PEMBUATAN SISTEM PENGOLAHAN CITRA DIGITAL UNTUK MENGHITUNG UANG KOIN MENGGUNAKAN TRANSFORMASI HOUGH BERBASIS OPENCV

[1]Hegi Tri Saputra, [2]Wendi Zarman
[12]Jurusan Sistem Komputer Unikom, Bandung
[1]hegitrisaputra@gmail.com, [2]wendizar@gmail.com

ABSTRAK

Keterbatasan waktu dan orang untuk menghitung uang koin disaat jumlah uang koin tersebut berjumlah banyak, bisa saja mengakibatkan kurangnya ketelitian dalam penghitungan dan kurangnya efisiensi dari segi waktu. Pada tugas akhir ini, sebuah kamera dari *smartphone* akan dimanfaatkan sebagai alat untuk menangkap citra uang koin, yang mana citra koin tersebut nantinya akan dihitung secara keseluruhan menggunakan aplikasi yang dirancang pada tugas akhir ini. Metode transformasi Hough digunakan sebagai metode untuk mendeteksi keberadaan uang koin pada citra. Proses berikutnya adalah identifikasi citra yang bertujuan untuk mengenali dan mengidentifikasi jenis nominal uang koin yang terdeteksi pada citra. Hasil dari pengujian yang didapat yaitu aplikasi mampu mendeteksi, mengenali dan menghitung uang koin dengan baik pada citra yang memiliki tingkat kekontrasan yang tinggi dengan nilai akurasi mencapai 90,33 %.

Kata kunci: citra, uang koin, *transformasi Hough.*

I. PENDAHULUAN

Seiring dengan pesatnya perkembangan ilmu pengolahan citra, banyak aplikasi yang dapat diciptakan dalam hal bertujuan untuk membantu pekerjaan manusia, salah satunya adalah bertujuan untuk membantu menghitung uang koin disaat berjumlah banyak. Saat ini terdapat beberapa jenis uang koin yang masih sering beredar di masyarakat seperti uang koin berbahan aluminium dengan nominal 100 rupiah, 200 rupiah dan 500 rupiah. Uang koin tersebut memiliki ukuran diameter yang berbeda-beda pada setiap nominalnya, hal ini akan dijadikan sebagai parameter utama dalam aplikasi yang akan dibangun untuk mengidentifikasi jenis nominal uang koin yang terdapat pada sebuah citra.

II. TEORI PENUNJANG

2.1 Citra

Sebuah citra adalah kumpulan piksel yang disusun dalam larik 2 dimensi. Indeks baris dan kolom (x,y) dari sebuah piksel dinyatakan dalam bilangan bulat. Piksel adalah unsur gambar atau representasi sebuah titik terkecil dalam sebuah gambar grafis yang dihitung per inchi. Piksel (0,0) terletak pada sudut kiri atas pada citra, indeks *x* bergerak ke kanan dan indeks *y* bergerak ke bawah. Konvensi ini dipakai merujuk pada cara penulisan larik yang digunakan dalam pemrograman komputer.

2.2 Citra Biner

Citra biner (*binary image*) adalah citra yang hanya mempunyai 2 nilai derajat keabuan yaitu hitam dan putih. Piksel objek bernilai 1 dan piksel *background* bernilai 0. Pada waktu menampilkan gambar, 0 adalah putih dan 1 adalah hitam. Karena hanya ada 2 kemungkinan nilai pada setiap piksel, maka yang diperlukan hanya 1 bit per piksel. Citra seperti ini sangat efisien untuk penyimpanan. Citra seperti ini bisa dimanfaatkan untuk representasi biner dari teks, tanda tangan, sidik jari, atau rancangan arsitektur.

2.3 Citra Grayscale

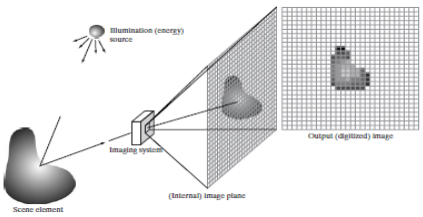
Citra abu-abu (citra *grayscale*) merupakan citra yang memiliki nilai intensitas dari 0 – 255, dimana 0 adalah putih dan 1 adalah hitam sedangkan nilai diantaranya merupakan level keabuan. Rentang nilai intensitas 0 (hitam) sampai 255 (putih) berarti bahwa setiap piksel dapat direpresentasikan oleh 8 bit, atau 1 *byte*. Citra *grayscale* bisa digunakan untuk merepresentasikan citra medis (sinar x), tulisan/buku, dan lain-lain. Citra *grayscale* dengan 256 level abu-abu dapat digunakan untuk mengenali kebanyakan objek.

2.3 Citra RGB

Citra RGB (*Red, Green, Blue*) merupakan sebuah citra yang dimana setiap pikselnya memiliki suatu warna khusus. Warna RGB merupakan model warna *aditif* yaitu ketiga berkas cahaya yang ditambahkan bersama-sama dengan menambahkan panjang gelombang untuk membuat spektrum warna akhir. Pada model warna RGB setiap piksel mengandung minimal 24 bit data, dengan komposisi yaitu 8 bit untuk merah, 8 bit untuk hijau, dan 8 bit untuk biru. Jika setiap komponen warna tersebut memiliki rentang intensitas 0 - 255, maka terdapat sejumlah 255^3 = 16.777.216 kemungkinan jenis warna pada citra ini. Karena dibutuhkan 24 bit per piksel, maka citra ini disebut pula dengan citra warna 24 bit.

2.4 Pengolahan Citra Digital

Citra digital adalah citra kontinu yang diubah dalam bentuk diskrit, baik koordinat ruang maupun intensitas cahayanya. Proses digitalisasi terdiri dari 2 tahap yaitu pencuplikan (*sampling*) dan kuantisasi intensitas. Pengolahan citra digital adalah suatu pemrosesan sebuah citra yang bertujuan untuk memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin dalam hal ini komputer. Citra digital merupakan sebuah fungsi 2 dimensi f (x,y), di mana x dan y adalah koordinat spasial dan *amplitude* dari f pada sembarang pasangan koordinat (x,y) disebut *intensity* (intensitas) atau *gray level* (level keabuan) dari citra pada titik tersebut. Ketika x,y dan nilai intensitas dari f adalah semua terbatas, *discrete quantities*, kita sebut citra tersebut *digital image* (citra digital).



Gambar 2.1 Pengolahan Citra Digital

2.4 Uang Koin

Uang koin rupiah yang digunakan dalam penelitian ini yaitu uang koin berbahan aluminium dimana setiap nominalnya memiliki ukuran jari-jari yang berbeda-beda. Uang koin dengan nominal Rp 100 memiliki ukuran jari-jari 11,5 mm, uang koin dengan nominal Rp 200 memiliki ukuran diameter 12,5 mm, uang koin dengan nominal Rp 500 memiliki ukuran jari-jari 13,5 mm.



Gambar 2.2 Uang Koin Berbahan Aluminium

Tabel 2.1 Jari-Jari Uang Koin Aluminium

Jenis Nominal	Jari-Jari
Rp 100	11,5 mm
Rp 200	12,5 mm
Rp 500	13,5 mm

2.5 Deteksi Tepi

Deteksi tepi adalah suatu proses untuk menghasilkan tepi-tepi dari sebuah obyek pada sebuah citra, yang bertujuan untuk menandai bagian yang menjadi detail citra dan untuk memperbaiki detail citra yang kabur akibat dari proses akuisisi citra yang tidak baik.

Suatu titik dikatakan sebagai sebuah tepi (edge) dari sebuah citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangganya.

2.6 Transformasi Hough

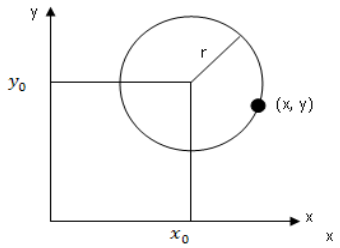
Prinsip kerja dari metode transformasi Hough sendiri adalah dengan mencari bentuk geometri yang paling sesuai dengan kumpulan titik pada suatu citra. Jika objek yang dicari adalah berupa lingkaran, maka digunakan transformasi lingkaran Hough. Prosedur yang digunakan dalam mendeteksi lingkaran adalah sama dengan transformasi Hough pada objek garis, tapi dikerjakan pada ruang dimensi yang lebih kompleks, yaitu dalam parameter ruang 3D (x_0,y_0,r) dimana x_0 dan y_0 merupakan koordinat pusat lingkaran dan r adalah jari-jari lingkaran seperti persamaan berikut :

$$(x-x_0)^2 + (y-y_0)^2 = r^2$$

Sedangkan untuk vektor parameternya adalah :

$$\vec{p} = \begin{bmatrix} x_0 \\ y_0 \\ r \end{bmatrix}$$

Dan jika diilustrasikan dalam koordinat (x_0,y_0, r) adalah sebagai berikut :



2.7 Microsoft Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak yang dapat digunakan untuk pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *console*, aplikasi *windows*, ataupun aplikasi web.

2.8 OpenCV

OpenCV (*Open Computer Vision*) adalah sebuah *library* bersifat *open source* dibawah lisensi BSD (*Berkeley Software Distribution*) yang memiliki sejumlah algoritma *computer vision* atau pengolahan citra didalamnya. *Computer vision* sendiri merupakan salah satu cabang dari bidang ilmu pengolahan citra (*Image Processing*) yang memungkinkan komputer untuk dapat melihat seperti halnya manusia. Dengan *vision* tersebut komputer dapat mengambil keputusan, melakukan aksi, atau mengenali suatu objek. Beberapa implementasi dari *computer vision* adalah *face recognition*, *face detection*, *face/object tracking*, *road tracking*, dan lain sebagainya.

2.9 QtCreator

QtCreator adalah sebuah IDE (*Integrated Development Software*) multi *platform*, termasuk juga sebuah *visual debugger* dan sebuah *integrated GUI layout and forms designer* untuk mendesain dan membangun *graphical user interface* (GUI) dari Qt widgets. Qt creator mendukung pembuatan aplikasi desktop (windows, linux, freeBSD dan mac os) dan aplikasi mobile.

III. PERANCANGAN SISTEM

Aplikasi untuk menghitung uang koin ini dirancang dengan menggunakan metode transformasi Hough. Berikut merupakan gambaran proses secara keseluruhan.



Gambar 3.1 Proses Secara Keseluruhan

Analisis Kebutuhan Proses

- Proses akuisisi citra.
- Proses pendeteksian lingkaran uang koin.
- Proses identifikasi nominal uang koin.
- Proses penghitungan nominal uang koin.

Analisis Kebutuhan Perangkat Lunak

- Microsoft Visual Studio 2010 digunakan untuk menuliskan program aplikasi.
- *Library* OpenCV 2.4.9 digunakan untuk menyediakan modul-modul *computer vision* yang dibutuhkan dalam proses pembuatan aplikasi.
- QtCreator 5.3.1 digunakan untuk membuat GUI (*Graphical User Interface*) aplikasi.

Analisis Kebutuhan Perangkat Keras

- Komputer dengan spesifikasi sistem operasi windows 7, prosesor core2duo 2,93 ghz, *RAM* berkapasitas 2 *GB*.
- Kamera *smartphone* dengan resolusi 1 MP dan 2 MP.
- Tempat untuk menaruh *smartphone* dengan ketinggian 50 cm dan alas tempat menaruh koin.

3.1 Diagram Blok

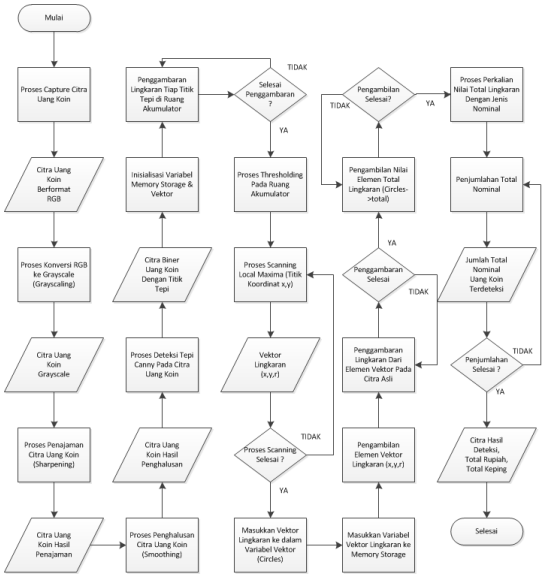


Gambar 3.2 Diagram Blok



Gambar 3.3 Proses *Capture* Koin

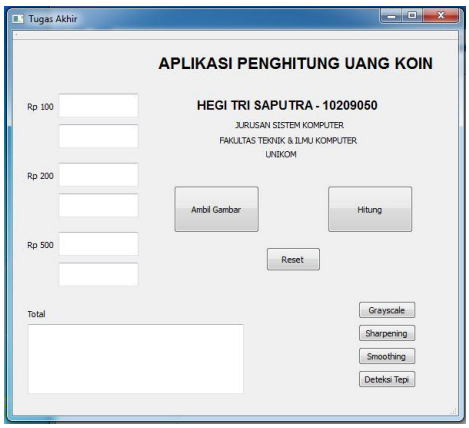
3.2 Diagram Alir



Gambar 3.4 Diagram Alir

3.4 Perancangan Tampilan Aplikasi

Berikut merupakan rancangan tampilan aplikasi penghitung uang koin, dimana terdapat dua tombol utama yaitu tombol ambil gambar untuk memasukkan citra input dan tombol hitung untuk melakukan proses penghitungan koin dan juga kotak *plain text* sebagai *output text* jumlah nominal dan jumlah keping baik per nominal maupun untuk total keseluruhan.



Gambar 3.5 Tampilan *Form* Aplikasi

Setiap citra koin yang terdeteksi akan ditandai dengan tanda titik bulat berwarna

yang berbeda sesuai dengan jenis nominalnya, seperti :

- Titik bulat merah untuk koin nominal Rp 100.
- Titik bulat hijau untuk koin nominal Rp 200.
- Titik bulat biru untuk koin nominal Rp 500.



Gambar 3.6 Penandaan Koin Dengan Titik Warna Sesuai Jenis Koin

3.5 Prosedur Pengujian

Prosedur pengujian/percobaan sendiri yaitu setiap citra yang diuji akan dibandingkan antara nilai hasil perhitungan hasil aplikasi dengan nilai hasil perhitungan secara manual. Terdapat beberapa variabel penilaian keberhasilannya seperti jumlah koin terdeteksi tepat, jumlah koin terdeteksi salah dan jumlah koin yang tidak terdeteksi. Variabel-variabel tersebut dihitung per nominal. Maksud dari variabel terdeteksi tepat adalah penandaan koin dengan warna yang telah ditetapkan tepat menandai koin tersebut, sedangkan maksud dari variabel terdeteksi salah adalah penandaan koin dengan warna yang telah ditetapkan tidak menandai koin yang seharusnya/menandai jenis koin lainnya, dan maksud dari variabel tidak terdeteksi adalah tidak ditandainya sebuah koin dengan tanda warna apapun. Berikut merupakan gambar contohnya.



Gambar 3.7 Koin Rp 500 Yang Terdeteksi Tepat



Gambar 3.8 Koin Rp 100 Yang Terdeteksi Salah (Titik Hijau Untuk Rp 200).



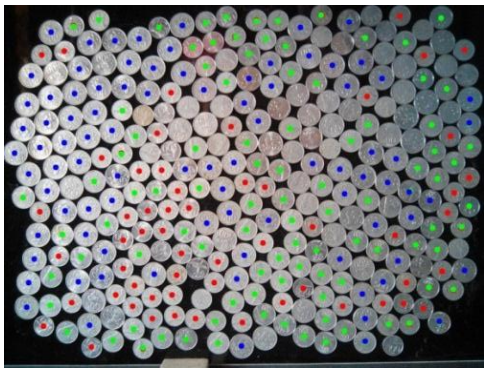
Gambar 3.9 Koin Rp 200 Yang Tidak Terdeteksi (Tidak Ada Titik Warna Pada Koin)

BAB IV. HASIL DAN ANALISA

4.1 Pengujian Citra

Berikut merupakan salah satu contoh pengujian aplikasi terhadap citra yang diambil menggunakan kamera *smartphone* pada siang hari di dalam ruangan kamar dengan cahaya matahari yang masuk cukup banyak/terang, dengan spesifikasi :

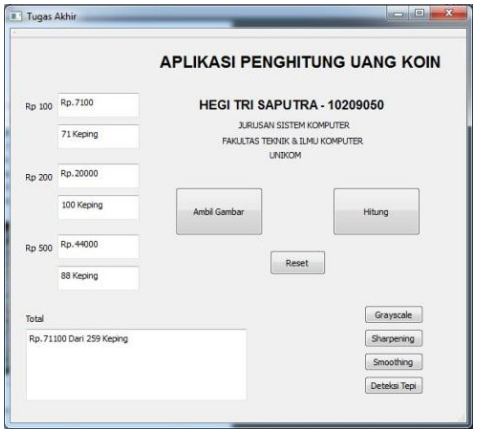
- Resolusi : 1 mega piksel (1024x768 piksel).
- Susunan koin : acak.
- Tingkat pencahayaan : terang.
- Jenis Koin : Rp 100, Rp 200 dan Rp 500 berbahan aluminium dengan jumlah masing-masing sebanyak 100 keping (total = 300 keping).



Gambar 4.1 Pengujian Terhadap Sebuah Citra Uji

Tabel 4.1 Penghitungan Manual

Jenis	Terdeteksi Tepat	Salah Men deteksi	Tidak Terdeteksi	Akurasi
Rp 100	63	0	37	63 %
Rp 200	81	21	19	81 %
Rp 500	70	9	30	70 %
Jumlah	234	30	86	



Gambar 4.2 Tampilan *Output* Pada Aplikasi

$$\sum \frac{\text{Jumlah koin terdeteksi tepat}}{\text{Jumlah koin yang diujikan}} \text{ x } 100\ \% = \frac{234}{300} \text{ x } 100\ \% = 71{,}33\ \%$$

Tabel 4.3 Tingkat Akurasi Pada Citra Dengan Resolusi 1 Mega Piksel

1 MP	Acak	Teratur
Terang	71,33 %	90,33 %
Redup	49,66 %	74 %

Tabel 4.4 Tingkat Akurasi Pada Citra Dengan Resolusi 2 Mega Piksel

2 MP	Acak	Teratur
Terang	90,33 %	84,33 %
Redup	71,33 %	65,33 %

4.2 Analisa Sistem

Berdasarkan data dari tabel tingkat akurasi pada hasil uji terhadap 8 citra diatas, bisa dilihat bahwa citra yang diambil saat tingkat pencahayaan yang terang memiliki nilai persentase akurasi yang lebih tinggi dibandingkan dengan dengan citra yang diambil disaat tingkat pencahayaannya redup, hal ini disebabkan oleh tingkat kontras yang tinggi antara warna koin dengan warna latar belakang. Semakin terang dan kontras antara citra dengan latar, semakin baik tingkat pendeteksian. Sedangkan untuk penyusunan koin tidak terlalu berpengaruh terhadap tingkat akurasi pendeteksian jika mengacu pada tabel diatas. Citra dengan resolusi 1 mega piksel (1024x768 piksel) yang diambil pada siang hari dengan kondisi susunan koin yang teratur dan pencahayaan yang terang memiliki tingkat akurasi yang paling tinggi diantara citra uji yang lain dengan nilai akurasi 90,33 %.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari pengujian dan analisa yang telah dilakukan dapat diambil beberapa kesimpulan, antara lain sebagai berikut :

1. Aplikasi dapat mendeteksi, mengidentifikasi dan menghitung uang koin berbahan aluminium dengan cukup baik dengan nilai akurasi tertinggi yaitu 90,33 %.
2. Aplikasi dapat mendeteksi dengan tingkat akurasi yang baik apabila tingkat kontras antara warna koin dan latarnya tinggi, maka waktu pengambilan citra yang baik adalah pada saat siang hari disaat pencahayaan cukup banyak/terang.
3. Aplikasi sulit mendeteksi lingkaran koin pada citra jika tingkat kontras antara koin dan latarnya rendah.
4. Susunan koin tidak terlalu berpengaruh terhadap tingkat akurasi.

5.2 Saran

Aplikasi yang dibangun masih jauh dari sempurna, oleh karena itu saran bagi yang ingin mengembangkan aplikasi ini kedepannya, antara lain :

- Pengambilan citra bisa langsung dilakukan pada aplikasi.
- Koin yang dideteksi lebih banyak.
- Koin yang dideteksi koin dengan jenis bahan yang lain, seperti koin berbahan kuningan.

VI. DAFTAR PUSTAKA

[1] Bradski, G, Learning OpenCV : Computer Vision With The OpenCV Library, California : O'Reilly, 2008.

[2] Anonim, *Hough Transform*, (online). diakses 16 Mei 2014 dari world wide web : http://en.wikipedia.org/wiki/Hough_transform.

[3] OpenCV dev team, *How To Build Application with OpenCV Inside The Microsoft Visual Studio* , (online). diakses 16 Mei 2014 dari world wide web : http://docs.opencv.org/doc/tutorials/introduction/windows_visual_studio_Opencv/windows_visual_studio_Opencv.html.

[4] OpenCV dev team, *Hough Circle Transform*, (online). diakses 16 Mei 2014 dari world wide web : http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/hough_circle/hough_circle.html.

[5] Anonim, Detecting Circles in Images Using OpenCV and Hough Circles, (online). diakses 16 Mei 2014 dari world wide web : http://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/.

[6] Anonim, *Qt Creator*, (online). diakses 7 Agustus 2014 dari world wide web : http://en.wikipedia.org/wiki/Qt_Creator.

[7] Anonim, *Create a Qt Widget Based Application*, (online). diakses 7 Agustus 2014 dari world wide web : https://qt-project.org/doc/qtcreator-2.5/creator-writing-program.html.

[8] Anonim, *Qt Documentation*, (online). diakses 8 Agustus 2014 dari world wide web : http://qtproject.org/wiki/Category:Tools::QtCreator.