

APLIKASI PENGENALAN RAMBU BERBENTUK BELAH KETUPAT

Andhika Pratama, Izzati Muhimmah

Laboratorium Komputasi dan Sistem Cerdas, Jurusan Teknik Informatika,
Fakultas Teknologi Industri, Universitas Islam Indonesia
Jl. Kaliurang Km. 14 Yogyakarta 55584
Telp. (0274) 895287 ext. 122, Faks. (0274) 895007 ext. 148
E-mail: dhicka_1k4@yahoo.com, emma@fti.uui.ac.id

ABSTRAK

Pada penelitian ini akan dikembangkan sebuah metode pengenalan rambu lalu lintas secara otomatis dengan menggunakan *distance set*. Tahapan awalnya yaitu proses deteksi tepi agar mendapatkan citra dalam bentuk binary. Hal ini akan mempermudah dalam pengenalan pola atau bentuk dari objek. Dari citra binary yang dihasilkan oleh proses deteksi tepi akan bisa mendapatkan titik koordinat (x,y) dari setiap pixel yang merepresentasikan garis tepi dari pola objek tersebut. Pada langkah selanjutnya dilakukan pelabelan dengan melakukan perhitungan jarak antara beberapa pixel terdekat. Sistem pengenalan rambu berbentuk belah ketupat dikembangkan dengan menggunakan bahasa pemrograman java. Sehingga diperoleh sebuah perangkat lunak yang dapat mengenali pola dan bentuk dari rambu berbentuk belah ketupat

Kata Kunci: Distance set, Pengenalan Rambu Lalulintas, Deteksi objek otomatis, shape recognition.

1. PENDAHULUAN

1.1 Latar Belakang

Selama ini, panduan berkendara yang populer digunakan masih memanfaatkan sinkronisasi sinyal satelit yang biasanya disebut dengan perangkat GPS (*Global Position System*). Dapat kita lihat bahwa, sistem tersebut belum dilengkapi panduan yang lengkap seperti halnya pendeteksi tanda jalan atau rambu rambu lalu lintas. Dalam pengembangannya disebut juga dengan *Vision Based Navigation* merupakan salah satu solusi dari permasalahan ini. atau juga disebut dengan *Traffic Sign Detection*.

Traffic Sign Detection merupakan suatu teknologi dimana mampu mengenali tanda-tanda lalu lintas, hal ini akan mendorong pengemudi untuk berkendara secara aman dan tidak membahayakan pengguna jalan lainnya.

Proses yang diterapkan pada *Traffic Sign Detection* ini yaitu *image processing*. *Image processing* merupakan proses pengolahan dan analisis citra yang banyak melibatkan persepsi visual. Proses ini mempunyai ciri data masukan dan informasi keluaran yang berbentuk citra. Istilah pengolahan citra digital secara umum didefinisikan sebagai pemrosesan citra dua dimensi dengan komputer (Wilhelm Burger, et al. 2008). Dalam definisi yang lebih luas, pengolahan citra digital juga mencakup semua data dua dimensi. Citra digital adalah barisan bilangan nyata maupun kompleks yang diwakili oleh bit-bit tertentu.

Pada proses pendeteksian bentuk objek dari rambu rambu diterapkan teknik *distance set*. Teknik ini bertujuan untuk melakukan pengenalan terhadap bentuk dari objek yang ingin dideteksi. Pengenalan ini disebut juga dengan *Shape Recognition* (Carmichael, et al. 2003).

Dari permasalahan yang terdapat pada teknologi sebelumnya yang masih menggunakan sinkronisasi dari sinyal satelit diperlukan sebuah pengembangan agar dapat mengetahui tanda-tanda lalu lintas sehingga dapat mempermudah pengemudi dalam perjalanan. Berdasarkan latar belakang tersebut, maka pada penelitian ini akan dibangun sebuah *tool* untuk mendeteksi tanda lalu lintas secara otomatis.

1.2 Tujuan

Aplikasi ini akan bisa mengenal rambu penyeberangan dengan baik.

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

- Untuk sampel pada penelitian ini berupa citra rambu yang terdiri dari sampel rambu lalu lintas berbentuk diamon.
- Hanya menentukan rambu penyeberangan.
- Sudut pandang pengambilan sampel hanya dari depan.
- Untuk kondisi pengambilan sampel hanya satu kondisi yaitu di siang hari yang cerah.
- Jarak pengambilan sampel hanya pada jarak tertentu yaitu jarak 5 meter.
- Data berupa image dengan format JPG.

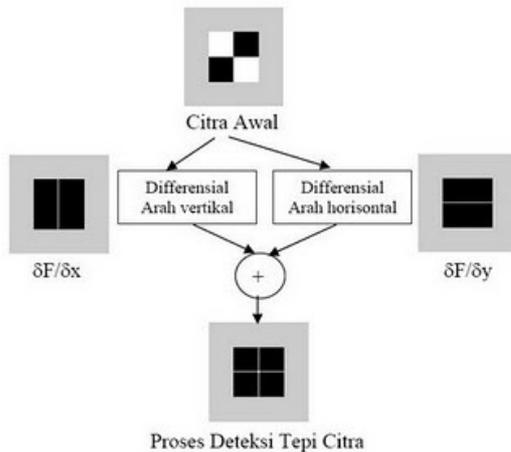
2. DASAR TEORI

2.1 Deteksi Tepi

Deteksi tepi (*Edge Detection*) pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari obyek-obyek citra, tujuannya adalah (Joe, 2009):

- Untuk menandai bagian yang menjadi detail citra
- 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. Gambar1 berikut ini meng-gambarkan bagaimana tepi suatu gambar diperoleh.



Gambar 1 Proses Deteksi Tepi (Yoeda, 2010)

Pada proses deteksi tepi metode canny adalah salah satu algoritma deteksi tepi modern adalah. 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 (Canny, 1986):

a. 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.

b. Melokalisasi dengan baik (kriteria lokalisasi)

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

c. Respon yang jelas (kriteria respon)

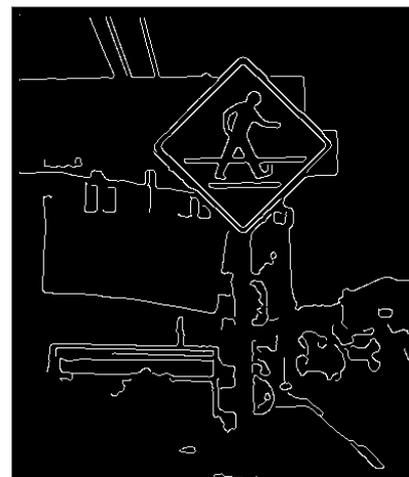
Hanya ada satu respon untuk tiap tepi. Sehingga mudah dideteksi dan tidak menimbulkan kerancuan pada pengolahan citra selanjutnya. Pemilihan parameter deteksi tepi Canny sangat mempengaruhi hasil dari tepian yang dihasilkan. Beberapa parameter tersebut antara lain :

1. Nilai Standart Deviasi Gaussian
2. Nilai Ambang

Langkah-langkah dalam proses deteksi tepi dengan metode canny sebagai berikut:

1. Pertama-tama dilakukan penghalusan (*smoothing*) citra untuk menghilangkan noise. Contohnya menggunakan filtering dengan Gaussian Filter.

2. Selanjutnya dicari *gradient magnitude* citra untuk melihat daerah-daerah yang memiliki turunan spasial yang tinggi.
3. Ditentukan arah dari tepi dengan menggunakan invers tangen dari *gradient magnitude* Y (Gy) dibagi *gradient magnitude* X (Gx). Arah yang diperoleh dari perhitungan ini kemudian dipetakan ke 0, 45, 90, atau 135 derajat berdasarkan kedekatannya dengan keempat derajat arah tadi.
4. Kemudian dilakukan **Non Maximum Suppression**. Yaitu, penghilangan nilai-nilai yang tidak maksimum.
5. Selanjutnya dilakukan **Hysteresis**. *Hysteresis* menggunakan dua *threshold* T1 (*threshold* bawah) dan T2 (*threshold* atas). Bila *magnitude* ada di bawah T1, titik tersebut di-set nol (dijadikan non-tepi). Bila *magnitude* ada di atas T2, maka termasuk tepi. Bila *magnitude* ada diantara T1 dan T2, di-set nol kecuali jika ada jalan (*path*) dari titik tersebut ke titik yang memiliki *magnitude* di atas T2.



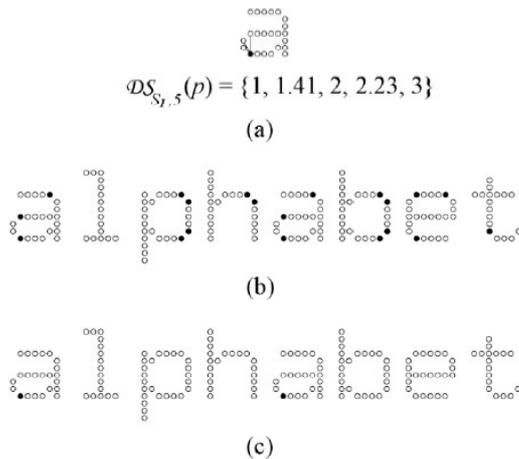
Gambar 2 Hasil Deteksi Tepi dengan T1 = 0.010 dan T2 = 18

2.2 Distance Set

Distance set (Grigorescu, *et al.* 2003) merupakan sebuah algoritma yang menerapkan pemilihan titik-titik penting dari suatu objek (*perceptually significant points*) pada image yang biasanya juga disebut dengan titik fitur. Hubungan titik fitur yang diberikan kepada titik-titik fitur lain dapat dicirikan oleh serangkaian jarak ke titik-titik lainnya. Untuk satu titik tertentu dapat memilih hanya sebagian dari poin atau titik fitur tetangga terdekat. Biasanya jga disebut dengan deskriptor lokal.

$$DS_{s,n}(P) = \{d_1(p), d_2(p), d_3(p), \dots, d_n(p_n)\} \quad (1)$$

dimana $DS_{S,n}$ adalah himpunan n- jarak terdekat dari himpunan titik fitur S . $d_n(p)$ adalah jarak Ecludiean. Ilustrasi proses pengenalan pola dengan prinsip *distance set* dapat dilihat pada Gambar 3.



Gambar 3 (a) Sebuah titik $p \in S_1$ (titik hitam) dari karakter “a” dengan aturan $DS_{S_1, N_1}(P)$ yang memiliki jarak pertama $N_1 = 5$ dengan titik terdekatnya. (b) Penerapan aturan $DS_{S_1, N_1}(P)$ dengan nilai p yang sama dengan nilai p pada karakter “a” mendeteksi adanya kesamaan jarak dengan karakter lain. (c) dengan menaikkan jumlah N yang di hitung akan mengidentifikasi bahwa dengan jarak $DS_{S_1, 15}$; $DS_{S_2, 20}$ ($p,q = 0$ itu merupakan ciri dari karakter “a”).

3. HASIL DAN PEMBAHASAN

3.1 Model Sistem

Gambaran umum perangkat lunak untuk pengenalan bentuk pola rambu penyeberangan dapat dilihat pada Gambar 4.



Gambar 4 Gambaran Umum Sistem

3.2 Analisis Kebutuhan Masukan

Kebutuhan masukan dari user hanya berupa data citra yang akan di uji. Data masukan yang diharapkan dengan format JPG.

3.3 Analisis Kebutuhan Proses

Beberapa proses yang dibutuhkan untuk mengolah input menjadi output berupa informasi yang diharapkan. Beberapa proses tersebut antarlain:

1. Proses input data awal.
2. Proses deteksi tepi dengan menerapkan filter canny.
3. Proses perhitungan jarak sebuah titik pixel tertentu dengan beberapa titik pixel yang berdekatan.
4. Proses menghilangkan nilai pixel yang tidak sesuai dengan pola yang diharapkan.

Dari beberapa proses tersebut diharapkan aplikasi ini dapat mengenali posisi dan bentuk (*diamond*) dari rambu penyeberangan.

3.4 Analisis Kebutuhan Antar Muka

Kebutuhan antar muka yang diperlukan antara lain:

1. Antar muka file chooser, untuk proses memilih data yang akan di masukkan
2. Antar muka halaman utama, dimana terdapat frame untuk menampilkan data citra asli, citra yang telah di deteksi tepi, dan citra hasil.
3. Antar muka about, untuk menjelaskan tentang data programmer.
4. Antar muka help, untuk menjelaskan penggunaan perangkat lunak pengenalan rambu penyeberangan.

3.5 Perancangan Sistem

Pada proses perancangan diawali dengan merancang antar muka perangkat lunak. Rancangan antar muka yang dikembangkan sangat *user friendly*. Perangkat lunak untuk pengenalan rambu penyeberangan ini sudah berbasis *GUI* dimana terdapat *button* dengan *icon* yang mempermudah user dalam menggunakannya. Pemilihan bahasa pemrograman Java karena memiliki kemudahan dalam pemakaian, kelengkapan fungsi, stabil, aman dan *cross-platform* serta masih banyak lagi kelebihan dari Java.

Rancangan antar muka pada perangkat lunak pengenalan rambu penyeberangan antara lain:

1. Antar muka halaman *Chooser*.
2. Antar muka halama utama..
3. Antar muka halaman *About*.
4. Antar muka halaman *Help*.

3.6 Kinerja Yang Diharapkan

Dari hasil analisis sebelumnya diharapkan perangkat lunak ini dapat mengenali bentuk pola dari rambu penyeberangan dengan tepat.

4. IMPLEMENTASI

Pada implementasi perangkat lunak pengenalan rambu penyeberangan akan dijelaskan setiap tahap dari proses dengan menampilkan form form yang dibuat. Gambar 5 menunjukkan form yang menampilkan hasil proses deteksi tepi.

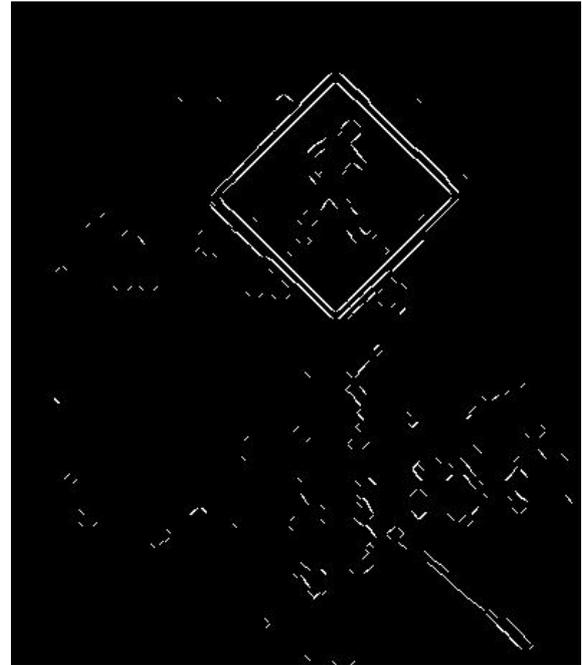


Gambar 5 Proses deteksi tepi

Dari hasil deteksi tepi bisa didapatkan indeks dari pixel yang berwarna putih. Sebelum mendapatkan himpunan S yang merupakan indeks piksel berbentuk rambu lalu-lintas dalam proses perhitungan *distance set*, diperlukan tahap pengurangan piksel yang bukan anggota rambu berbentuk diamond. Proses ini dilakukan dengan menerapkan operasi morfologi dengan stuktur elemen $se_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ dan $se_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$, kedua

operasi ini dilakukan secara independen pada citra hasil deteksi tepi untuk mendapatkan BW_1 dan BW_2 secara berturut-turut. Kemudian kedua hasil digabungkan dengan operasi $BW = BW_1 \cup BW_2$. Hasil operasi morfologi (BW) dapat dilihat pada Gambar 6. Dari gambar 6 tersebut, nampak bahwa garis horisontal (sudut 0°) maupun vertikal (sudut 90°) terhilangkan.

Untuk menghitung *distance set*, diperlukan penentuan himpunan S yang mewakili objek yang akan dikenali. Himpunan S ini diperoleh dari pengumpulan piksel berwarna putih pada BW . Dari contoh pada Gambar 5, diketahui banyaknya piksel adalah anggota S sangatlah jauh berkurang dibandingkan dengan citra hasil deteksi tepi pada Gambar 6.



Gambar 6 Gabungan hasil operasi morfologi dengan se_1 dan se_2 .

Rambu lalu lintas penyeberangan berbentuk belah ketupat, dapat dikenali dari ciri garis luar bentuknya yang berkemiringan sekitar 45° atau 135° . Karakter ini memungkinkan digunakannya transformasi Hough (Hough, 1962) untuk mengenali rambu tersebut.

5. PENGUJIAN

Pengujian pada perangkat lunak pengenalan rambu penyeberangan ini meliputi pengujian terhadap proses pengenalan terhadap pola dari rambu penyeberangan dan juga akan dilakukan pengujian terhadap pola rambu lain dengan bentuk yang sama (*diamond*). Hasil pengujian dapat dilihat dalam Gambar 7. Garis merah yang ditimpakan pada citra rambu diperoleh dari hasil transformasi Hough. Dari Gambar 7 (b) dan (d), nampak bahwa sistem yang dikembangkan mengenali rambu berbentuk belah ketupat dengan tepat. Namun, ada beberapa outliers yang ikut dikenali (bagian dari atas) seperti nampak pada Gambar 7 (b).

Sistem yang dikembangkan belum spesifik membedakan rambu peringatan hati-hati dan rambu penyeberangan. Hal ini akan menjadi fokus pada penelitian berikutnya.

6. KESIMPULAN

Dari hasil penelitian dan pembahasan yang telah dilakukan, dapat diambil kesimpulan bahwa perangkat lunak pengenalan rambu penyeberangan dapat mengetahui titik koordinat dari pola rambu penyeberangan dengan melakukan proses deteksi tepi dan perhitungan jarak antara satu titik pixel dengan beberapa titik tetangganya.



(a)



(b) Hasil pengenalan rambu pada (a)



(c)



(d) Hasil pengenalan rambu pada (c)

Gambar 7. Hasil deteksi rambu berbentuk belah ketupat.

Garis merah menunjukkan hasil perhitungan sistem dengan menggunakan transformasi Hough.

7. DAFTAR PUSTAKA

- Grigorescu, C., dan Petkov, N. “*Distance Sets for Shape Filter and Shape Recognition*”, IEEE Image Processing., vol 12, No.10, 2003.
- Hough, PVC. Method and means for recognizing complex pattern, U.S. Patent 3069654, 1962.
- Canny, John “*A Computational Approach to Edge Detection*”, IEEE Transaction On Pattern Analysis and Machine Intelligence., vol PAMI -8 No 6, November 1986.
- O. Carmichael and M. Hebert. “*Shape-based recognition of wiry objects. In CVPR*”, pp. 401–408, 2003.
- Wilhelm Burger., and Mark J. Burge. “*Digital Image Processing: An Algorithmic Approach Using Java*”. Springer, 2008.
- Yoeda. (2010). “*Deteksi Tepi Menggunakan Metode Prewitt Dan Canny Dengan Matlab*”. Diakses pada 1 mei 2011 dari <http://yoedha-parasites.blogspot.com/2010/11/deteksi-tepi-menggunakan-metode-prewitt.html>
- Joe. (2009). “*Deteksi Tepi Suatu Citra atau Gambar*”. Diakses pada 5 mei 2011 dari <http://joyhomework.wordpress.com/2009/12/04/deteksi-tepi-suatu-citra-atau-gambar/>