

Aplikasi Pendeteksi Rambu Lalu-Lintas Menggunakan Operator Sobel dan Metode Hamming

Ericks Rachmat Swedia¹, M. Ridwan Dwi Septian²

Fakultas Teknik Informatika
Universitas Gunadarma
Jakarta, Indonesia

¹ericks_rs@staff.gunadarma.ac.id,

²ridwandwiseptian@staff.gunadarma.ac.id

Margi Cahyanti

Fakultas Sistem Informasi
Universitas Gunadarma
Jakarta, Indonesia

margi@staff.gunadarma.ac.id

Abstrak— Penelitian ini membuat aplikasi untuk mendeteksi rambu lalu-lintas berdasarkan bentuk dengan menggunakan operator sobel dan mencari kemiripan citra menggunakan metode hamming distance. Pengguna dapat memasukkan citra yang akan dicocokkan dengan citra yang terdapat dalam database. Citra yang dipilih oleh pengguna akan ditransformasi menjadi gray dan dilakukan proses deteksi tepi sobel. Citra ini kemudian dicocokkan dengan citra yang terdapat dalam database dengan metode hamming distance, nilai hamming yang paling kecil merupakan citra yang paling mirip. Untuk database digunakan 205 citra berukuran 100x100 piksel yang dapat diunduh pada halaman https://id.wikipedia.org/wiki/Rambu_lalu_lintas. Citra-citra ini dijadikan basis data setelah melalui proses grayscale dan deteksi tepi. Proses uji coba dilakukan dengan mengambil 10 citra rambu yang dipilih secara acak dari pencarian google sebagai proses input untuk dicari kemiripan dengan citra yang terdapat dalam database. Berdasarkan hasil uji coba, aplikasi ini dapat digunakan untuk mendeteksi rambu lalu-lintas.

Kata Kunci—citra; sobel; hamming; rambu; lalu-lintas.

I. PENDAHULUAN

Tanda lalu-lintas adalah fitur penting dalam memberikan informasi keselamatan untuk pengemudi tentang kondisi jalan. Dengan menggambarkan situasi lalu-lintas saat ini, menentukan arah mengemudi, memberikan peringatan tentang potensi risiko, dan mengizinkan atau melarang akses jalan agar pengemudi mengetahui situasi lalu-lintas sekitarnya dengan melihat tanda-tanda yang ada di depan atau yang baru saja dilewati pengemudi. Pengenalan tanda lalu-lintas dapat mengurangi beban pengemudi dalam mengingat rambu dan meningkatkan keselamatan.

Keselamatan menjadi hal utama bagi pengguna lalu-lintas untuk ketertiban dan keamanan saat berkendara, rambu lalu-lintas menjadi hal yang wajib dipatuhi oleh semua pengguna jalan raya. Kenyataannya, saat berkendara ada banyak rambu lalu-lintas yang tidak dimengerti oleh pengguna jalan raya sehingga mengakibatkan pelanggaran lalu-lintas dan terjadinya kecelakaan. Hal ini tentu akan merugikan orang lain atau pengguna jalan raya dengan meningkatkan resiko kecelakaan [12].

Selama ini, panduan berkendara yang populer digunakan masih memanfaatkan sinkronisasi sinyal satelit yang biasanya disebut dengan perangkat GPS (*Global Position System*). Dapat

dilihat bahwa, sistem tersebut belum dilengkapi panduan yang lengkap seperti halnya pendeteksi tanda jalan atau 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* [7].

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 [1]. 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 lalu-lintas diterapkan teknik distance set. Teknik ini bertujuan untuk melakukan pengenalan terhadap bentuk dari objek yang ingin dideteksi. Pengenalan ini disebut juga dengan *Shape Recognition* [2].

Salah satu permasalahan yang muncul adalah sulitnya mengetahui macam-macam rambu lalu-lintas. Berdasarkan masalah diatas, penelitian mencoba membuat aplikasi agar dapat mengenali tanda-tanda rambu berdasarkan bentuk dengan menggunakan operator deteksi tepi sobel dan mencari kemiripan menggunakan metode hamming distance yang diimplementasikan dengan menggunakan bahasa pemrograman C#.

II. PEMBAHASAN

A. Rambu Lalu-Lintas

Rambu lalu-lintas adalah bagian dari perlengkapan jalan yang memuat lambang, huruf, angka, kalimat dan/atau perpaduan di antaranya, yang digunakan untuk memberikan peringatan, larangan, perintah dan petunjuk bagi pemakai jalan. Rambu lalu-lintas diatur menurut Peraturan Menteri Perhubungan Nomor 13 tahun 2014 [11]. Pengelompokan rambu diantaranya rambu lalu-lintas, rambu larangan, rambu perintah, dan rambu petunjuk. Citra rambu lalu-lintas yang digunakan dalam aplikasi ini dapat diunduh pada halaman https://id.wikipedia.org/wiki/Rambu_lalu_lintas



Gambar 1. Contoh Citra Rambu Lalu-Lintas

B. Grayscale

Citra grayscale adalah suatu citra yang terdiri dari warna hitam sebagai warna minimum dan warna putih sebagai warna maksimumnya, sehingga warna antara ke dua warna, (minimum dan maksimum) tersebut adalah warna abu-abu. Untuk mengubah citra menjadi citra grayscale dapat dilakukan dengan menggunakan rumus berikut [10] :

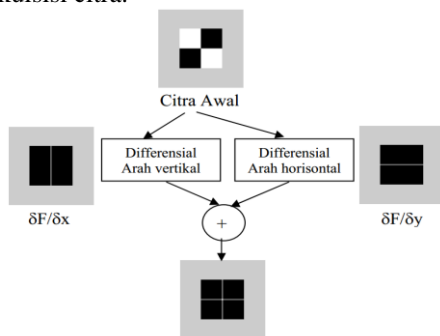
$$I = 0.333R + 0.5G + 0.1666B \quad (1)$$

Dimana, R = Nilai piksel Red
 G = Nilai piksel Green
 B = Nilai piksel Blue

C. Deteksi Tepi

Deteksi tepi (*edge detection*) pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari obyek-obyek citra, tujuannya adalah [3]:

- 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.



Gambar 2. Proses Deteksi Tepi Citra [4]

Suatu titik (x,y) dikatakan sebagai tepi (*edge*) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangganya. Gambar 2 menggambarkan bagaimana tepi suatu gambar diperoleh.

Sobel merupakan salah satu pengembangan dari teknik *edge detection* sebelumnya (Metode Robert) dengan menggunakan HPF (*High Pass Filter*) yang diberi satu angka nol penyangga, juga pengembangan dari operator Prewit. [8] Algoritma ini

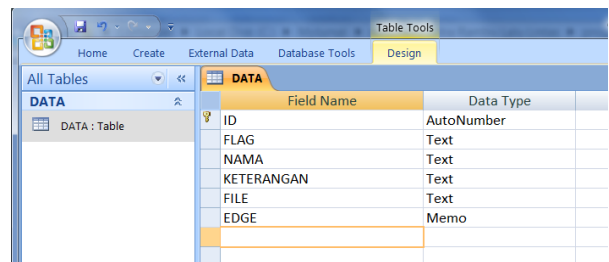
termasuk algoritma pemrograman yang berfungsi sebagai filter image. Filter ini mendeteksi keseluruhan edge yang ada. Dalam prosesnya filter ini menggunakan sebuah operator, yang dinamakan operator sobel. Operator sobel menggunakan matriks NxN dengan berordo 3 x 3, 5 x 5, 7 x 7, dan sebagainya. Matriks seperti ini digunakan untuk mendapatkan piksel tengah sehingga menjadi titik tengah matrik (a_{ij}). Piksel tengah ini merupakan piksel yang akan diperiksa. Cara pemanfaatan matrik ini sama seperti pemakaian sebuah grid, yaitu dengan cara memasukkan piksel-piksel disekitar yang sedang diperiksa (piksel tengah) ke dalam matrik [5].

D. Hamming Distance

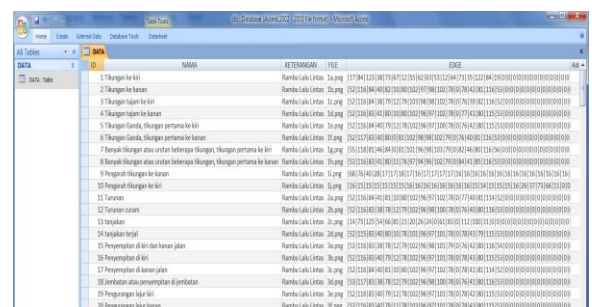
Hamming distance adalah cara untuk mengukur jarak antara dua string yang ukurannya sama dengan membandingkan simbol-simbol yang terdapat pada kedua string pada posisi yang sama. Metode hamming distance merupakan salah satu dari algoritma approximate string matching yang ditemukan oleh Richard Hamming, pada tahun 1950. Metode hamming distance pertama kali digunakan untuk mendeteksi dan memperbaiki telekomunikasi sebagai estimasi error [6]. Hamming distance dari dua string adalah jumlah simbol dari kedua string yang berbeda. Sebagai contoh Hamming Distance antara string "karolin" dan "kerstin" adalah 3. Hamming distance juga dapat digunakan untuk mengukur jarak antar dua string binary misalnya jarak antara "1011101" dengan "1001001" adalah 2.

III. PERANCANGAN APLIKASI

Langkah pertama dalam membuat aplikasi yang menampilkan hasil dari pencarian kemiripan, adalah dengan membuat database citra rambu lalu-lintas dengan ukuran 100x100 piksel. Database citra dibuat dengan menggunakan Microsoft Access berformat mdb, dengan design dapat dilihat pada gambar 3a. Nilai deteksi tepi dari citra rambu lalu-lintas berukuran 100x100 piksel disimpan dalam field EDGE, dengan cara mengubah data citra 2 dimensi (baris dan kolom) menjadi satu baris seperti terlihat dalam gambar 3b [9].

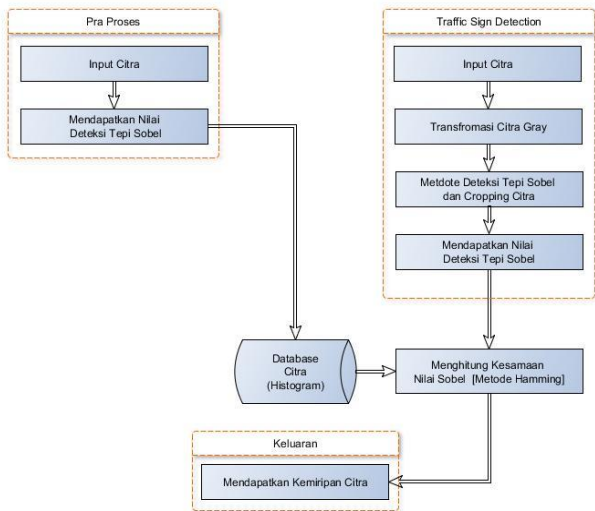


Gambar 3a. Database Citra Rambu Lalu-lintas



Gambar 3b. Nilai Citra yang disimpan dalam database

Alur program *Traffic Sign Detection* ini disajikan pada Gambar 4 di bawah ini.



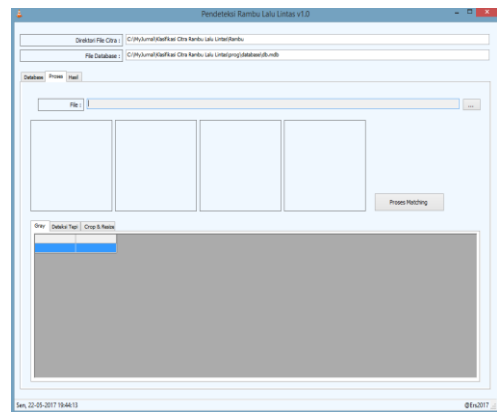
Gambar 4. Alur Program

Keterangan Gambar 4 :

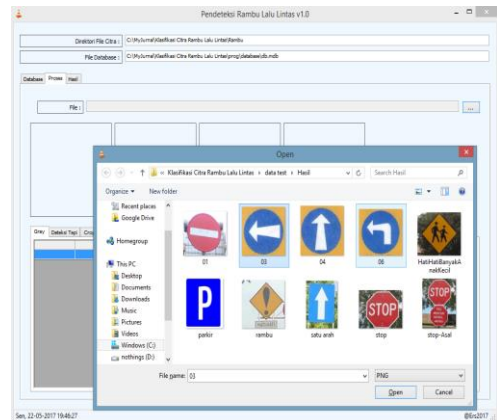
- **Pra Proses**
 - Diawali dengan menerima input citra berupa gambar yang telah ditentukan sebelumnya.
 - Gambar yang telah diinput selanjutnya akan di proses untuk mendapatkan nilai deteksi tepi sobel. Nilai ini kemudian akan disimpan dalam database.
- **Traffic Sign Detection**
 - Diawali dengan menerima input citra berupa gambar yang telah dipilih sebelumnya.
 - Citra ditransformasi ke dalam bentuk grayscale.
 - Setelah citra ditransformasi dalam bentuk grayscale, dilakukan pendeteksian tepi citra, cropping dan zooming citra agar berukuran 100x100 piksel.
 - Selanjutnya nilai dari pendeteksian tepi tersebut disimpan ke dalam tabel di aplikasi.
 - Citra yang diinput tersebut diproses untuk menghitung kemiripan pada citra yang terdapat di dalam database. Menghitung kemiripan menggunakan metode hamming distance berdasarkan string binary. Nilai piksel hasil deteksi tepi sobel diatas nilai 0 dianggap bernilai 1.
 - Nilai hamming distance yang terkecil merupakan citra yang mirip, aplikasi akan mengurutkan nilai hamming distance dari yang terkecil dan menampilkan kemiripan tersebut beserta dengan keterangan rambu.

IV. UJI COBA DAN PEMBAHASAN

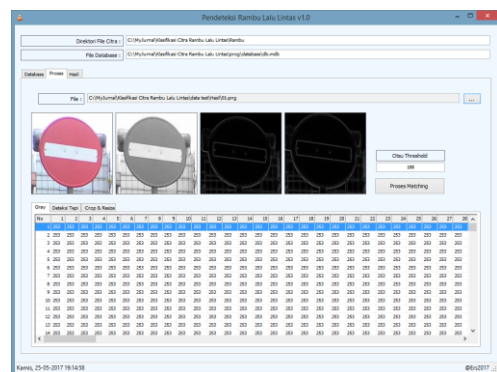
Tahap ini adalah tahapan terakhir untuk memastikan bahwa aplikasi yang telah dibuat dapat dioperasikan dengan baik. Pada pencarian kemiripan ini, pengguna dapat memilih citra untuk melakukan proses kemiripan citra.



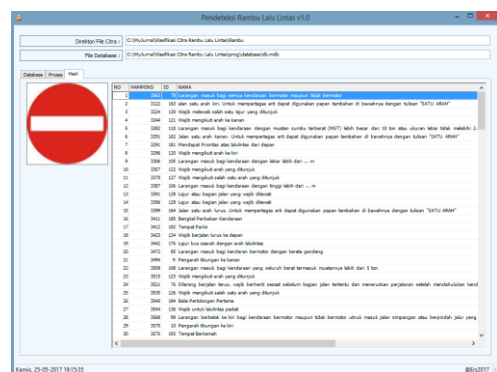
Gambar 5. Interface Aplikasi



Gambar 6. Proses Pemilihan Citra

















Gambar 7. Proses Pengolahan Citra



Gambar 8. Hasil

Pada tahap uji coba, program menggunakan 10 citra rambu yang diambil secara acak dari pencarian google sebagai proses input untuk dicocokkan ke citra yang terdapat dalam database. Hasil uji coba aplikasi dapat dilihat pada tabel berikut:

TABEL I. Hasil Uji Coba Aplikasi

No	Citra Input	Output		
		Citra Output	Nama Rambu	Hamming
1			Larangan masuk bagi semua kendaraan bermotor maupun tidak bermotor	3062
2			Wajib mengikuti arah ke kiri	2013
3			Wajib berjalan lurus ke depan	2585
4			Wajib mengikuti arah yang ditunjuk	1550
5			Banyak anak-anak	3720
6			Tempat Parkir	1329
7			Hati-hati	3666
8			Jalan satu arah lurus. Untuk mempertegas arti dapat digunakan papan tambahan di bawahnya dengan tulisan "SATU ARAH"	2976
9			Dilarang berjalan terus, wajib berhenti sesaat dan meneruskan perjalanan setelah mendapat kepastian aman dari lalu-lintas arah lainnya	4093
10			Banyak tikungan atau urutan beberapa tikungan, tikungan pertama ke kiri	4316

V. KESIMPULAN

Berdasarkan pada analisa hasil pengujian aplikasi pendeteksi rambu lalu-lintas, dapat diambil kesimpulan bahwa dengan metode hamming dan operator sobel dapat digunakan untuk mendeteksi bentuk rambu lalu-lintas.

Uji coba aplikasi ini menggunakan 10 data citra rambu yang diambil secara acak dari pencarian google. Dengan adanya aplikasi ini diharapkan dapat mempermudah user dalam mengetahui fungsi dari rambu-rambu yang terdapat di jalan raya.

DAFTAR PUSTAKA

- [1] Wilhelm Burger., and Mark J. Burge. "Digital Image Processing: An Algorithmic Approach Using Java". Springer, 2008.
- [2] O. Carmichael and M. Hebert. "Shape-based recognition of wiry objects. In CVPR", pp. 401–408, 2003.
- [3] Joe. (2009). "Deteksi Tepi Suatu Citra atau Gambar". Diakses pada 21 mei 2017 dari <http://joyhomework.wordpress.com/2009/12/04/deteksi-tepi-suatu-citra-atau-gambar/>.
- [4] Yoeda. (2010). "Deteksi Tepi Menggunakan Metode Prewitt Dan Canny Dengan Matlab". Diakses pada 1 mei 2011 dari <http://yoedhaparasites.blogspot.com/2010/11/deteksi-tepimenggunakan-metode-rewitt.html>.
- [5] Sobel, Irwin. 2014. History and Definition of the Sobel Operator.
- [6] Hamming, Richard W. (1950). "Error detecting and error correcting codes" (PDF). Bell System Technical Journal. 29 (2): 147–160. doi:10.1002/j.1538-7305.1950.tb00463.x. MR 0035935.
- [7] Pratama, A. dan Muhimmah I. (2011). "Aplikasi Pengenalan Rambu Berbentuk Belah Ketupat". Seminar Nasional Aplikasi Teknologi Informasi 2011 (SNATI 2011). Yogyakarta, 17-18 Juni 2011. ISSN: 1907-5022.
- [8] Yunus, M. Perbandingan Metode Edge Detection Untuk Proses Segmentasi Citra Digital. Jurnal Teknologi Informasi, 3(2),146–160.
- [9] Swedia, Ericks Rachmat., dan Cahyanti, Margi. 2015. Algoritma Transformasi Ruang Warna. Depok: Indie Publishing.
- [10] Finlayson, G. D., Qiu, G., Qiu, M., 1999, Contrast Maximizing and Brightness Preserving Color to Grayscale Image Conversion.
- [11] https://id.wikipedia.org/wiki/Rambu_lalu_lintas, Diakses pada 1 Mei 2017.
- [12] Rr. Octanty M., Dzuratul U., Sayyidina Auliya., Anastasia C.U., M. Hilman Fatah., Shelani M., "Deteksi Gambar Rambu Lalu Lintas Dengan Algoritma Jaringan Syaraf Tiruan", Program Studi Teknik Informatika, Universitas Brawijaya, Diakses pada 1 Mei 2017 dari <http://yudistira.lecture.ub.ac.id/files/2014/04/DETEKSI-GAMBAR-RAMBU-LALU-LINTAS-DENGAN-ALGORITMA-JARINGAN-SYARAF-TIRUAN-.pdf>.