

Kinerja Algoritma Canny untuk Mendeteksi Tepi dalam Mengidentifikasi Tulisan pada Citra Digital Meme

Nisfal Filsa¹, Widodo², Bambang P. Adhi³

^{1,2,3} Pendidikan Teknik Informatika dan Komputer Fakultas Teknik
Universitas Negeri Jakarta
¹nisfalfilsa@yahoo, ²widodo@unj.ac.id, ³bambangpadhi@unj.ac.id

ABSTRAK

Citra digital Meme merupakan sarana penyampaian informasi, teks pada Meme sebagian besar akan bergabung dengan latar pada gambar. Untuk membedakan latar dan teks dapat dilakukan dengan deteksi Tepi. Algoritma Canny merupakan salah satu algoritma deteksi Tepi yang memiliki tingkat kesalahan yang minimum dan menghasilkan citra tepian yang optimal. Salah satu penggunaan deteksi Tepi dapat diterapkan citra digital Meme untuk menentukan wilayah teks yang terdapat pada citra Meme. Hasil algoritma Canny mendeteksi Tepi untuk menentukan wilayah tulisan pada Meme lalu diidentifikasi menggunakan pengenalan karakter optis (OCR) akan dijadikan perhitungan untuk menilai kinerja algoritma deteksi Tepi Canny. Kinerja algoritma Canny mendeteksi Tepi untuk menentukan wilayah kandidat teks meningkatkan akurasi deteksi tulisan pada OCR (Object Character Recognition) dengan akurasi keberhasilan secara keseluruhan sebesar 65,47% dibandingkan dengan deteksi tulisan langsung menggunakan OCR sebesar 47,91%. Selain itu mengurangi tingkat kesalahan deteksi tulisan pada OCR dengan akurasi kesalahan secara keseluruhan yaitu kehilangan karakter sebesar 34,53% dan kelebihan karakter sebesar 35,98% dibandingkan deteksi tulisan langsung menggunakan OCR dengan akurasi kehilangan karakter sebesar 52,09% dan kelebihan karakter sebesar 52,62. Kinerja algoritma Canny mendeteksi wilayah kandidat teks pada OCR secara keseluruhan meningkatkan akurasi kebenaran dalam mendeteksi tulisan pada citra digital Meme dan mengurangi persentase kesalahan.

Kata Kunci: Algoritma Deteksi Tepi Canny, Pengenalan Karakter Optis, Deteksi Tulisan, Deteksi Tepi, Citra Digital, Meme

1. PENDAHULUAN

Citra (image) merupakan salah satu komponen multimedia yang mempunyai peranan sangat penting sebagai suatu bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi.

Pengolahan citra digital merupakan pengolahan dan analisis yang banyak melibatkan persepsi visual. Citra digital dapat diperoleh secara otomatis dari sistem penangkapan citra membentuk matrik yang elemen-elemennya menyatakan nilai intensitas cahaya atau tingkat keabuan suatu piksel. Pengolahan citra adalah salah satu aplikasi yang dapat mengubah gambar menjadi suatu informasi. Pengolahan citra bertujuan untuk memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (dalam hal ini komputer). Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra yang lain. Jadi masukannya adalah citra dan keluarannya juga citra. Namun citra keluaran

mempunyai kualitas lebih baik dari pada masukan. Pada proses pengolahan citra ini dikenal salah satu teknik yaitu Object Analysis yang terdiri dari beberapa konsep yaitu Edge Detection, Hough Transform, Corner Detection, Boundry Tracing in Images dan Quadtree Decoposition.

Tepi (Edge) adalah perubahan nilai intensitas derajat keabuan yang cepat atau tiba-tiba (perubahan besar) dalam jarak yang singkat. Sedangkan Deteksi Tepi (Edge Detection) pada suatu citra adalah sebuah proses yang menghasilkan tepi-tepi dari obyek-obyek citra tujuannya untuk menandai bagian yang menjadi detail citra.

Terdapat banyak jenis algoritma deteksi Tepi, seperti Canny, Sobel, Prewitt, Robert, dan Laplacian of Gaussian (LoG). Algoritma Canny merupakan salah satu algoritma deteksi Tepi modern. Deteksi Tepi Canny ditemukan oleh Marr dan Hildreth ypada tahun 1980 yang meneliti pemodelan persepsi visual manusia. Kelebihan dari

algoritma Canny yaitu mendeteksi karakter dengan baik, melokalisasi karakter dengan baik, dan respon yang jelas dengan satu respon untuk setiap Tepi. Dalam beberapa penelitian sebelumnya algoritma Canny sudah digunakan dalam Deteksi Kanker, Deteksi Plat Nomor dan Automatically Detect and Recognize Text in Natural Image.

Gambar Meme memiliki Tepi yang lebih rumit dibanding gambar plat nomor karena background pada Meme. Algoritma Canny akan mengubah gambar Meme menjadi Tepi untuk diimplementasikan pada program. Oleh karena itu, penelitian ini membahas kualitas kinerja algoritma Canny dalam mendeteksi Tepi dalam mengidentifikasi tulisan pada Meme yang diasumsikan dapat mendeteksi dan memfilter tulisan dengan hasil lebih baik.

2. Dasar Teori

2.1. Kajian Teoritis

2.1.1. Definisi Algoritma

Algoritma merupakan serangkaian langkah-langkah yang tersusun secara sistematis yang berdasarkan logika dan mudah dimengerti yang tujuannya untuk memecahkan masalah.

Sedangkan menurut Ema Utami dan Suwanto bahwa algoritma merupakan urutan atau deskripsi langkah-langkah penyelesaian masalah yang tersusun secara logis, ditulis dengan notasi yang mudah dimengerti sedemikian sehingga langkah-langkah tersebut dapat dilaksanakan oleh pemroses^[1].

2.1.2. Definisi Algoritma Canny

Menurut Fajar Astuti bahwa, algoritma Canny memiliki kelebihan dapat memberikan hasil deteksi Tepi yang optimal dan sekaligus juga memberikan fleksibilitas. Algoritma deteksi Tepi Canny memiliki tahap-tahap sebagai berikut:^[2]

1. Penghalusan dengan Gaussian filter untuk mengurangi noise yang ada pada gambar
2. Cari turunan pertama arah horizontal, S_x , dan vertical, S_y , dengan menggunakan operator sobel, kemudian hitung besaran atau magnitudenya.
3. Non-maximum Suppression, sesuai dengan tujuan Canny adalah mendeteksi lokasi Tepi yang tepat, maka non-maximum suppression bertujuan untuk mempertahankan puncak (dan ekuivalen dengan diferensiasi tegak lurus terhadap Tepi). Non-maximum suppression pada dasarnya adalah mencari titik tertinggi pada data besaran Tepi, ini diperoleh dengan menggunakan informasi arah Tepi untuk memeriksa apakah titik tersebut berada pada puncak punggung bukit (ridge).
4. Hysteresis Thresholding

- Jika gradient pada sebuah piksel diatas nilai 'High', piksel tersebut merupakan 'edge pixel'
- Jika gradient pada sebuah piksel dibawah nilai 'Low', piksel tersebut bukan 'edge pixel'
- Jika gradient pada sebuah piksel antara nilai 'Low' dan 'High', maka piksel tersebut termasuk 'edge pixel' jika dan hanya jika piksel tersebut terhubung dengan sebuah 'edge pixel' secara langsung atau melalui piksel-piksel antara 'Low' dan 'High'

2.1.3. Definisi Deteksi Tepi

Deteksi Tepi (*Edge Detection*) adalah operasi yang dijalankan untuk mendeteksi garis Tepi (*edges*) yang membatasi dua wilayah citra homogen yang memiliki tingkat kecerahan yang berbeda.

Menurut Sutoyo, operator-operator dalam deteksi tepi antara lain:^[3]

1. Operator Roberts

Operator Robert adalah operator yang berbasis gradient yang menggunakan dua buah kernel yang berukuran 2x2 piksel. Operator ini mengambil arah diagonal untuk penentuan arah dalam penghitungan nilai gradient, sehingga sering disebut dengan operator silang. Operator Roberts merupakan variasi dari rumus Gradient Operator dengan arah orientasi sebesar 45 derajat dan 135 derajat pada bidang citra. Ini berarti gradient dihitung dengan memanfaatkan titik yang berada pada arah orientasi 45 dan 135 yaitu :

$$f(x + 1, y + 1) \text{ dan } f(x - 1, y + 1)$$

Selain itu operator ini merupakan penjabaran dari teknik diferensial pada arah horisontal dan diferensial pada arah vertikal dengan menambahkan proses konversi biner dengan meratakan distribusi warna hitam dan warna putih. Kernel filter yang digunakan dalam metode Robert ini adalah:

$$H = \begin{bmatrix} 1 & -1 \end{bmatrix} \text{ dan } V = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

2. Operator Prewits

Pengembangan dari gradient operator dengan menggunakan 2 mask (horizontal dan vertikal) ukuran 3x3. Pada operator ini kekuatan gradient ditinjau dari sudut pandang horizontal dan vertikal (memperhatikan titik disekitar pada posisi horizontal dan vertikal). Selain itu metode Prewitt merupakan pengembangan metode Robert dengan menggunakan filter HPF yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi laplacian yang dikenal sebagai fungsi untuk membangkitkan HPF. Kernel filter yang digunakan dalam metode Prewitt ini adalah:

$$H = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ dan } V = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

3. Operator Sobel

Operator ini menggunakan dua buah kernel yang berukuran 3x3 piksel untuk perhitungan gradient sehingga perkiraan gradient berada tepat di tengah jendela. Satu cara untuk menghindari gradien yang dihitung pada titik interpolasi dari piksel-piksel yang terlibat adalah dengan menggunakan jendela 3x3 untuk perhitungan gradien, sehingga perkiraan gradien berada tepat di tengah jendela. Operator Sobel adalah operator yang paling banyak digunakan sebagai pelacak Tepi karena kesederhanaan dan keampuhannya. Selain itu metode Sobel merupakan pengembangan metode Robert dengan menggunakan filter HPF yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi laplacian dan gaussian yang dikenal sebagai fungsi untuk membangkitkan HPF. Kelebihan dari metode sobel ini adalah kemampuan untuk mengurangi noise sebelum melakukan perhitungan deteksi Tepi. Kernel filter yang digunakan dalam metode Sobel ini adalah:

$$H = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ dan } V = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

2.1.4. Definisi Identifikasi

Menurut Koenjatiningrat bahwa identifikasi adalah suatu bentuk pengenalan terhadap suatu ciri-ciri fenomena sosial secara jelas dan terperinci.^[4]

Identifikasi berarti dorongan untuk menjadi identik dengan orang lain. Identifikasi merupakan proses pengenalan terhadap sesuatu yang secara terperinci dan seidentik mungkin dengan aslinya.

2.1.5. Definisi Tulisan

Menurut Lado bahwa tulisan adalah simbol grafis yang diletakkan dalam kertas atau media menulis lain sebagai upaya untuk mengungkapkan perasaan yang bisa orang lain mengerti apa yang ingin kita sampaikan.^[5]

Tulisan terdiri dari huruf, angka dan tanda-tanda lainnya sesuai dengan bahasa masing-masing. Sejarah menca tat bahwa bahasa telah berkembang secara berbeda pada tiap peradaban manusia. Awal mula tulisan diketahui pada masa proto dengan sistem ideografik dan simbol menemonik. Berdasarkan sejarah tulisan dibagi menjadi dua kategori yaitu zaman perunggu dan modern.

2.1.6. Definisi Citra Digital

Citra (*image*) adalah gambar pada bidang dua dimensi. Dalam tinjauan matematis, citra merupakan fungsi kontinu dari intensitas cahaya pada bidang dua dimensi. Ketika sumber cahaya menerangi objek, objek memantulkan kembali sebagian cahaya tersebut. Pantulan ini ditangkap oleh alat-alat pengindera optik, misalnya mata manusia, kamera, scanner dan sebagainya. Bayangan objek tersebut akan terekam sesuai intensitas

pantulan cahaya. Ketika alat optik yang merekam pantulan cahaya itu merupakan mesin digital, misalnya kamera digital, maka citra yang dihasilkan merupakan citra digital. Pada citra digital, kontinuitas intensitas cahaya dikuantisasi sesuai resolusi alat perekam.

2.1.7. Definisi Meme

Menurut Widya Arifianti, Meme menampilkan image picture semsantara rage comic biasanya terdiri dari beberapa panel gambar dan mengungkapkan ekspresi kemarahan. Istilah Meme sendiri pertama kali diperkenalkan oleh seorang ahli biologi asal Britania Raya, Richard Dawkins. Akar katanya berasal dari bahasa Yunani, yakni mimesis, yang berarti tiruan.^[6]

Meme sebagai suatu unit informasi budaya (berupa pemikiran, ide, gagasan, kebiasaan, lagu, fesyen) yang membentuk pola-pola kebudayaan tertentu. Meme bukanlah sekedar komik, ini berawal dari sebuah fans page yang menyebut diri mereka adalah Meme Comic .

Walaupun Meme itu memang bisa berbentuk apa saja namun komik pada Meme itu biasanya disebut *Rage Comic* dan pada umumnya Meme adalah sebuah image macro yaitu sebuah gambar yang diberi caption yang diambil dari quote tokoh terkenal dari sebuah film, untuk dijadikan Meme berdasarkan perasaan user yg membuatnya, dan para pembaca akan membaca gambar tersebut persis dengan nada tokoh asli tersebut dengan kalimat yang berbeda. Bukan cuma dari film, banyak faktor yg membuat sebuah gambar bisa menjadi *image macro* dalam lingkup internet meme.

2.2. Pengolahan Citra

Pengolahan citra merupakan proses pengolahan dan analisis citra yang banyak melibatkan persepsi visual. Pengolahan citra bertujuan untuk memanipulasi gambar, yang melingkup teknik-teknik untuk memperbaiki atau mengurangi kualitas gambar, menampilkan bagian tertentu dari gambar, membuat sebuah gambar yang baru dari gambar, membuat sebuah gambar yang baru dari beberapa bagian gambar yang sudah ada, dan beberapa teknik manipulasi gambar lainnya.

Menurut Rinaldi Munir, pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik ^[7].

2.2.1. Definisi Pengenalan Pola

Menurut Rinaldi Munir bahwa pengenalan pola bertujuan menentukan kelompok atau kategori pola berdasarkan ciri-ciri yang dimiliki oleh pola tersebut. Dengan kata lain, pengenalan pola membedakan suatu objek dengan objek lain. Terdapat dua pendekatan yang dilakukan dalam pengenalan pola: pendekatan secara statistik dan pendekatan secara sintaktik atau structural^[7].

2.2.2. Definisi Pengenalan Karakter Optis

Pengenalan karakter optis merupakan suatu aplikasi untuk mengidentifikasi tulisan sesuai dengan karakter atau huruf yang telah ditentukan berdasarkan pola matriks yang telah disimpan didalam basis data.

Menurut Shelly, Cashman dan Veermat, pengenalan karakter optis adalah kegiatan pembacaan karakter-karakter hasil ketikan, cetakan komputer, atau tulisan tangan dari dokumen-dokumen biasa serta bisa menerjemahkan gambar-gambar tersebut menjadi bentuk yang dapat diproses komputer^[8].

2.2.3. Definisi Segmentasi

Segmentasi merupakan teknik untuk membagi citra menjadi beberapa daerah (region) di mana setiap daerah memiliki kemiripan atribut. Operasi segmentasi area adalah operasi untuk mengidentifikasi semua piksel yang mempunyai intensitas yang sama, dengan cara lain mengelompokkan piksel-piksel tersebut ke dalam suatu range terdekat.

2.3. MSER

Menurut Aneta bahwa MSER (Maximally Stable Extremal Region) dipelopori oleh Matas dkk. MSER merupakan kumpulan dari wilayah berbeda yang di deteksi dari citra grayscale. Semua wilayah didefinisikan sebagai property external dari fungsi intensitas dalam wilayah dan diatas batas luarnya. MSER memiliki sifat yang membentuk kemampuan superiornya sama stabil dengan detector local. MSER tertutup dalam transformasi geometri perubahan intensitas.

Pohon komponen adalah struktur yang memperbolehkan deteksi MSER dalam sebuah citra, dan sebagai tambahan, menyusun dasar dari pelacakan MSER. Pohon komponen telah digunakan oleh Couprie dkk untuk implementasi pemisahan batas yang efisien. Pohon komponen adalah pohon yang berakar dan saling terhubung dan bisa dibangun untuk citra apapun dengan nilai pixel berasal dari rangkaian yang digunakan secara total. Tiap cabang dari pohon merepresentasikan wilayah yang terkoneksi dalam input citra.

Proses ini menghasilkan struktur data yang menyimpan area dari tiap komponen terkoneksi sebagai fungsi intensitas. Gabungan dari 2 komponen dapat dilihat sebagai eksistensi terminal dari komponen yang lebih kecil dan penyisipan dari semua piksel komponen yang lebih kecil ke yang lebih besar. Kemudian, level intensitas yang meruokan local minima dari tingkat perubahan fungsi area dipilih sebagai threshold yang menghasilkan MSER. Sebagai output, MSER direpresentasikan sebagai posisi dari local intensitas minimum (atau maksimum) dan threshold. ^[9]

2.4. Kerangka Konseptual

Berdasarkan kajian teori yang telah dijelaskan, perkembangan gambar Meme semakin pesat dengan didukung kemajuan teknologi informasi. Meme merupakan citra digital dan dapat disebar ke dunia maya khususnya media sosial dengan bebas dan mudah. Dengan kebebasan tersebut Meme dapat digunakan sebagai sarana komunikasi untuk menyebarkan informasi-informasi negatif. Maka diperlukan aplikasi untuk menyaring gambar Meme tersebut di dunia maya. Untuk membuat aplikasi tersebut menggunakan deteksi dan lokalisasi teks pada gambar. Tulisan pada Meme biasanya akan bergabung dengan background sehingga akan lebih rumit dalam deteksi dan lokalisasi teks. Untuk mendeteksi dan melokalisasi tulisan tersebut melakukan proses segmentasi untuk pembagian wilayah selanjutnya dilakukan pengenalan karakter optis. Pada proses pengenalan segmentasi dibutuhkan algoritma deteksi Tepi untuk mengubah gambar menjadi gambar biner (perubahan gambar berwarna atau abu-abu menjadi hitam-dan-putih). Maka diterapkan algoritma Canny edge detection pada aplikasi tersebut untuk menghasilkan gambar biner. Setelah proses binerisasi ini maka akan dihasilkan gambar biner yang akan dilakukan proses segmentasi, proses segmentasi bertujuan mengelompokkan pixel-pixel objek menjadi wilayah (region). Setelah gambar biner dibagi berdasarkan wilayah maka pengenalan karakter optis lebih maksimal karena bagian dari latar gambar yang tidak mengandung tulisan akan dihilangkan. Proses pengenalan pola optis akan menghasilkan tulisan yang diidentifikasi berdasarkan basis data. Huruf pada citra akan dibandingkan dengan hasil tulisan dari aplikasi. Selanjutnya hasil deteksi tulisan menggunakan aplikasi dengan algoritma Canny akan dinilai berdasarkan perbandingan nilai persentase kesamaan tulisan pada gambar Meme.

2. Metodologi Penelitian

3.1. Tempat, Waktu dan Subjek Penelitian

Penelitian di lakukan di laboratorium komputer Jurusan Teknik Elektro Universitas Negeri Jakarta, dimulai dari bulan Januari 2016 sampai bulan Maret 2016.

3.2. Populasi dan Sampel Penelitian

3.2.1. Populasi

Populasi adalah wilayah generalisasi yang terdiri atas: objek/ subjek yang mempunyai kualitas dan karakteristik tertentu yang ditetapkan oleh peneliti untuk dipelajari dan kemudian ditarik kesimpulannya^[10]. Populasi dalam penelitian ini adalah citra digital Meme yang berasal dari website-website penghasil Meme seperti 9gag.com, 1cak.com dan memegenerator.com serta mesin pencari Google.

3.2.2. Sampel

Menurut Sugiyono, sampel adalah bagian dari jumlah dan karakteristik yang dimiliki oleh populasi tersebut. Bila populasi besar, dan peneliti tidak mungkin mempelajari semua yang ada pada populasi, misalnya karena keterbatasan dana tenaga dan waktu, maka peneliti dapat menggunakan sampel yang diambil dari populasi itu.

Sampel yang digunakan dalam penelitian adalah lima puluh gambar yang terdiri dari lima kategori berdasarkan latar gambar yaitu manusia, hewan, benda, pemandangan dan kartun. Setiap kategori terdapat sepuluh sampel gambar.

Teknik pengambilan data yang digunakan dalam pengumpulan data penelitian adalah Non participant Observation, yaitu observasi yang peneliti tidak ikut secara langsung dalam kegiatan atau proses yang sedang diamati, artinya peneliti mengumpulkan data melalui catatan-catatan pribadi atau hasil karya seseorang, teknik ini disebut juga sebagai studi dokumenter.

2.3. Definisi Operasional

3.3.1. Variabel Penelitian

a. Variabel Bebas

Variabel bebas (*independent variable*) merupakan variabel yang dipilih oleh peneliti untuk mencari pengaruhnya terhadap variabel terikat (*dependent variable*). Variabel bebas dalam penelitian ini adalah Algoritma Deteksi Tepi Canny.

b. Variabel Terikat

Variabel terikat (*dependent variable*) merupakan variabel yang kehadirannya dipengaruhi oleh variabel yang lain. Variabel terikat dalam penelitian ini adalah hasil Identifikasi Tulisan Pada Citra Digital Meme.

2.4. Metode dan Rancangan Penelitian

3.4.1. Metode Penelitian

Metode yang akan digunakan adalah metode eksperimen. Menurut Arief penelitian eksperimen adalah penyelidikan ilmiah yang menuntut peneliti memanipulasi dan mengendalikan satu atau lebih variabel bebas serta mengamati variabel terikat, untuk melihat perbedaan yang sesuai dengan manipulasi variabel-variabel bebas tersebut. Tujuan utama eksperimen adalah untuk menetapkan apa yang mungkin terjadi.

Sedangkan menurut Sugiyono, Penelitian eksperimen adalah metode penelitian yang digunakan untuk mencari perbedaan perlakuan tertentu terhadap yang lain dalam kondisi yang terkendali.

2.4.2. Teknik Analisis Data

Berdasarkan Teknik yang digunakan adalah teknik perhitungan akurasi. Pertama-tama mencari akurasi tulisan pada setiap gambar dengan rumus sebagai berikut:

$$X_n = \frac{\text{karakter benar hasil aplikasi}}{\text{karakter hasil asli gambar}} \times 100\%$$

Dimana:

X_n = hasil akurasi data ke-n

Setelah akurasi pada setiap gambar telah didapatkan maka dihitung rata-rata akurasi berdasarkan kategori rumus sebagai berikut:

$$\begin{aligned} \text{Rata - rata akurasi} \\ = \frac{X_1 + X_2 + X_3 + \dots + X_k}{K} \times 100\% \end{aligned}$$

Dimana:

X_k = hasil akurasi data ke-k

K = banyak data pada kategori

Setelah akurasi pada setiap gambar telah didapatkan maka dihitung rata-rata akurasi dengan rumus sebagai berikut:

$$\begin{aligned} \text{Rata - rata akurasi} \\ = \frac{X_1 + X_2 + X_3 + \dots + X_n}{N} \times 100\% \end{aligned}$$

Dimana:

X_n = hasil akurasi data ke-n

N = banyak data

Untuk menghitung tingkat kesalahan dibagi menjadi dua yaitu berdasarkan karakter benar yang hilang dan karakter lebih yang tidak seharusnya ada pada citra. Rumus menentukan persentase karakter kurang:

$$Y = \frac{KA - KB}{KA} \times 100\%$$

Dimana:

Y = hasil persentase karakter hilang

KA = karakter hasil asli gambar

KB = karakter benar hasil aplikasi

Rumus menentukan persentase karakter Lebih:

$$Z = \frac{KX - KB}{KA} \times 100\%$$

Dimana:

Z = hasil persentase karakter lebih

KX = semua karakter hasil deteksi aplikasi

KA = karakter hasil asli gambar

KB = karakter benar hasil aplikasi

2.5. Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan adalah tes yaitu dengan menguji program dengan menggunakan data sample yang telah dicari, selanjutnya hasil dari program akan disimpan dan dihitung akurasinya. Tahapan pengumpulan data tersebut adalah sebagai berikut:

1. Langkah awal pada tahap pelaksanaan penelitian adalah peneliti mencari berdasarkan kategori gambar Meme secara acak di halaman web penghasil Meme dan mesin pencari Google.
2. Menguji setiap gambar pada program satu per satu lalu disimpan hasilnya.
3. Menghitung hasil akurasi kebenaran dan persentase kesalahan pada setiap gambar Meme.
4. Melakukan analisis data hasil akhir secara keseluruhan maupun berdasarkan kategori.

Dalam penelitian ini peneliti mengumpulkan data dengan menguji gambar pada program lalu menghitung nilai akurasi ketepatan deteksi tulisan dan persentase kesalahan dalam mendeteksi tulisan. Persentase kesalahan berdasarkan tulisan terdeteksi yang hilang atau lebih.

3. Hasil Penelitian Dan Pembahasan

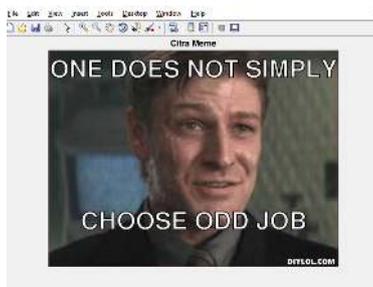
4.1. Deskripsi Data Hasil Penelitian

Data Penelitian yang terdiri dari pengumpulan dan pengelompokan data, konversi citra menjadi Tepi, proses deteksi wilayah berupa teks pada citra dengan Canny, proses pembacaan tulisan dengan OCR, hingga mendapatkan hasil teks pada citra, sebanyak lima puluh citra diuji dengan program implementasi algoritma deteksi Tepi Canny menentukan wilayah teks pada citra Meme. Hasil teks dari program akan dicocokkan dengan hasil pembacaan teks secara manual.

4.1.1. Implementasi Algoritma Deteksi Tepi Canny dalam Program

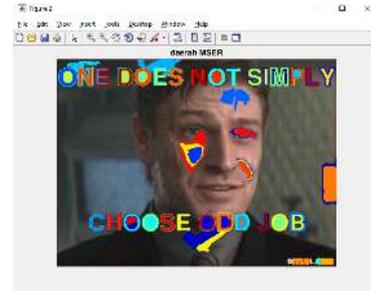
Dalam Berikut ini adalah salah satu proses citra uji dalam implementasi algoritma Canny mendeteksi Tepi untuk pemilihan wilayah tulisan pada citra dengan program. Program dibangun menggunakan Matlab, dan source code implementasi algoritma Canny dengan OCR (Object Character Recognition). Berikut ini adalah penjelasan proses citra dalam menghasilkan tulisan menggunakan program.

Citra dipilih dengan ukuran minimal 335 x 335 piksel dan maksimal 1028 x 1028 piksel dilakukan proses pengujian seperti Gambar 4.1.



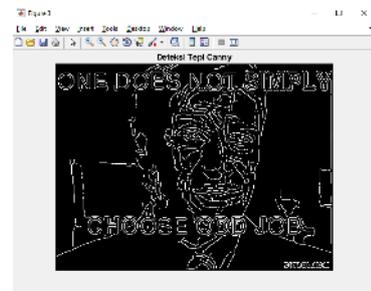
Gambar 4.1. Pengujian pada Citra

Sebelum dilakukan tahap implementasi algoritma Canny untuk deteksi Tepi, citra dilakukan deteksi dengan MSER (Maximally Stable Extremal Region) dengan memisahkan bagian-bagian wilayah berdasarkan warna dan kontras. Berikut hasil dari proses MSER pada gambar 4.2.



Gambar 4.2. Hasil Deteksi MSER

Setelah itu, citra yang berupa RGB akan dirubah menjadi grayscale dengan mengambil rata-rata dari bit RGB tiap pikselnya. Nilai intensitas grayscale tersebut disimpan dalam array pada suatu variabel dan setelah itu digunakan sebagai parameter fungsi untuk melakukan deteksi Tepi dengan memfilter citra dengan operator Sobel. Proses implementasi algoritma deteksi Tepi Canny dilakukan akan menghasilkan garis Tepi dari citra tersebut. Citra diubah menja Tepi seperti gambar 4.3.



Gambar 4.3. Hasil Algoritma Canny

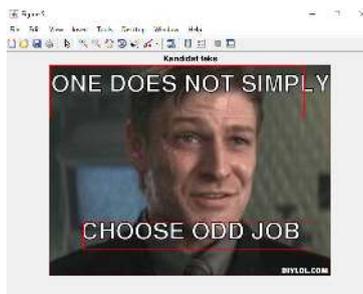
Setelah didapat hasil Tepi dari algoritma Canny maka selanjutnya dengan mengkombinasikan hasil citra Tepi Canny dengan hasil wilayah dari MSER untuk dilakukan filtering dan mengurangi background pada citra. Hasil dari kombinasi Canny dengan MSER pada gambar 4.4.



Gambar 4.4. Hasil Kombinasi MSER dan Canny

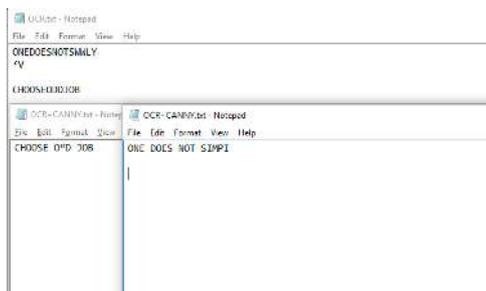
Setelah proses tersebut selanjutnya dilakukan filter kandidat teks dengan menganalisis komponen yang berhubungan antara citra hasil kombinasi MSER dan Canny dengan citra asli. Kandidat teks ditentukan apabila komponen piksel dari data asli yang berhubungan dengan data hasil. Tahap kedua dilakukan filter kandidat teks untuk mengidentifikasi bentuk garis Tepi dengan template

karakter yang ada. Hasil dari filter kandidat pada gambar 4.5



Gambar 4.5. Hasil Filter Kandidat Teks

Setelah menemukan kandidat teks maka tahap selanjutnya dilakukan pemotongan element yang berpotensi menjadi teks. Pemotongan dapat dilihat pada Gambar 4.5. berdasarkan tanda dari garis merah. Setelah dilakukan pemotongan element, setiap element diidentifikasi menggunakan OCR (Object Character Recognition) menggunakan Algoritma Template Matching Correlation berdasarkan data uji dari Tesseract untuk mendapatkan hasil tulisan yang terdapat pada citra dengan keluaran berupa teks pada notepad. Berikut gambar hasil akhir hasil program.



Gambar 4.6. Hasil Keluaran OCR dan OCR+Canny

4.1.2. Analisis Data Penelitian

Dalam Dari hasil akurasi keberhasilan akan menentukan kinerja Algoritma Canny untuk mendeteksi Tepi dalam mengidentifikasi tulisan pada citra digital Meme untuk menentukan wilayah dari tulisan. Setelah ditentukan lokasi suatu tulisan pada citra maka wilayah tersebut akan diidentifikasi menggunakan Object Character Recognition (OCR). Pada hasil penelitian ini akan dibahas akurasi pengenalan huruf dengan menggunakan identifikasi wilayah dengan Algoritma Canny yang selanjutnya diidentifikasi tulisan dengan OCR dan sebagai pembanding dilakukan pengujian dengan identifikasi tulisan hanya dengan OCR. Nilai akurasi dihitung berdasarkan tulisan benar yang terdapat pada citra dan kesalahan pengenalan tulisan yaitu kurang/lebih jumlah tulisan yang terdeteksi. Nilai akurasi rata-rata berdarkan 5 kategori dan rata-rata secara keseluruhan berada pada table 4.2.

Tabel 4.1. Hasil Teks pada Citra Uji

Perkategorori	Akurasi Kebenaran		Akurasi Kesalahan			
	OCR (%)	OCR + CANNY (%)	Kurang OCR (%)	Kurang OCR+ Canny (%)	Lebih OCR (%)	Lebih OCR+ Canny (%)
Manusia	43.60	76.26	56.40	23.74	29.42	29.67
Pemandangan	39.00	67.15	61.00	32.85	29.05	29.36
Kartun	57.99	51.33	42.01	48.67	56.70	29.50
Benda	60.84	68.22	39.16	31.78	102.76	35.35
Binatang	38.15	64.38	61.85	35.62	45.19	56.00
Keseluruhan	47.91	65.47	52.09	34.53	52.62	35.98

Dari data hasil pada Tabel 4.2. maksud dari akurasi kebenaran pada OCR merupakan persentase hasil karakter benar yang terdeteksi hanya dengan menggunakan OCR. Sedangkan OCR+CANNY merupakan persentase hasil karakter benar yang terdeteksi menggunakan pemilihan wilayah menggunakan algoritma Canny dan deteksi tulisan menggunakan OCR. Untuk akurasi kesalahan merupakan persentase kesalahan yaitu berupa kurang dan lebih. Maksud dari kurang adalah persentase karakter benar yang hilang pada tulisan. Maksud dari lebih adalah persentase karakter lebih yang seharusnya tidak terdapat pada tulisan.

Berdasarkan keterangan data pada Tabel 4.2. nilai persentase akurasi keberhasilan secara keseluruhan:

Dengan OCR:

- Pada Latar Manusia = 43.60%
- Pada Latar Pemandangan = 39.00%
- Pada Latar Kartun = 57.99%
- Pada Latar Benda = 60.84%
- Pada Latar Binatang = 38.15%
- Dengan OCR secara keseluruhan = 47.91%

Dengan OCR+Canny:

- Pada Latar Manusia = 76.26%
- Pada Latar Pemandangan = 67.15%
- Pada Latar Kartun = 51.33%
- Pada Latar Benda = 68.22%
- Pada Latar Binatang = 64.38%
- Dengan OCR+Canny keseluruhan = 65.47%

Persentase Tingkat Kesalahan:

Dengan OCR:

- Tulisan Benar Tidak Terdeteksi (Kurang):
- Pada Latar Manusia = 56.40%

Pada Latar Pemandangan	= 61.00%
Pada Latar Kartun	= 42.01%
Pada Latar Benda	= 39.16%
Pada Latar Binatang	= 61.85%
Dengan OCR secara keseluruhan	= 52.09%

Tulisan Yang Tidak Seharusnya Pada Citra (Lebih):

Pada Latar Manusia	= 29.42%
Pada Latar Pemandangan	= 29.05%
Pada Latar Kartun	= 56.70%
Pada Latar Benda	= 102.76%
Pada Latar Binatang	= 45.19%
Dengan OCR secara keseluruhan	= 52.62%

Dengan OCR+Canny:

Tulisan Benar Yang Tidak Terdeteksi (Kurang):

Pada Latar Manusia	= 23.74%
Pada Latar Pemandangan	= 32.85%
Pada Latar Kartun	= 48.67%
Pada Latar Benda	= 31.78%
Pada Latar Binatang	= 35.62%
Dengan OCR+Canny keseluruhan	= 34.53%

Tulisan Yang Tidak Seharusnya Pada Citra (Lebih):

Pada Latar Manusia	= 29.67%
Pada Latar Pemandangan	= 29.36%
Pada Latar Kartun	= 29.50%
Pada Latar Benda	= 35.35%
Pada Latar Binatang	= 56.00%
Dengan OCR+Canny keseluruhan	= 35.98%

4.3. Pembahasan Hasil Penelitian

Model Pada kategori latar berupa manusia, kinerja algoritma Canny mendeteksi Tepi untuk menentukan wilayah kandidat teks memiliki hasil akurasi kebenaran lebih besar dengan deteksi tulisan pada OCR (Object Character Recognition) sebesar 76,26% dibandingkan deteksi tulisan hanya dengan OCR sebesar 43,60%. Untuk tingkat kesalahan kehilangan karakter pada kinerja algoritma Canny memiliki hasil akurasi kesalahan lebih kecil pada deteksi tulisan OCR dengan hasil akurasi kehilangan sebesar 23,74% dibandingkan hasil OCR dengan hasil akurasi sebesar 56,40%. Untuk tingkat kesalahan kelebihan karakter kinerja algoritma Canny memiliki akurasi kesalahan lebih besar dengan deteksi tulisan OCR dengan hasil akurasi kelebihan sebesar 29,67% dibandingkan hasil OCR dengan hasil akurasi sebesar 29,42%.

Pada kategori latar berupa pemandangan, kinerja algoritma Canny mendeteksi Tepi untuk menentukan wilayah kandidat teks memiliki akurasi kebenaran lebih besar dengan deteksi tulisan pada OCR (Object Character Recognition) sebesar 67,15% dibandingkan deteksi tulisan hanya dengan

OCR sebesar 39,00%. Untuk tingkat kesalahan kehilangan karakter pada kinerja algoritma Canny memiliki hasil akurasi kesalahan lebih kecil pada deteksi tulisan OCR dengan hasil akurasi kehilangan sebesar 32,85% dibandingkan hasil OCR dengan hasil akurasi sebesar 61,00%. Untuk tingkat kesalahan kelebihan karakter kinerja algoritma Canny memiliki akurasi kesalahan lebih besar dengan deteksi tulisan OCR dengan hasil akurasi kehilangan sebesar 29,36% dibandingkan hasil OCR dengan hasil akurasi sebesar 29,05%.

Pada kategori latar berupa kartun, kinerja algoritma Canny mendeteksi Tepi untuk menentukan wilayah kandidat teks memiliki akurasi kebenaran lebih kecil dengan deteksi tulisan pada OCR (Object Character Recognition) sebesar 51,33% dibandingkan deteksi tulisan hanya dengan OCR sebesar 57,99%. Untuk tingkat kesalahan kehilangan karakter pada kinerja algoritma Canny memiliki hasil akurasi kesalahan lebih besar pada deteksi tulisan OCR dengan hasil akurasi kehilangan sebesar 48,67% dibandingkan hasil OCR dengan hasil akurasi sebesar 42,01%. Untuk tingkat kesalahan kelebihan karakter kinerja algoritma Canny memiliki akurasi kesalahan lebih kecil dengan deteksi tulisan OCR dengan hasil akurasi kehilangan sebesar 29,50% dibandingkan hasil OCR dengan hasil akurasi sebesar 56,70%.

Pada kategori latar berupa benda, kinerja algoritma Canny mendeteksi Tepi untuk menentukan wilayah kandidat teks memiliki akurasi kebenaran lebih besar dengan deteksi tulisan pada OCR (Object Character Recognition) sebesar 68,22% dibandingkan deteksi tulisan hanya dengan OCR sebesar 60,84%. Untuk tingkat kesalahan kehilangan karakter pada kinerja algoritma Canny memiliki hasil akurasi kesalahan lebih kecil pada deteksi tulisan OCR dengan hasil akurasi kehilangan sebesar 31,78% dibandingkan hasil OCR dengan hasil akurasi sebesar 39,16%. Untuk tingkat kesalahan kelebihan karakter kinerja algoritma Canny memiliki akurasi kesalahan lebih kecil dengan deteksi tulisan OCR dengan hasil akurasi kehilangan sebesar 35,35% dibandingkan hasil OCR dengan hasil akurasi sebesar 102,76%.

Pada kategori latar berupa binatang, kinerja algoritma Canny mendeteksi Tepi untuk menentukan wilayah kandidat teks memiliki akurasi kebenaran lebih besar dengan deteksi tulisan pada OCR (Object Character Recognition) sebesar 64,38% dibandingkan deteksi tulisan hanya dengan OCR sebesar 38,15%. Untuk tingkat kesalahan kehilangan karakter pada kinerja algoritma Canny memiliki hasil akurasi kesalahan lebih kecil pada deteksi tulisan OCR dengan hasil akurasi kehilangan sebesar 35,62% dibandingkan hasil OCR dengan hasil akurasi sebesar 61,85%. Untuk tingkat kesalahan kelebihan karakter kinerja algoritma Canny memiliki akurasi kesalahan lebih besar dengan deteksi tulisan OCR dengan hasil akurasi kehilangan sebesar

56,00% dibandingkan hasil OCR dengan hasil akurasi sebesar 45,19%.

Berdasarkan kategori latar gambar, kinerja algoritma Canny dengan OCR yang memiliki akurasi tingkat kebenaran terbaik adalah pada latar manusia sebesar 76,26% dan akurasi tingkat kebenaran terburuk pada latar kartun sebesar 51,33%. Sedangkan yang memiliki persentase tingkat kesalahan kehilangan karakter terkecil adalah pada latar manusia sebesar 23,74% dan persentase kehilangan karakter terbesar pada latar kartun sebesar 48,67%. Dan untuk akurasi tingkat kesalahan kelebihan karakter terkecil adalah pada latar pemandangan sebesar 23,74% dan akurasi kesalahan kelebihan karakter terbesar pada latar binatang sebesar 56,00%.

Dari 50 gambar Meme yang diuji, implementasi kinerja algoritma Canny mendeteksi Tepi untuk menentukan wilayah kandidat teks meningkatkan akurasi deteksi tulisan pada OCR (Object Character Recognition) dengan akurasi keberhasilan secara keseluruhan sebesar 65,47% dibandingkan dengan deteksi tulisan langsung menggunakan OCR sebesar 47,91%. Selain itu mengurangi tingkat kesalahan deteksi tulisan pada OCR dengan akurasi kesalahan secara keseluruhan yaitu kehilangan karakter sebesar 34,53% dan kelebihan karakter sebesar 35,98% dibandingkan deteksi tulisan langsung menggunakan OCR dengan akurasi kehilangan karakter sebesar 52,09% dan kelebihan karakter sebesar 52,62. Implementasi algoritma deteksi Tepi Canny untuk menentukan wilayah pada Meme secara keseluruhan meningkatkan kinerja OCR dalam mendeteksi tulisan pada Meme dan menguraingi persentase tingkat kesalahan kinerja OCR.

5 Kesimpulan Dan Saran

1.1. Kesimpulan

Berdasarkan Dari hasil penelitian yang telah diimplementasikan, maka dapat ditarik kesimpulan sebagai berikut:

1. Implementasi kinerja algoritma Canny mendeteksi wilayah kandidat teks pada OCR secara keseluruhan meningkatkan akurasi kebenaran dalam mendeteksi tulisan pada citra digital Meme.
2. Implementasi kinerja algoritma Canny mendeteksi wilayah kandidat teks pada OCR secara keseluruhan mengurangi tingkat akurasi kesalahan berupa kehilangan karakter dan kelebihan karakter dalam mendeteksi tulisan pada citra digital Meme.
3. Kinerja algoritma Canny mendeteksi wilayah kandidat teks dengan OCR pada kategori latar manusia memiliki hasil terbaik diantara kelima kategori. Sedangkan kategori latar kartun

merupakan kategori hasil terburuk diantara kelima kategori.

4. Analisis kinerja algoritma Canny mendeteksi wilayah kandidat teks dipengaruhi kontras warna latar gambar Meme sehingga mempengaruhi hasil Tepi.

1.2. Saran

Saran yang dapat dilakukan untuk penelitian selanjutnya menurut penulis antara lain:

1. Melakukan penelitian yang sama dengan objek ataupun kategori yang berbeda.
2. Dilakukan perbandingan dengan algoritma deteksi
3. Tepi lain untuk mengetahui hasil akurasi terbaik.
4. Dilakukan pembuatan aplikasi untuk dapat memilih gambar yang mengandung tulisan negatif.
5. Dilakukan kombinasi dengan spell checker untuk pembacaan teks untuk mendapatkan hasil kata yang lebih baik apabila kata kehilangan beberapa karakter.

6 DAFTAR PUSTAKA

- [1] Suwanto Raharjo dan Ema Utami. 2004. *Logika, Algoritma dan Implementasinya dalam Bahasa Python di GNU/LINUX*. Yogyakarta : Andi
- [2] Astuti Hermawati, Fajar. 2013. *Pengolahan Citra Digital*. Yogyakarta : Andi
- [3] Sutoyo, dkk. 2009. *Teori Pengolahan Citra Digital*. Yogyakarta : Andi
- [4] Koenjtaraningrat. 1981. *Metode Penelitian Masyarakat*. Jakarta : Balai Pustaka
- [5] Lado, Robert. 1964. *Language Teaching: A Scientific Approach*. New York: McGraw-Hill.
- [6] Arifanti, Widya. 2015. *Meme Comic Indonesia*. Jakarta: Cahaya Insan Suci
- [7] Munir, Rinaldi. 2004. *Pengolahan Citra Digital Dengan Pendekatan Algoritmik*. Bandung : Informatika
- [8] Shelly, B., Cashman, J., & Veermat, Misty. 2007. *Discovering Computers Ed 3*. Jakarta : Salemba Infotek
- [9] Aneta. 2012. *Perbandingan Pelacakan Wayang dengan MSER dan MSER Efisien*. Depok : Universitas Indonesia
- [10] Sugiyono. 2012. *Metode Penelitian Pendidikan, Pendekatan Kuantitatif, Kualitatif dan R&D*. Bandung : Alfabeta