

ANALISIS DAN PERANCANGAN SOFTWARE UNTUK MENENTUKAN WARNA KENDARAAN GELAP DAN TERANG

Haruno Sajati, Yenni Astuti

Jurusan Teknik Informatika STT Adisutjipto Yogyakarta
Jl. Janti Blok R Yogyakarta
informatika@stta.ac.id

Abstrak

Kemacetan parah yang terjadi di jalan raya memaksa pemerintah daerah memerlukan strategi khusus untuk mengatasinya, salah satunya pembatasan kendaraan bermotor berdasarkan warna. Strategi ini rentan permasalahan dan sangat merepotkan untuk petugas jalan raya untuk menerapkannya. Cara yang paling mudah adalah membuat sebuah sistem otomatisasi yang dapat dengan mudah menentukan sebuah kendaraan berwarna gelap atau terang. Tujuan penelitian ini adalah membuat sebuah aplikasi untuk menentukan warna kendaraan gelap dan terang secara mudah.

Metode deteksi tepi pada pengenalan pola digital dapat digunakan untuk memperoleh informasi obyek dalam sebuah citra. Pada penelitian ini, metode deteksi tepi sobel digunakan untuk memperoleh informasi mengenai jenis kendaraan kemudian mengolah warna obyek tersebut. Permasalahan yang muncul pada metode deteksi tepi adalah kualitas citra yang dipergunakan sebagai masukan seperti kualitas cahaya, warna obyek terhadap latar belakang atau *noise* pada obyek tersebut.

Program yang dihasilkan dalam penelitian ini dapat membedakan warna kendaraan dengan persentase kebenaran pendeteksian tepi 76.57 dan persentase kebenaran pendeteksian warna 74.19.

Kata kunci: Citra digital, Deteksi Tepi, Pengenalan Pola, Metode Sobel

Abstract

An enormous traffic jam forces district government to solve it with a special strategy - by limiting vehicle based on its color. This strategy is difficult to implement especially for the officer. The easiest way to resolve it is to make an automatic system which can separate a vehicle by its color, dark or light. Border Detection Method for digital pattern recognition can be used to obtain object information from an image.

In this research, Sobel Border Detection Method is applied to acquire information about the vehicle type, which then processes its color. The problem that occurs on border detection method is the images qualities that are used, such as color quality, object color against its background, and noise from the object.

The application that made from this research can distinguish the vehicle color with the truth border detection value of 76.57 percent and the truth color detection value of 74.19 percent.

Keywords: Digital Image, Edge Detection, Pattern Recognition, Sobel Method

1. PENDAHULUAN

Peningkatan volume kendaraan bermotor menyebabkan kemacetan parah di beberapa ruas jalan protokol tidak hanya di ibukota tetapi di daerah-daerah berkembang lainnya. Untuk mengurangi kemacetan ini, pemerintah daerah memiliki beberapa solusi antara lain peningkatan minat pengguna jalan untuk menggunakan angkutan umum, pemberlakuan aturan *three in one*, pemberlakuan aturan plat nomor ganjil dan genap maupun aturan warna kendaraan gelap dan terang. Peraturan pembatasan kendaraan gelap dan terang ini juga dapat menjadi perdebatan yang cukup panjang bagi pengguna jalan dengan petugas karena sangat sulit membedakan warna gelap dan terang dengan mata telanjang dan dapat menjadikan salah penafsiran warna.

Hipotesis

Dugaan yang akan dibuktikan pada penelitian ini adalah:

- a. Pembedaan warna gelap dan terang dapat dilakukan dengan mengambil sample titik secara acak di beberapa tempat pada kendaraan, merubah warna sample tersebut menjadi nilai RGB kemudian merata-ratakan nilai tersebut.
- b. Pengambilan gambar harus menggunakan kamera dengan resolusi tinggi sehingga gambar yang diperoleh dapat akurat sehingga dapat diperoleh masukan yang jelas akhirnya dapat diolah dengan cepat dan benar.

Pendekatan penyelesaian masalah adalah dengan menganalisa sebuah citra, menentukan batas kendaraan, menghitung luminasi warna dan menghasilkan output warna kendaraan gelap dan terang. Hasil yang diharapkan adalah dapat diperoleh hasil pengolahan citra berupa warna kendaraan yang tepat.

2. METODE PENELITIAN

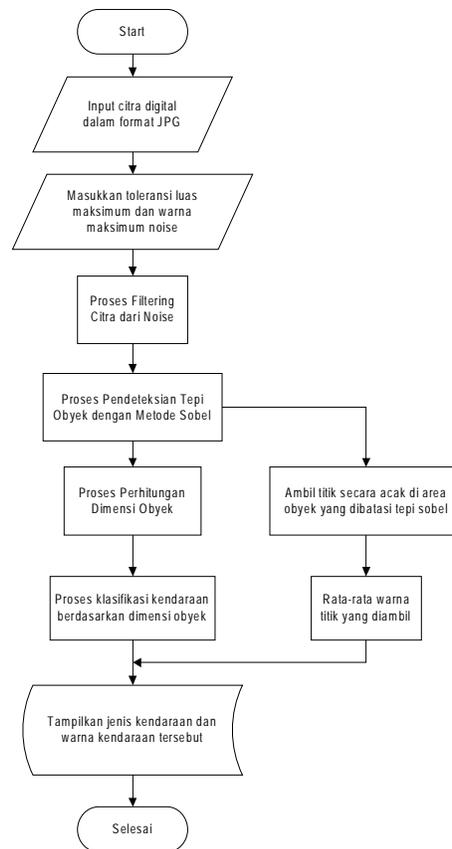
2.1 Tahapan pembuatan perangkat lunak

Tahapan pembuatan perangkat lunak pengenalan pola kendaraan adalah sebagai berikut:

- a) Mempelajari metode yang digunakan dari jurnal maupun sumber-sumber ilmiah yang ada.
- b) Merancang perangkat lunak dengan menggunakan metode yang telah dipelajari.
- c) Membuat perangkat lunak berdasarkan analisis dan perancangan yang dilakukan.
- d) Uji coba dan analisa perangkat lunak yang telah dibuat

2.2 Alur Pengolahan Citra

Alur pengolahan citra sebelum akhirnya dideteksi dan dikenali ditunjukkan pada Gambar 2.1 sebagai berikut:



Gambar 2.1 Alur Pendeteksian Obyek

2.3. Pengacuan Pustaka

2.3.1 Pengenalan Pola

Setelah diperoleh data tepi dari sebuah obyek dalam sebuah citra, langkah berikutnya mengenali obyek tersebut. Proses pengenalan obyek inilah yang disebut sebagai pengenalan pola (*pattern recognition*). Pengenalan pola dapat dilakukan dengan memperhatikan ciri tertentu dari sebuah obyek, yang ditunjukkan pada tabel 2.1 berikut [5].

Terdapat beberapa pendekatan yang dapat dilakukan untuk mengenali sebuah pola yaitu:

- a. Pengenalan Pola Statistikal
- b. Pengenalan Pola Sintaktik
- c. Pengenalan Pola Neural

Penelitian ini menggunakan pendekatan pengenalan pola statistikal yaitu sekelompok karakteristik pengukuran yang menunjukkan ciri diekstraksi dari data *input* dan digunakan untuk menentukan setiap vektor fitur ke dalam suatu kelas. Ciri (*feature*) diasumsikan dihasilkan secara natural, sehingga model yang bersangkutan merupakan kelas-kelas probabilitas atau fungsi kepadatan probabilitas (*Probability Density Function*) yang telah dikondisikan. Pola kendaraan ditentukan dengan menghitung dimensi (panjang dan lebar) dari citra kendaraan yang diolah kemudian ditentukan jenis kendaraan dari luas citra tersebut.

3. HASIL DAN PEMBAHASAN

3.1 Perancangan Program

Setiap piksel dalam sebuah citra digital merepresentasikan nilai RGB (*Red Green Blue*). Fungsi mencari nilai RGB pada sebuah piksel citra adalah sebagai berikut:

```
$source_image = "hasil.jpg";
function rgb_value($source_image) {
    $starting_img = imagecreatefromjpeg($source_image);
    $rgb = imagecolorat($starting_img, $x, $y);
    $colors = imagecolorsforindex($im, $rgb);
    return $colors;
}
```

Fungsi pada gambar 4.1 akan mengembalikan nilai RGB yang tersimpan dalam array \$colors. Proses parsing nilai tersebut adalah sebagai berikut:

```
$merah=$colors['red'];
$hijau=$colors['green'];
$biru=$colors['blue'];
```

Luminance (luminansi) merupakan karakteristik yang menyatakan tingkat pencahayaan pada sebuah benda atau obyek. Tingkat kecerahan ini menunjukkan seberapa terang atau gelap warna sebuah pixel di sebuah citra digital. Nilai luminansi (Y) sebuah piksel ditentukan pada formula sebagai berikut (Sazzal, 2013):

$$Y = 0.2126R + 0.7152G + 0.0722B$$

Dimana:

- Y: Luminansi warna sebuah pixel
- R: Komposisi nilai warna merah dalam desimal
- G: Komposisi nilai warna hijau dalam desimal
- B: Komposisi nilai warna biru dalam desimal

Fungsi untuk mencari nilai iluminansi adalah sebagai berikut:

```
function get_luminance($im, $x, $y) {
    $rgb = imagecolorat($im, $x, $y);
    $colors = imagecolorsforindex($im, $rgb);
    $merah=$colors['red'];
    $hijau=$colors['green'];
    $biru=$colors['blue'];
    $ilum = (0.2126*$merah) + (0.7152*$hijau) + (0.0722*$biru);
    return $ilum;
}
```

Penyaringan citra digunakan untuk membersihkan gambar dari *noise* yang dapat dideteksi sebagai obyek nantinya oleh Metode Sobel yang berikutnya akan membuat kesalahan proses pengenalan pola kendaraan. Pada umumnya aspal memiliki kontur yang tidak teratur mengingat jalan tidak 100% dibuat menggunakan aspal, tetapi campuran batu, kerikil dan semen. Hal ini mengakibatkan beberapa bagian aspal akan dideteksi sebagai sebuah obyek oleh metode deteksi tepi Sobel seperti ditunjukkan pada gambar 3.1 berikut



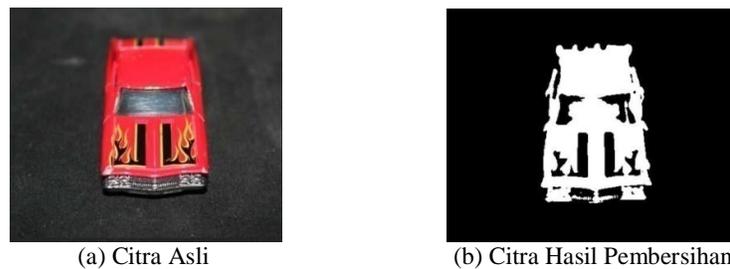
Gambar 3.1 Citra Hasil Deteksi Tepi Sobel

Proses pembersihan citra dari *noise* mirip dengan matriks sobel yaitu dengan membandingkan piksel saat ini dengan piksel-piksel di sekitarnya. Berbeda dengan metode Sobel, penelitian ini melebarkan piksel pembanding yang “d disesuaikan” dengan *noise* aspal yang diuji. Pada pengujian di beberapa tempat, kontur aspal jalan berbeda-beda, ada yang cukup “rapi” dalam artian jalannya halus, ada juga yang konturnya “berantakan”. Proses perbandingan nilai piksel ini ditunjukkan pada matriks gambar 3.2 berikut ini:

[1][1]	[1][2]	[1][3]	[1][4]	[1][5]
[2][1]	[2][2]	[2][3]	[2][4]	[2][5]
[3][1]	[3][2]	[3][3]	[3][4]	[3][5]
[4][1]	[4][2]	[4][3]	[4][4]	[4][5]
[5][1]	[5][2]	[5][3]	[5][4]	[5][5]

Gambar 3.2 Perbandingan Sel Piksel dengan Sel Piksel di Sekitarnya

Pada gambar 3.2 di atas, sel [3][3] akan dibandingkan dengan sel sekitarnya dengan jarak 2. Penentuan jarak ini ditentukan dengan seberapa parah kontur aspal yang diuji. Setiap warna di sekitar sel yang diuji dijumlahkan kemudian dicari rata-rata nilainya. Hasil proses pembersihan ini ditunjukkan pada gambar 4.7 sebagai berikut:

Gambar 3.3 Citra Hasil Pembersihan *Noise*

Setelah diperoleh citra yang bersih pada gambar 4.7 (b), proses berikutnya baru proses deteksi tepi Sobel yang ditunjukkan pada gambar 3.3 berikut:



(a) Citra Hasil Filtering (b) Citra Hasil Deteksi Tepi Sobel
 Gambar 3.4 Citra Hasil Deteksi Sobel

Dengan asumsi noise dari latar belakang sudah tidak ada, maka pendeteksian pojok dapat dilakukan dengan melakukan scanning pixel dari pojok kiri bawah ke pojok kanan atas. Saat ada transisi pertama dari “gelap” ke “terang” dan transisi terakhir dari “gelap” ke “terang” maka diperoleh titik corner dari obyek yang diuji seperti ditunjukkan pada gambar 3.5 (a) sebagai berikut. Setelah pojok diketahui koordinatnya, proses berikutnya adalah membuat bidang persegi panjang antar kedua titik pojok tersebut pada gambar 3.5 (b). Fungsi membuat bidang persegi panjang adalah sebagai berikut.

```
$canvas = imagecreatetruecolor($awal_canvas, $akhir_canvas);
$white = imagecolorallocate($canvas, 255, 255, 255);
imagerectangle($canvas, $x1, $y1, $x2, $y2, $white);
```



(a) Deteksi Pojok (b) Bidang Persegi Panjang
 Gambar 3.5 Proses Pembuatan Bidang Persegi Panjang

3.2 Pengujian Program

Pengujian data dilakukan dengan memberikan masukan gambar kendaraan sebenarnya dengan membandingkan hasil pendeteksian warna program dengan mata telanjang sekaligus menghitung waktu eksekusi berbagai varian jumlah masukan data. Teknologi image processing memiliki waktu eksekusi yang cukup lama karena komputasinya dilakukan per-pixel sehingga panjang waktu eksekusi tergantung pada dimensi gambar yang diolah.

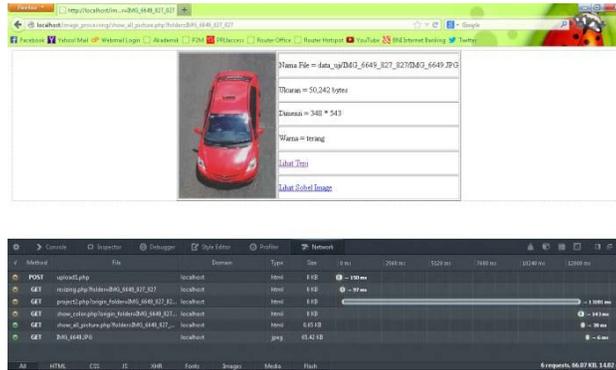
Prosentase kebenaran dihitung menggunakan perbandingan hasil pendeteksian warna program dengan mata telanjang dengan formula:

$$\% \text{ warna valid} = \frac{\sum \text{data warna benar}}{\sum \text{seluruh data}} * 100$$

Prosentase kebenaran pendeteksian tepi dihitung menggunakan perbandingan pendeteksian tepi dengan mata telanjang dengan formula:

$$\% \text{ deteksi tepi} = \frac{\sum \text{data tepi benar}}{\sum \text{seluruh data}} * 100$$

Waktu eksekusi dihitung menggunakan *inspect element* pada browser Mozilla Firefox dengan contoh yang terlihat pada gambar 3.6 sebagai berikut:



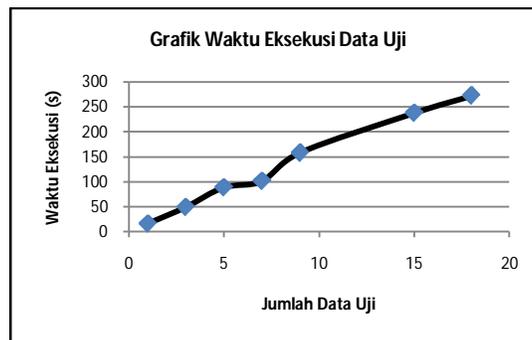
Gambar 3.6 Menghitung Waktu Eksekusi

Hasil pengolahan data uji ditunjukkan pada tabel 3.1 sebagai berikut:

Tabel 3.1 Hasil pengolahan data uji

Jumlah Data	Prosentase pendeteksian tepi (%)	Prosentase kebenaran warna (%)	Waktu eksekusi (s)
1	100	100	16.67
3	83.35	66.67	49.765
5	100	100	89.29
7	64.28	64.28	102.04
9	61.115	61.13	158.87
15	63.335	63.335	238.5
18	63.89	63.89	273.285
21	Tidak terhitung	Tidak terhitung	Tidak terhitung

Grafik waktu eksekusi data uji ditunjukkan pada gambar 3.7 sebagai berikut:



Gambar 3.7 Grafik Waktu Eksekusi Data

Hubungan antara waktu eksekusi dengan jumlah data ditunjukkan pada persamaan

$$y = 15.25 * x - 6.247$$

Tingginya waktu eksekusi diakibatkan karena ukuran citra yang diolah oleh perangkat lunak. Hal ini menjadi wajar mengingat proses komputasi metode Sobel dilakukan per-pixel (*picture element*) sehingga semakin besar dimensi sebuah gambar, maka semakin besar pula pixel yang dimiliki oleh gambar tersebut sehingga proses komputasinya menjadi semakin lambat. Pada tabel percobaan 3.1, saat data uji berjumlah 21, program tidak mampu lagi mengeksekusi data. Hal ini disebabkan web server yang digunakan dalam penelitian ini memiliki batas waktu eksekusi 300 detik.

Pada tabel percobaan 5.1 diperoleh rata-rata kebenaran pendeteksian tepi data dan pendeteksian warna sebesar 76.57 dan 74.19. Semua kesalahan pendeteksian warna diakibatkan karena kesalahan pendeteksian tepi. Kegagalan pendeteksian tepi diakibatkan beberapa faktor:

- Kualitas citra yang kurang baik
- Kualitas pencahayaan yang berubah-ubah
- Gangguan (*noise*) yang muncul saat pengambilan gambar seperti munculnya obyek-obyek lain yang terdeteksi oleh perangkat lunak.

Tabel jumlah kendaraan warna gelap dan terang ditunjukkan pada tabel 5.3 sebagai berikut:

Tabel 3.2 Rekap Warna Kendaraan

Warna Gelap	Warna Terang	Tidak Terdeteksi
11	31	9

Pada tabel 3.1 diketahui seluruh warna yang “Tidak Terdeteksi” adalah warna kendaraan gelap seperti ditunjukkan pada sample percobaan gambar 3.3 sebagai berikut:



Gambar 3.3 Kegagalan pendeteksian warna

Hal ini diakibatkan tipisnya perbedaan antara latar belakang (aspal) dengan warna kendaraan tersebut sehingga mengakibatkan program tidak bisa mendeteksi obyek kendaraan gelap yang terlalu mirip dengan latar belakang. Selain itu juga adanya obyek *noise* yang mengganggu obyek yang dideteksi seperti terlihat pada gambar 3.3. Sebenarnya hal ini bisa dihindari dengan memberikan *mask* obyek *noise* atau memberikan warna latar belakang yang kontras dengan obyek yang dideteksi. Kesalahan pendeteksian warna kendaraan dari sisi perangkat lunak diakibatkan pengambilan warna sample secara random 20 titik di citra kendaraan. Pemilihan yang dilakukan secara acak ini dilakukan untuk mengurangi beban komputasi yang cukup tinggi apabila dihitung secara keseluruhan warna kendaraan.

4. KESIMPULAN

Dari analisa program dan pengujian data uji diperoleh kesimpulan sebagai berikut:

- a. Program pendeteksian tepi obyek menggunakan metode Sobel berbasis web telah dapat digunakan dengan nilai validitas 76.57%.
- b. Program pendeteksian warna kendaraan gelap dan terang telah dapat menentukan warna dengan nilai validitas 74.19%.
- c. Waktu yang digunakan untuk proses pendeteksian tepi dan pembedaan warna kendaraan adalah $lama\ eksekusi = 15.25 * jumlah\ data - 6.247$.
- d. Gangguan utama yang terjadi pada kesalahan pembacaan data uji adalah munculnya noise yang berupa obyek-obyek lain dalam data yang diuji.
- e. Gangguan lain yang menjadi penyebab kegagalan pembacaan data uji adalah cahaya yang berubah-ubah.

5. SARAN

- a. Pada penelitian berikutnya diharapkan dapat meningkatkan nilai validitas pendeteksian tepi dan warna.
- b. Diharapkan terdapat mekanisme pendeteksian multi obyek sehingga dalam satu data uji, bisa diperoleh lebih dari satu obyek kendaraan sekaligus

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Direktorat Jendral Perguruan Tinggi yang telah memberi dukungan financial terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] Sutoyo, T. *et al.* (2009). *Teori Pengolahan Citra Digital*. Yogyakarta: Andi.
- [2] Nadernejad E., Sharifzadeh S., Hassanpour H., Edge Detection Techniques: Evaluations and Comparisons, *Applied Mathematical Sciences*, Vol. 2, 2008, no. 31, 1507 – 1520
- [3] Riyanto Sigit, Modul Praktikum Pengolahan Citra, Politeknik Elektronika Negeri Surabaya
- [4] Murdianto A, Ekstraksi Fitur Wajah dengan Metode Gabor Wavelet, Fasilkom UI, 2007
- [5] Munir Rinaldi, E-Book Pengolahan Citra Digital, <http://informatika.stei.itb.ac.id/~rinaldi.munir/Buku/Pengolahan%20Citra%20Digital>
- [6] T. M. Shahriar Sazzad, Sabrin Islam, Mohammad Mahbubur Rahman Khan Mamun, Md. Zahid Hasan, Establishment of an Efficient Color Model from Existing Models for Better Gamma Encoding In Image Processing, *International Journal of Image Processing (IJIP)*, Volume (7): Issue (1), 2013.

