

PENGGUNAAN METODE *CANNY EDGE DETECTION* UNTUK MENDETEKSI TEPI UANG KERTAS ASLI BERBASIS JAVA

Hermansyah Alam¹⁾, Yendri Manurung²⁾, Helma Widya³⁾,
Wahyul Huda Nanda⁴⁾

¹⁾Dosen Tetap Teknik Elektro, Fakultas Teknik UNPAB,

²⁾Alumni Teknik Informatika ITM,

³⁾Dosen tetap Bisnis Digital LP3I Medan,

⁴⁾Alumni Teknik Informatika ITBI Medan

Abstrak

Aplikasi object recognition adalah proses identifikasi citra pada uang kertas berdasarkan RGB. Dalam proyek akhir ini akan merancang bangun untuk mengidentifikasi tepi uang kertas dengan menggunakan metode canny. Dalam pembagian warna umumnya harus dilakukan inisialisasi jumlah yang di inginkan terlebih dahulu. Peredaran uang palsu dari tahun ke tahun terus mengalami peningkatan. Peningkatan ini karena mudahnya mendapatkan informasi cara membuat uang palsu di internet. Untuk itu, perlu adanya suatu teknologi yang dapat mengetahui dan membedakan uang palsu tersebut. Berdasarkan kasus di atas, perlu membuat aplikasi yang dapat membedakan uang asli dengan uang palsu menggunakan deteksi tepi uang dengan metode canny. Metode Canny Edge Detection akan mendeteksi semua edge atau garis-garis yang membentuk objek gambar dan akan memperjelas kembali pada bagian-bagian tersebut. Tujuan pendeteksian ini adalah bagaimana agar objek di dalam gambar dapat dikenali dan disederhanakan dari bentuk sebelumnya. Hal ini akan menghasilkan sebuah tampilan gambar yang berbeda dengan menampilkan efek relief didalamnya. Efek relief adalah seperti sebuah tampilan batu kasar yang diukir, yaitu garis-garis kasar yang membentuk sebuah penggambaran objek di dalamnya sehingga dapat menentukan perbedaan uang asli dan uang palsu.

Kata Kunci : *Metode Canny Edge Detection, Uang Asli, Java*

I. PENDAHULUAN

Kejahatan yang menggunakan teknologi sangat berkembang, Salah satu kejahatan yang memanfaatkan kemajuan teknologi adalah pembuatan uang palsu. Uang palsu yang beredar terdiri dari pecahan Rp.20.000 hingga pecahan Rp.100.000. Peredaran uang palsu dari tahun ke tahun terus mengalami peningkatan, Peningkatan ini karena mudahnya mendapatkan informasi cara membuat uang palsu di internet, untuk itu perlu adanya suatu teknologi yang dapat mengetahui dan membedakan Uang Asli dan uang palsu. Maka diciptakanlah alat untuk mendeteksi uang palsu tersebut. Berbagai macam teknologi digunakan antara lain menggunakan sinar ultraviolet, deteksi tepi dan lain-lain. Dengan pengenalan objek aplikasi *object recognition* yang telah ada salah satu nya proses indentifikasi citra menggunakan metode *CannyEdge Detection* akan mendeteksi semua *edge* atau garis-garis yang membentuk objek gambar dan akan memperjelas kembali pada bagian-bagian tersebut. Untuk mencapai hal tersebut di perlukan suatu perancangan perangkat keras atau lunak pendeteksian keaslian uang kertas dalam penelitian ini dirancang suatu perangkat lunak yang mampu mengenali keaslian uang kertas dengan teknik yang digunakan untuk membedakan uang palsu dengan uang asli adalah mendeteksi keberadaan benang pengaman, tanda air, perbedaan warna dan tekstur serta perbedaan bahan kertas. Salah satu teknik yang sering digunakan adalah dengan mendeteksi

keberadaan tanda air dari suatu mata uang kertas. Saat ini deteksi tepi muncul sebagai salah satu cara yang bisa membedakan uang asli dengan uang palsu dengan cara mendeteksi keberadaan tanda air pada suatu mata uang, untuk itu dalam proyek tugas akhir ini menggunakan aplikasi java untuk menyelesaikan masalah tersebut.

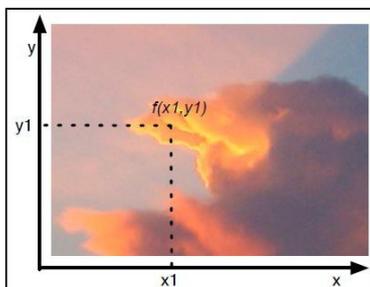
II. TINJAUAN PUSTAKA

2.1. Citra Digital

Citra adalah gambar dua dimensi yang dihasilkan dari gambar analog dua dimensi yang kontinu menjadi gambar diskrit melalui proses *sampling*. Gambar analog dibagi menjadi N baris dan M kolom sehingga menjadi gambar diskrit. Persilangan antara baris dan kolom tertentu disebut dengan piksel. Contohnya adalah gambar/titik diskrit pada baris n dan kolom disebut dengan piksel [n,m].

Citra didefinisikan sebagai intensitas cahaya dua dimensi $f(x,y)$, dimana x dan y menyatakan koordinat dan nilai f pada setiap titik (x,y) menyatakan intensitas atau kecerahan dari citra pada titik tersebut. Citra digital adalah citra $f(x,y)$ yang diubah koordinatnya dan kecerahannya kedalam bentuk diskrit. Citra digital dapat dianggap sebagai matrik dimana index baris dan kolomnya menyatakan kecerahan pada titik tersebut. Titik pada citra digital disebut pixel, jangkauan nilai f disebut jangkauan aras kelabu (kecerahan). Jika f hanya mempunyai dua nilai f maka disebut citra

biner. Citra digital 1 bit merupakan citra digital yang mempunyai jangkauan aras kelabu 1 bit atau mempunyai dua nilai intensitas (21). Nilai 0 menyatakan warna hitam dan nilai 1 menyatakan warna putih. Citra aras kelabu memperbolehkan mempunyai lebih dari dua nilai intensitas. Jangkauan aras kelabu dinyatakan bit dimana 1 bit menyatakan citra mempunyai 21 atau dua intensitas, 2 bit menyatakan citra mempunyai 22 atau empat nilai intensitas, dan seterusnya.



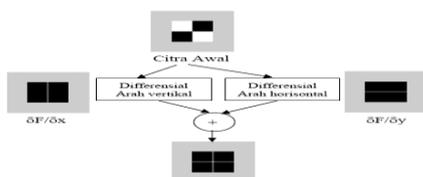
Gambar 1. Contoh Citra Digital

2.2. Deteksi Tepi

Deteksi tepi (*Edge Detection*) pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari obyek-obyek citra, tujuannya adalah :

- a) Untuk menandai bagian yang menjadi detail citra
- b) Untuk memperbaiki detail dari citra yang kabur, yang terjadi karena error atau adanya efek dari proses akuisisi citra

Suatu titik (x,y) dikatakan sebagai tepi (*edge*) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangganya. Gambar berikut ini menggambarkan bagaimana tepi suatu gambar diperoleh.



Gambar 2. Proses Deteksi Tepi Citra

2.3. Metode Canny

Salah satu algoritma deteksi tepi modern adalah deteksi tepi dengan menggunakan metode Canny. Deteksi tepi Canny ditemukan oleh Marr dan Hildreth yang meneliti pemodelan persepsi visual manusia. Ada beberapa kriteria pendeteksi tepian paling optimum yang dapat dipenuhi oleh algoritma Canny:

- 1. Mendeteksi dengan baik (kriteria deteksi) Kemampuan untuk meletakkan dan menandai semua tepi yang ada sesuai dengan pemilihan parameter-parameter konvolusi yang dilakukan. Sekaligus juga memberikan fleksibilitas yang sangat tinggi dalam hal menentukan tingkat deteksi ketebalan tepi sesuai yang diinginkan.

- 2. Melokalisasi dengan baik (kriteria lokalisasi)

Dengan Canny dimungkinkan dihasilkan jarak yang minimum antara tepi yang dideteksi dengan tepi yang asli.

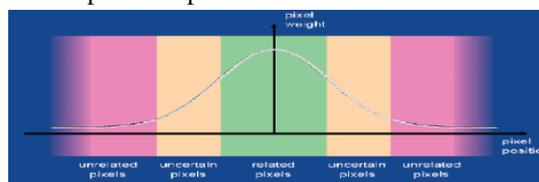
- 3. Respon yang jelas (kriteria respon)

Hanya ada satu respon untuk tiap tepi. Sehingga mudah dideteksi dan tidak menimbulkan kerancuan pada pengolahan citra selanjutnya. Pemilihan parameter deteksi tepi Canny sangat mempengaruhi hasil dari tepian yang dihasilkan. Beberapa parameter tersebut antara lain :

- a) Nilai Standart Deviasi Gaussian fungsi gaussian dapat digunakan untuk menghaluskan sebuah kurva 1 dimensi. Gaussian sendiri memiliki rumus sebagai berikut

$$G(X) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x - m)^2}{2\sigma^2}}$$

Dengan $\frac{1}{\sqrt{2\pi\sigma^2}}$ merupakan koefisien normalisasinya. Gaussian memiliki bentuk grafik beserta profile seperti di bawah ini



Gambar 3. Gaussian Blur

2.4. Uang Kertas

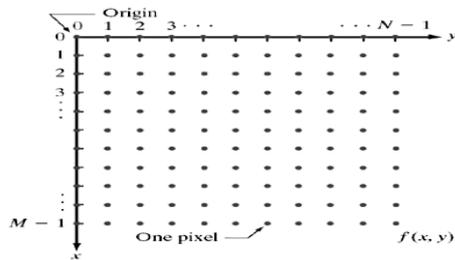
Uang kertas rupiah adalah uang dalam bentuk lembaran yang terbuat dari bahan kertas atau bahan lainnya (yang menyerupai kertas) yang dikeluarkan oleh pemerintah indonesia, dalam hal ini Bank Indonesia, dimana penggunaannya dilindungi oleh UU No.23 tahun 1999 dan sah digunakan sebagai alat tukar pembayaran di wilayah Negara Kesatuan Republik Indonesia (Wicaksono, 2008).



Gambar 4. Citra Uang Kertas

2.5. Koordinat Citra Digital

Sebuah citra digital dapat diwakili oleh sebuah matriks yang terdiri dari M kolom dan N baris, dimana perpotongan antara kolom dan baris disebut piksel (piksel = *picture element*) yaitu elemen terkecil dari sebuah citra. Piksel mempunyai dua parameter, yaitu koordinat dan intensitas atau warna. Nilai yang terdapat pada koordinat (x,y) adalah $f(x,y)$, yaitu besar intensitas atau warna dari piksel di titik itu. Oleh sebab itu citra digital dapat ditulis dalam bentuk matriks.



Gambar 5. Koordinat citra digital

2.6 . Jenis jenis Citra Digital

Citra digital dikelompokkan menjadi 4 jenis, yaitu :

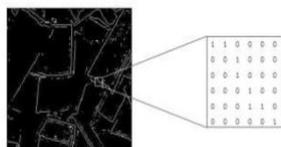
1 Citra Biner Masing-masing piksel hanya berwarna hitam dan putih . Oleh karena itu hanya terdapat dua kemungkinan yaitu hitam dan putih, maka hanya diperlukan satu bit per piksel. Oleh karena itu citra biner sangat efisien atau irit tempat penyimpanan (*storage*). Citra yang direpresentasikan seba gai citra biner sangat cocok digunakana untuk citra yang menggambarkan teks, *fingerprint* atau rencana arsitektural. Seperti terlihat pada Gambar di bawah ini:

$$I_{bin}(x,y) = \begin{cases} 0 & I_{gr}(x,y) < T \\ 255 & I_{gr}(x,y) \geq T \end{cases} \quad (4)$$

dan dalam bentuk *floating point*

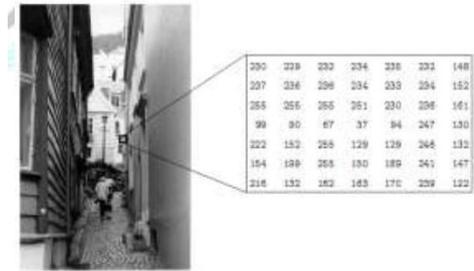
$$I_{bin}(x,y) = \begin{cases} 0 & I_{gr}(x,y) < T \\ 1 & I_{gr}(x,y) \geq T \end{cases} \quad (5)$$

dimana $I_{gr}(x,y)$ = nilai piksel Gray titik (x,y), $I_{bin}(x,y)$ = nilai piksel Binary titik (x,y), sedangkan T adalah nilai *threshold*.



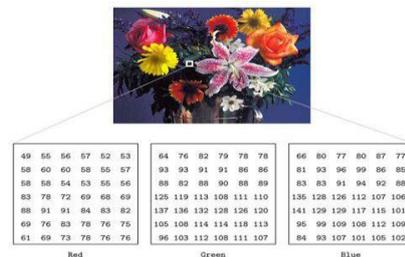
Gambar 6. Citra Biner

2 Citra abu-abu (*Grayscale*). Masing-masing piksel berisikan warna abu-abu dengan nilai normal antara 0 (hitam) sampai 255 (putih). *Range* tersebut berarti masing-masing piksel dapat direpresentasikan oleh nilai 8 bit atau 1 *byte*. Citra *Grayscale* dengan *range* nilai yang lain juga digunakan. Akan tetapi pada umumnya citra tersebut memiliki *range* pangkat dua dari 8 bit. Seperti terlihat pada Gambar di bawah ini.



Gambar 7. Citra Grayscale

3 Citra RGB. Untuk citra RGB masing-masing piksel mempunyai sebuah warna khusus. Warna dideskripsikan oleh kombinasi warna merah (*Red*), hijau (*Green*), biru (*Blue*). Jika masing-masing komponen (Merah, Hijau, Biru) mempunyai *range* antara 0-255, maka total *range* yang digunakan untuk citra RGB adalah 155 pangkat 3 atau 16.777.216 kemungkinan warna. Seperti terlihat pada Gambar 8 di bawah ini.



Gambar 8. Citra RGB

III. METODE PENELITIAN

3.1. Perangkat Keras (Hardware)

Adapun spesifikasi standar minimal kebutuhan perangkat keras yang akan dibutuhkan untuk menjalankan aplikasi adalah sebagai berikut:

1. Processor intel pentium III 800 Mhz.
2. Menggunakan RAM minimal 512 MB
3. Harddisk yang digunakan untuk menjalankan dan menyimpan data program
4. Monitor
5. Printer scanner

3.2. Perangkat Lunak (Software)

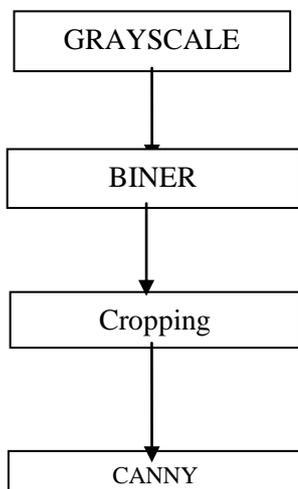
Perangkat lunak atau software yang digunakan dalam pembuatan identifikasi warna ini yaitu:

- a. Perangkat lunak aplikasi (*Application software*)
Program yang ditulis dengan diterjemahkan oleh language software untuk aplikasi tertentu dan untuk membantu pemakai komputer untuk menyelesaikan pekerjaannya dengan menggunakan *Java Netbeans*.
- b. Sistem operasi (*Operating System*)

Sutau sistem operasi yang terdiri dari kumpulan program-program sistem yang terorganisasi dan dibuat untuk membantu menghubungkan antara perangkat keras komputer dengan perangkat lunak yang dibuat oleh pemakai. *Operating system* yang digunakan windows 7

3.3 Prapengolahan Citra

Tahapan proses pengolahan citra dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Prapengolahan Citra

Pengolahan citra dimulai dengan mengubah citra warna (RGB) menjadi citra grayscale, kemudian dilakukan pengurangan derau pada citra tersebut untuk mendapatkan citra biner. Lalu di crop dan Tahap selanjutnya adalah proses *canny* untuk mengubah Kualitas *edge* yang dihasilkan atau morfologi garis tepi yang terbentuk.

3.4 Pembentukan Citra Keabuan (Grayscale)

Tahap pemrosesan citra dimulai dengan mengubah citra warna (RGB) menjadi citra keabuan. Nilai R (merah), G (hijau) dan B (biru) dari setiap piksel dijumlahkan dan dihitung rata-ratanya. Nilai dari proses ini adalah nilai keabuan (grayscale) 8 bit dengan rentang 0 (hitam) - 254 (putih). proses grayscale. Citra keabuan yang telah melalui proses penghilangan derau kemudian akan diubah menjadi citra biner, yaitu citra yang hanya memiliki dua nilai warna, yaitu 0 (hitam) dan 254 (putih). Pada tahap ini ditentukan nilai *threshold* yang akan menjadi batasan. Nilai *threshold* dimasukkan oleh pengguna sebelum proses dimulai. Pada penelitian ini nilai *threshold* yang digunakan adalah 190. Nilai yang lebih kecil dari *threshold* diubah menjadi 0 dan nilai yang lebih besar dari *threshold* diubah menjadi 254.

3.5. Pemotongan Citra (Cropping)

Pemrosesan citra selanjutnya adalah pemotongan citra (*crop*), yaitu menghilangkan area putih yang tidak berisi objek angka. Citra hasil *filling* dipotong sesuai dengan batasan objek angka tulisan

tangan, yaitu batas atas, batas bawah, batas kanan dan batas kiri. Dari batasan tersebut disisakan 1 piksel putih mengelilingi objek angka tulisan tangan. Hal ini dilakukan untuk membantu proses pengurusan objek (*thinning*).

3.6. Pembentukan Citra Canny

Pendekatan algoritma *canny* dilakukan dengan konvolusi fungsi gambar dengan operator gaussian dan turunan turunannya. Turunan pertama dari fungsi citra yang dikonvolusikan dengan fungsi gaussian, $g(x,y) = D[\text{gauss}(x,y) * f(x,y)]$ (5) ekuivalen dengan fungsi citra yang dikonvolusikan dengan turunan pertama dari fungsi gaussian, $g(x,y) = D[\text{gauss}(x,y)] * f(x,y)$ (6) Oleh karena itu, memungkinkan untuk mengkombinasikan tingkat kehalusan dan pendektasian tepi ke dalam suatu konvolusi dalam satu dimensi dengan dua arah yang berbeda (vertical dan horizontal) *Flowchart* proses *canny* dapat dilihat pada Gambar 2 di bawah ini.



Gambar 2..Flowchart Metode Canny

IV. HASIL PEMBAHASAN

Berdasarkan hasil pembahasan ada beberapa tahap untuk tampilan program dijelaskan tentang rancang bangun identifikasi nominal citra uang kertas Indonesia berdasarkan RGB yaitu sebagai berikut

4.1. Tampilan Menu Identifikasi

Merupakan Tampilan untuk memasukkan data uang kertas yang akan di cek warna berdasarkan RGB nya. Dan hasilnya akan bisa disimpan dalam data. Tampilan menu Indentifikasi dapat dilihat pada Gambar 3 berikut:

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari pembahasan yang telah dijelaskan maka dapat diambil kesimpulan sebagai berikut :

1. Aplikasi mendeteksi tepi uang kertas menggunakan metode **Canny Edge Detection** dapat mengetahui tingkat keaslian uang.
2. Software yang dapat digunakan untuk mengidentifikasi uang kertas yang berdasarkan warna pada RGB nya.

5.2 Saran

Saran yang bisa diberikan untuk pengembangan sistem ini adalah sebagai berikut:

1. Program aplikasi identifikasi uang kertas ini masih sederhana dan jauh dari sempurna, jadi diharapkan aplikasi ini nantinya dapat dikembangkan agar menjadi lebih baik lagi.
2. Evaluasi Program mendeteksi tepi uang kertas menggunakan metode canny dapat dilakukan dengan membandingkan metode deteksi tepi lainnya atau dengan metode kecerdasan buatan.

DAFTAR PUSTAKA

- [1]. Acmad Basuki, Josua F. Pelandi dan Fachurrochman, 2005, *Pengolahan Citra Digital*, Graha Ilmu.
- [2]. Ahmad, Usman. 2005, *Pengolahan Citra Digital & Teknik Pemrogramannya*, Graha Ilmu Yogyakarta.
- [3]. Edy Mulyanto, T.Sutoyo, Vincent Suhartono, 2009, *Teori Pengolahan Citra Digital*, Andi.
- [4]. Google. 2009. "Belajar Java: Perancangan Algoritma". Diunduh pada tanggal 13 Oktober 2013 di <http://www.java.lyracc.com/artikel/java-untuk-pemula/perancangan-algoritma>.
- [5]. Ginting, E. D. *Deteksi Tepi Menggunakan Metode Canny dengan Matlab untuk Membedakan Uang Asli dan Uang Palsu*. Jurusan Teknik Informatika Universitas Gunadarma.
- [6]. John Canny. *A computational approach to edge detection*. *Pattern Analysis and Machine Intelligence*, IEEE Transactions on, PAMI-8(6): 10-15.
- [7]. www.ilmu.komputer.com, Dasar-Dasar Pemrograman Java.pdf.



Gambar 3. Tampilan menu indentifikasi

4.2. Tampilan open file

Pada saat masuk ke menu identifikasi diminta untuk membuka file yang berisi pilih Citra gambar mata uang yang berguna untuk melakukan pengambilan file Citra dari komputer sebagai inputan. Tampilannya seperti berikut:



Gambar 4. Tampilan Open file

4.3. Tampilan Hasil Deteksi

Pada hasil deteksi akan menampilkan hasil dari gambar yang akan di analisa dan gambar yang teridentifikasi berupa hasil. Tampilan hasil deteksi dapat dilihat pada Gambar 5 berikut:



Gambar 5. Tampilan Hasil Deteksi

Dari Gambar 5. terlihat bahwa kejelasan uang asli tetap penampilan uang asli semula berdasarkan RGB menunjukkan gambar sebelum menggunakan Metode **Canny Edge Detection** tetap saja tidak ada perubahan sesuai aslinya dari hasil deteksinya.