	188
193	192	191

Gambar 21. Pixel Citra Perhitungan Ke-14.

$$K_x = (192) (-1) + (192) (-2) + (193) (-1) + (188) (0) + (190) (0) + (192) (0) + (185) (1) + (188) (2) + (191) (1) = -17$$

$$K_y = (192) (1) + (188) (2) + (185) (1) + (192) (0) + (190) (0) + (188) (0) + (193) (-1) + (192) (-2) + (191) (-1) = -15$$

$$K [f(x,y)] = \sqrt{-17^2 + (15^2)} = 22,671$$

Nilai pixel yang diperoleh dari perhitungan pixel yang telah dioperasikan. Terletak pada baris 1 dan kolom 14 adalah 22,671.

Thresholding

Menentukan Nilai Thresholding bisa dilihat dibawah ini:

$$g(x,y) = \begin{cases} 1, & \text{jika } f(x,y) > 128 \\ 0, & \text{jika } f(x,y) < 128 \end{cases}$$

Dengan melakukan pengecekan angka jika nilai angka > 128,maka nilai biner yang diberikan adalah 1. Jika nilai angka < 128,maka nilai biner yang diberikan adalah 0. Dengan hasil perhitungan piksel yang telah dijumlahkan dapat diubah ke biner,sebagai berikut:

(11,045), (1,414), (10,770), (10,295), (9,486), (21,260), (22,804), (12,165), (11,313), (12,727), (10,295), (4,472), (12,083), (22,627) Menjadi 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.

Posisi pada letak yang dimaksud baris dan kolom dari tiap hasil perhitungan piksel menjadi nilai grayscale dan nilai biner. Tujuannya untuk menyederhanakan representasi citra. Terlihat pada gambar 22 dan gambar 23:

5. KESIMPULAN DAN SARAN

Kesimpulan

Setelah melakukan penjelasan pada bab sebelumnya. Maka peneliti akan memberikan beberapa kesimpulan hasil program yang dibuat. Berikut ini adalah kesimpulan dari penelitian terkait dengan teknik pengolahan citra beton berdasarkan citra tekstur dengan Algoritma Canny Edge Detection: Dalam merancang dan menganalisa pemrosesan gambar menggunakan

metode Canny Edge Detection dapat efektif digunakan untuk penilaian kerusakan pada beton. Deteksi didalam pengujian berdasarkan klasifikasi tingkat kerusakan beton. Jika terdapat keretakan 5–10%,maka dikategorikan kerusakan ringan. Selanjutnya keretakan 10–25%,maka dikategorikan kerusakan sedang. Jika keretakan lebih dari 25%,maka dikategorikan kerusakan berat. Dari hasil pengujian pertama yang dilakukan terdapat keretakan sebesar 10,56% dikategorikan kerusakan sedang. Untuk pengujian kedua dan ketiga terdapat keretakan mencapai 13,10% dan 14,84% dikategorikan kerusakan sedang. Maka dapat disimpulkan dalam pengujian telah berhasil,tetapi kerusakan ringan dan berat,peneliti belum mendapatkan gambar/objek tersebut. Dikarenakan resolusi kamera masih kurang bagus.

Saran

Kesimpulan diatas menyampaikan saran yang diharapkan untuk kemajuan sistem yang akan datang sebagai berikut: Diperlukan pengembangan lebih lanjut dalam pengolahan citra untuk meningkatkan akurasi deteksi kerusakan, khususnya dalam kondisi gambar yang kurang optimal. Penelitian lebih lanjut juga disarankan untuk menggabungkan metode Canny Edge Detection dengan algoritma lain,seperti teknik pembelajaran mesin untuk meningkatkan hasil deteksi dan memungkinkan deteksi kerusakan yang lebih kompleks.

DAFTAR REFERENSI

- Agusti, D., & Nababan, A. (2022). Penerapan Metode Harmonic Mean Filter Dalam Mereduksi Gaussian Noise Pada Citra Digital. *Jurnal Nasional Komputasi Dan Teknologi Informasi (JNKTI)*, 5(3), 565-571.<https://doi.org/10.32672/jnkti.v5i3.4468>
- Al Fatta, H. (2009). *Rekayasa Sistem Pengenalan Wajah*. Yogyakarta: Penerbit Andi.
- Andono, P., & Sutojo, T. (2018). *Pengolahan citra digital*. Yogyakarta: Penerbit Andi.
- Bhahri, S. (2018). Transformasi Citra Biner Menggunakan Metode Thresholding Dan Otsu Thresholding. *E-JURNAL JUSITI: Jurnal Sistem Informasi dan Teknologi Informasi*, 7(2), 196-203.<https://doi.org/10.36774/jusiti.v7i2.254>
- Chau, S., Banjarnahor, J., Irfansyah, D., & Kumala, S. (2019). Analisis Pendeteksian Pola Wajah Menggunakan Metode Haar-Like Feature. *Journal of Informatics and Telecommunication Engineering*, 2(2), 69-76.<https://doi.org/10.31289/jite.v2i2.2133>
- Dijaya, R., & Setiawan, H. (2023). *Buku Ajar Pengolahan Citra Digital*. Sidoarjo: Umsida Press.<https://doi.org/10.21070/2023/978-623-464-075-5>

- Gunawan, C. (2024). Identifikasi Nilai Mata Uang Logam Menggunakan Metode OTSU THRESHOLDING. *Kohesi: Jurnal Sains dan Teknologi*, 2(11), 1-9.
- Haqi, B. (2019). Aplikasi SPK Pemilihan Dosen Terbaik Metode Simple Additive Weighting (SAW) Dengan Java. Yogyakarta: Deepublish.
- Id, I. (2021). *Machine Learning: Teori, Studi Kasus dan Implementasi Menggunakan Python (Vol. 1)*. Pekanbaru: Unri Press.
- Jumariana, I., & Sugiartawan, P. (2024). Identifikasi Pengenalan Pola Daun Kelor Kering Dengan You Only Look Once V8. *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, 14(1), 1-12. <https://doi.org/10.22146/ijeis.94871>
- Junianto, E., & Zuhdi, M. (2018). Penerapan Metode Palette untuk Menentukan Warna Dominan dari Sebuah Gambar Berbasis Android. *Jurnal Informatika*, 5(1), 61-72. <https://doi.org/10.31311/ji.v5i1.2740>
- Lestari, H. (2023). *Klasifikasi Citra Daun Tumbuhan Obat Menggunakan Deteksi Tepi Canny dan Metode K-Nearest Neighbor (KNN)*. Skripsi. Universitas Lampung.
- Maharani, A., & Bimantoro, F. (2020). Pengenalan Pola Tulisan Tangan Aksara Sasak Menggunakan Metode Linear Discriminant Analysis dan Jaringan Syaraf Tiruan Jenis Backpropagation. *Jurnal Teknologi Informasi, Komputer, Dan Aplikasinya (JTika)*, 2(2), 237-247. <https://doi.org/10.29303/jtika.v2i2.105>
- Marpaung, F., Aulia, F., & Nabila, R. (2022). *Computer Vision Dan Pengolahan Citra Digital*. Surabaya: Pustaka Aksara.
- Meiriyama, M., Devella, S., & Adelfi, S. (2022). Klasifikasi Daun Herbal Berdasarkan Fitur Bentuk Dan Tekstur Menggunakan KNN. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, 9(3), 2573-2584. <https://doi.org/10.35957/jatisi.v9i3.2974>
- Mulyana, D., & Pratama, A. (2023). Optimasi Deteksi Pengenalan Huruf Hijaiyah Dengan Metode Tepi Canny Dan Morfologi. *INTECOMS: Journal of Information Technology and Computer Science*, 6(3), 717-725. <https://doi.org/10.31539/intecom.v6i2.7663>
- Nafi'iyah, N. (2015). Algoritma Kohonen dalam Mengubah Citra Graylevel Menjadi Citra Biner. *Jurnal Ilmiah Teknologi Informasi Asia*, 9(2), 49-55.

- Narulita, S., Aulia, N., Handawati, R., Kusumawati, L., & Salsabila, Q. (2023). Penguatan Peran Masyarakat Bidara Cina dalam Adaptasi dan Mitigasi Bencana Banjir. *Sivitas: Jurnal Pengabdian dan Pemberdayaan Masyarakat*, 3(2), 57-63. <https://doi.org/10.52593/svs.03.2.01>
- Nurdin, R. (2023). Tumbuhan Bidara Dalam Al-Qur'an Dan Manfaatnya Bagi Kehidupan. *Al-Kauniah*, 4(2), 1-21. <https://doi.org/10.56874/alkauniah.v4i2.1448>
- Paturrehman, A., & Wijaya, I. (2020). Analisis pengenalan pola daun berdasarkan fitur canny edge detection dan fitur GLCM menggunakan metode klasifikasi K-Nearest Neighbor (K-NN). *Publikasi Tugas Akhir S-1 PSTI FT-UNRAM. J-Cosine*, 5(1), 68-76. <https://doi.org/10.29303/jcosine.v5i1.388>
- Prawira, A., Jayanta, J., & Widiastiwi, Y. (2021). Penerapan Metode Gray Level Co-Occurance Matrix dan Algoritma Support Vector Machine pada Klasifikasi Tanaman Bidara berdasarkan Tekstur Daun. In *Prosiding Seminar Nasional Mahasiswa Bidang Ilmu Komputer dan Aplikasinya*, 2(1), 569-578.
- Raharjeng, S., & Masliyah, A. (2020). Identifikasi morfologi bidara (*Ziziphus mauritiana*) di wilayah Sidoarjo. *Afamedis*, 1(2), 79-88.
- Rahma, F., & Hajijah, S. (2023). Inovasi Pembuatan Jelly dari Daun Bidara (*Ziziphus Mauritiana*) Sebagai Sumber Antioksidan. *Cendekia Sambas*, 1(1), 76-84. <https://doi.org/10.47767/agroindustri.v2i1.480>
- Rahmawati, E., Saifudin, S., Kesuma, C., & Rais, A. (2020). Rancang Bangun sistem informasi rekam medik studi kasus: UPTD Puskesmas Padamara Kabupaten purbalingga. *Indonesian Journal on Software Engineering (IJSE)*, 6(1), 133-144. <https://doi.org/10.31294/ijse.v6i1.7860>
- Ratna, S. (2020). Pengolahan citra digital dan histogram dengan phyton dan text editor pycharm. *Technologia: Jurnal Ilmiah*, 11(3), 181-186. <https://doi.org/10.31602/tji.v11i3.3294>
- Rosaly, R., & Prasetyo, A. (2019). Pengertian Flowchart Beserta Fungsi dan Simbol-simbol Flowchart yang Paling Umum Digunakan. *Jurnal Program Studi Teknik Informatika Politeknik Purbaya*, 1-8.
- Saputra, D., Pranata, T., & Handani, S. (2016). Prototype Aplikasi Pengolah Citra Invert Sebagai Media Pengolah Klise Foto. *Conference on Information Technology, Information System and Electrical Engineering*, 84-89.

- Satria, D. (2023). Pengantar Teknik Komputer: Konsep dan Prinsip Dasar. Jambi: PT. Sonpedia Publishing Indonesia.
- Sitohang, B., Sindar, A., & Nusantara, S. (2020). Analisis dan perbandingan metode sobel edge detection dan prewit pada deteksi tepi citra daun srilangka. *Jurnal Nasional Komputasi dan Teknologi Informasi*, 3(3), 314-322.
<https://doi.org/10.32672/jnkti.v3i3.2511>
- Supiyandi, S., Mujib, M., Azis, K., Abdillah, R., & Iskandar, S. (2024). Penerapan Teknologi Pengolahan Citra dalam Analisis Data Visual pada Tinjauan Komprehensif. *Jurnal Kendali Teknik dan Sains*, 2(3), 179-187.
<https://doi.org/10.59581/jkts-widyakarya.v2i3.3796>
- Supiyandi, S., Silalahi, D., Sari, D., Prahasti, R., & Putra, D. (2024). Analisis Performa Algoritma Pendeteksian Tepi pada Citra Multispektral. *Jurnal Sistem Informasi dan Ilmu Komputer*, 2(3), 1-12.
<https://doi.org/10.59581/jusiik-widyakarya.v2i3.3472>
- Yana, A. (2020). Implementasi Pengolahan Citra Digital Pada Penghitungan Anak Burung Puyuh Menerapkan Metode Blob. *Journal of Information System Research (JOSH)*, 1(4), 237-245.