

# PERANCANGAN DAN IMPLEMENTASI APLIKASI KAMERA DETEKSI KEMACETAN JALAN RAYA DENGAN CITRA PANORAMA

## DESIGN AND IMPLEMENTATION APPLICATION TRAFFIC JAM DETECTION USING CAMERA ON HIGHWAY WITH PANORAMIC IMAGE

Whida Adi Nugroho<sup>1</sup>, Burhannudin Dirgantara, Ir., MT.<sup>2</sup>, Asep Mulyana, ST., MT.<sup>3</sup>

<sup>1,2,3</sup> Prodi S1 Sistem Komputer, Fakultas Teknik Elektro, Universitas Telkom

<sup>1</sup>[whida15@gmail.com](mailto:whida15@gmail.com), <sup>2</sup>[burhanuddin@telkomuniversity.ac.id](mailto:burhanuddin@telkomuniversity.ac.id), <sup>3</sup>[asepm267@gmail.com](mailto:asepm267@gmail.com)

### Abstrak

Kemacetan merupakan suatu hal yang tak bisa dihindarkan di kota besar. Banyak parameter yang bisa mendefinisikan bahwa kemacetan sedang terjadi. Aplikasi ini menggunakan parameter kemacetan berupa kecepatan objek terdeteksi. Adapun langkah untuk memperoleh parameter yang diperlukan untuk menentukan kemacetan. Yaitu, masukan data dari kamera menggunakan citra, citra yang diperoleh membentuk sudut panorama 180°, kamera diletakkan pada pinggir jalan, pendeteksian objek dikhususkan berupa mobil, pendeteksian dilakukan pada salah satu ruas jalan. Citra asli diolah dengan pengolahan citra digital untuk memudahkan proses deteksi pada citra. Objek mobil dideteksi menggunakan metode pengenalan objek *Haar Cascade Classifier*. Pengaturan pendeteksian berupa mobil dengan kriteria tertentu pada citra, tidak menitikberatkan pada semua mobil bisa terdeteksi. Penentuan kecepatan berdasarkan lima citra dengan sudut berbeda. Pergerakan mobil dalam citra terdeteksi menggunakan rumus  $v = s / t$ . Mobil terdeteksi pada salah satu sudut dibandingkan dengan mobil di sudut berikutnya. Sehingga diketahui apakah mobil terdeteksi diam atau bergerak menggunakan teknik *Feature Matching*. Dengan pengukuran kondisi jalan sebenarnya, diperoleh data yang kemudian diolah menjadi kecepatan. Kecepatan diartikan sebagai parameter menentukan kondisi yang terjadi pada suatu ruas jalan tertentu. Sehingga diperoleh hasil akhir berupa kemacetan salah satu ruas jalan, lancar ataukah macet.

**Kata Kunci :** Pengolahan Citra, *Haar Cascade Classifier*, *Feature Matching*.

### Abstract

*Traffic Jam is a matter that can not be avoided in the big city. Many parameters can be defined that congestion is happening. This application uses parameters such as the speed of the object detected congestion. The steps to obtain the necessary parameters to determine bottlenecks. Namely, the input data from the camera using the image, the image obtained by forming an angle of 180 ° panorama, the camera is placed on the roadside, object detection is devoted in the form of a car, the detection is done on one of the roads. The original image is processed by digital image processing to facilitate the detection process in the image. Car objects detected using Haar Cascade Classifier. Setting the detection of a car with a certain criteria on the image, do not focus on all of the car can be detected. Determination of speed by five image on difference angle obtained. The movement of cars in the image is detected using the formula  $v = s / t$ . Car detected in one angle compared to a car in the next angle. So it is unknown whether stationary or moving car detected using methods Feature Matching. By measuring the actual road conditions, obtained data are then processed into speed. Speed translates into the condition which occurs at a certain roads. In order to obtain the final result of a road congestion, smooth or jammed.*

*Keywords: Image Processing, Haar Classifier Cascade, Feature Matching.*

### 1. Pendahuluan

Kemacetan merupakan masalah serius yang perlu diatasi. Kemacetan banyak menimbulkan bagi pengguna jalan. Kemacetan merupakan definisi kata yang mempunyai arti tumpukan atau kepadatan kendaraan pada ruas jalan dan waktu tertentu. Macet atau tidaknya suatu jalan bisa ditentukan oleh banyak faktor, faktor yang menjadi pokok masalah disini yaitu kecepatan. Kendaraan bisa dikatakan melaju dengan kecepatan tertentu sebagai kondisi kemacetan suatu ruas jalan. Implementasi program yaitu mengolah parameter kecepatan menjadi kriteria penentuan terjadinya kemacetan.

Perancangan dilakukan dengan metode analisis terhadap citra masukan, kondisi ruas jalan, tinggi rendah peletakan alat penangkap citra. Implementasi menggunakan citra berbentuk panorama agar pandangan citra tidak sebesar satu bingkai citra. Objek yang terdeteksi menggunakan metode *Haar-Cascade Classifier*. Objek terdeteksi dicocokkan dengan *Feature Matcher* pada seluruh bingkai citra. Objek kemudian memasuki proses pengolahan sehingga dapat ditentukan kecepatan. Dari kecepatan diperoleh hasil akhir berupa kemacetan, yaitu kondisi macet dan kondisi lancar.

## 2. Dasar Teori<sup>[1][6][3]</sup>

Pengolahan citra merupakan proses yang dilakukan pada citra digital dengan menggunakan alat bantu berupa komputer dengan tujuan menjadikan citra digital mempunyai kualitas lebih baik [1]. Transformasi Intensitas Citra adalah besar intensitas setiap piksel pada citra diubah, tetapi posisi piksel tetap. Transformasi dilakukan dengan fungsi transformasi skala keabuan atau *Grayscale Transformation Function* (Castleman, 1996).

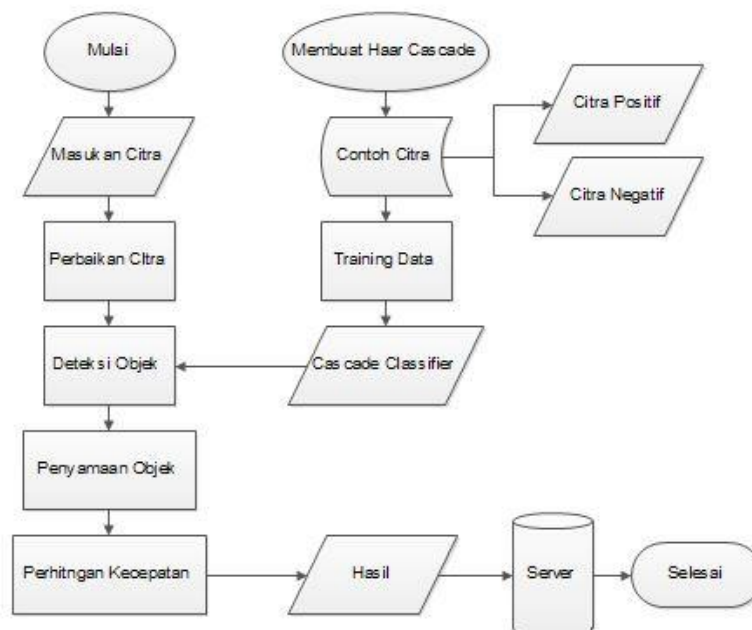
Citra yang mempunyai warna (RGB) dapat diubah menjadi citra *grayscale* dengan cara menghitung rata-rata elemen warna *Red*, *Green*, dan *Blue*. Operasi Ambang Batas (*Thresholding*) yaitu menentukan citra yang sudah berupa citra *grayscale* menjadi citra biner. Nilai piksel yang sudah memenuhi syarat ambang batas dipetakan ke suatu nilai yang sudah ditentukan sebelumnya.

*Haar-like feature* dikenal dengan sebutan *Haar Cascade Classifier*. *Haar-like features* merupakan *rectangular features*, yang memberikan indikasi secara spesifik suatu citra. *Haar-like features* mengenali objek berdasarkan nilai sederhana dari fitur tetapi bukan nilai piksel dari citra objek tersebut.

Metode ini memiliki kelebihan yaitu komputasinya sangat cepat, karena hanya bergantung pada jumlah piksel dalam persegi bukan setiap nilai piksel dari sebuah citra [6]. Metode ini merupakan metode yang menggunakan model statistik (*classifier*).

Cara kerja dari *haar* dalam mendeteksi objek alat transportasi darat dengan menggunakan teknik *sliding window* ukuran tergantung sudut citra dan mencari apakah terdapat bagian dari citra yang berbentuk seperti alat transportasi darat atau tidak. *Haar* juga memiliki kemampuan untuk melakukan *scaling* sehingga dapat mendeteksi adanya objek alat transportasi darat yang berukuran lebih besar ataupun lebih kecil dari citra hasil *classifier* [3].

## 3. Perancangan Sistem dan Implementasi



Gambar 3 Skema Perancangan

Seperti yang terlihat pada gambar, proses deteksi objek memerlukan data *Cascade Classifier* yang merupakan proses dari *training data* menggunakan metode *Haar Training*. Objek yang terdeteksi kemudian diolah menggunakan *Template Matching* berupa *feature Matcher* untuk menentukan apakah objek diam atau

bergerak dari kelima bingkai yang tersedia. Objek selanjutnya dihitung kecepatannya menggunakan data yang telah diperoleh. Hasil kemacetan dikirim ke server.

### 3.1 Haar Training

Citra latih merupakan proses terpenting pada pendeteksian suatu objek yang diinginkan. Mobil memiliki spesifikasi, warna dan bentuk yang beragam. Haar training memenuhi kriteria yang diterapkan untuk proses latih guna mendapatkan citra latih.

Sampling citra positif untuk citra dengan objek yang ingin dideteksi sebagai hasil dari citra latih. Sampling citra negatif berisi citra tanpa objek yang ingin dihindari pada hasil citra latih. Dari Sampling citra positif dan negatif dilatih menggunakan haar training dengan bantuan *library* dari *opencv*. Hasil berupa kumpulan matriks dan vektor yang disimpan dalam format *.txt* pada nama berkas sesuai dengan jumlah ranting dalam proses citra latih. Dari format tersebut, diubah menjadi format berbentuk *.xml* dan file bisa dipakai untuk mendeteksi objek yang telah berhasil menuntaskan proses citra latih.

### 3.2 Perbaikan Citra

Citra diperoleh melalui inputan *webcam* dengan lima kali pengambilan citra dengan sudut putar 32°. Citra hasil inputan diolah berdasarkan urutan sudut. Dilakukan proses penyamaan ukuran bingkai yaitu 320x240 yang bertujuan untuk mempercepat komputasi. Pada proses selanjutnya citra diolah dari citra RGB ke bentuk biner yaitu *Grayscale*. Tujuannya sebagai syarat *Haar Cascade Classifier* yang mengharuskan citra berbentuk biner. Untuk mempermudah mengenali objek. Citra *Grayscale* diolah menggunakan *Histogram Equalization*. Guna memperbaiki piksel yang memiliki derajat tingkat keabuan meragukan. Proses selanjutnya merupakan deteksi objek.

### 3.3 Deteksi Objek

Setelah mengalami peningkatan kualitas citra, citra dideteksi menggunakan metode *Haar Cascade Classifier* dengan pengaturan yang ditentukan. Apabila ditemukan objek yang sama dengan citra latih, dibuat garis kotak pada bingkai yang sudah ditentukan. Jika tidak ada objek yang sama dengan citra latih dapat ditarik kesimpulan bahwa citra latih memang belum melatih citra yang seharusnya terdeteksi, citra latih mengalami kerusakan pada saat pembuatan, dan tidak adanya objek atau citra berisi latar belakang semua.

### 3.4 Pengenalan Objek

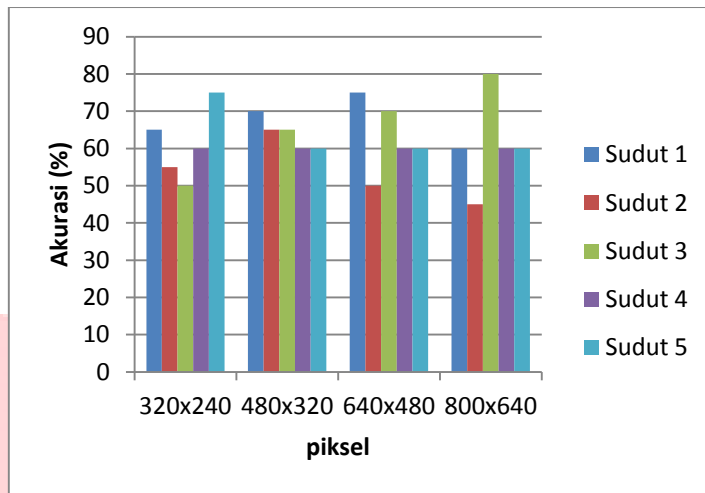
Objek yang berhasil terdeteksi pada citra dipotong dan disimpan dalam database. Citra selanjutnya diperoleh dengan sudut yang berbeda, citra diolah hingga proses deteksi objek. Apabila sudah lima kali dilakukan pengambilan citra berbeda maka citra terdeteksi dalam database dibandingkan mana yang sama dan mana yang tidak. Menggunakan metode *Feature Matcher* dapat diperoleh citra dengan tingkat piksel yang sama. Dari hasil tersebut ditarik kesimpulan bahwa objek adalah sama atau objek adalah tidak sama.

### 3.5 Perhitungan Kecepatan

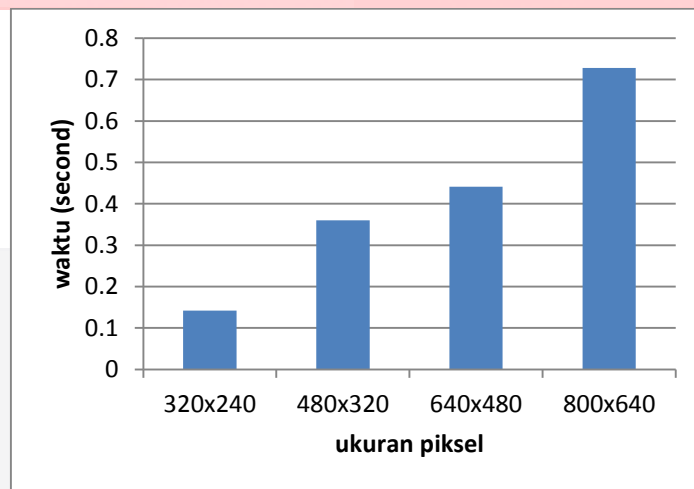
Kecepatan baru bisa diukur ketika dalam kondisi ada dua objek terdeteksi dalam lima kali pengambilan citra dengan sudut berbeda. Kecepatan diukur dengan parameter garis pada citra yang tercipta bila ada objek yang sama. Garis mewakili jarak dalam ukuran meter, per piksel didefinisikan berbeda pada tiap bingkai karena sudut tidak sama. Kamera menempuh satu gerakan digunakan sebagai parameter waktu. Menggunakan rumus sederhana  $v = s / t$  diperoleh kecepatan objek. Hasil kecepatan mepresentasikan tingkat kemacetan.

## 4. Pengujian Sistem dan Analisis

### A. Proses Deteksi Objek



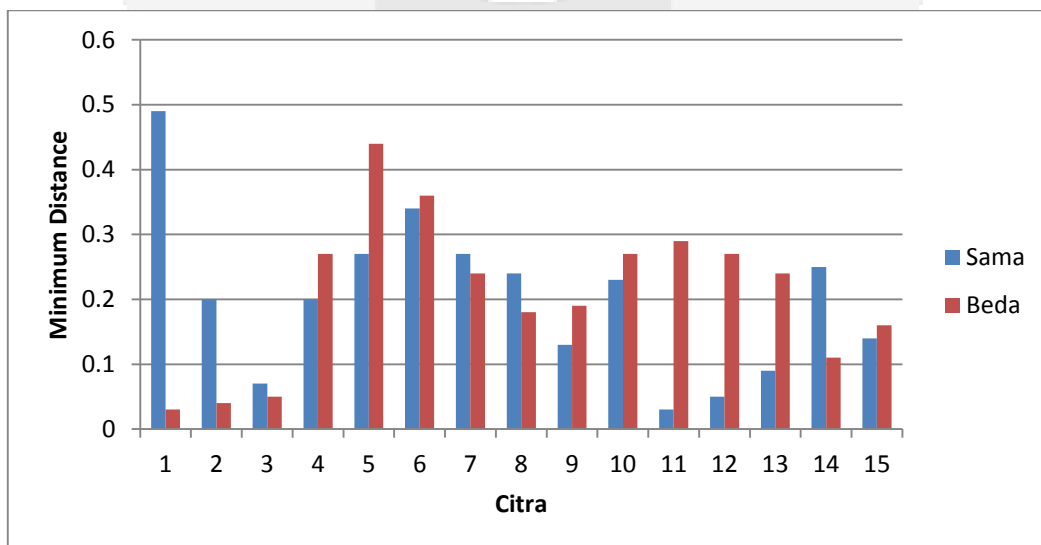
Gambar A.1 Grafik Akurasi Sistem



Gambar A.2 Grafik Waktu Komputasi

Dari pengujian diperoleh waktu komputasi dan akurasi sistem. Menunjukkan bahwa ukuran piksel terkecil dapat mempercepat komputasi. Pada gambar akurasi terlihat bahwa masing masing posisi sudut citra memiliki keunggulan tersendiri dengan di gantinya ukuran piksel.

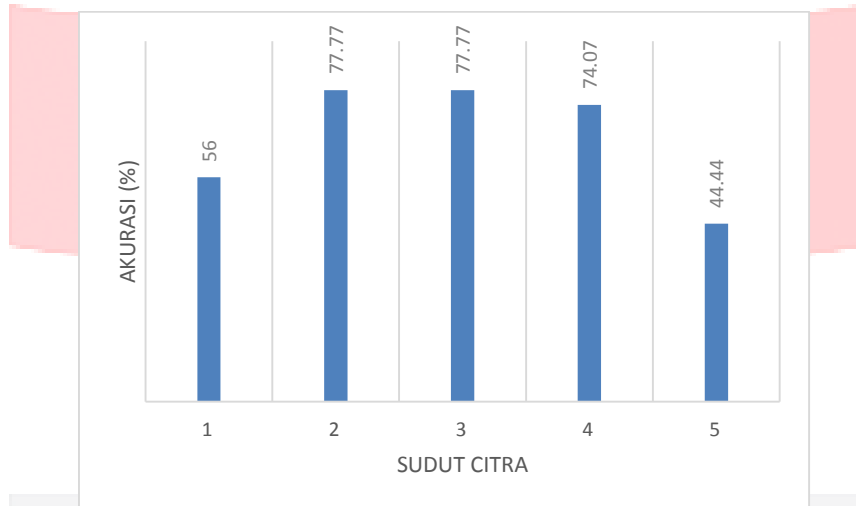
## B. Feature Matcher



Gambar B Grafik Uji Objek yang Sama

Dari pengujian disimpulkan bahwa parameter mengenali objek yang semirip mungkin menggunakan parameter *Minimum Distance*. Yaitu jarak terendah yang didapat saat titik kunci dicocokkan antara dua citra. Dengan nilai yang paling kecil, mempresentasikan bahwa citra tersebut mempunyai jumlah titik piksel yang sama. Namun pada kenyataannya, citra yang ingin dicocokkan merupakan citra berbeda sudut. Dari hasil pengujian diperoleh batas *minimum distance* sebesar 0,23 sebagai batas tertinggi citra bisa dikatakan sama. Pada hasil percobaan diperoleh akurasi 60%.

### C. Deteksi Kecepatan



**Gambar C** Grafik Hasil Deteksi Citra dengan Menentukan Kecepatan

Grafik menunjukkan tingkat akurasi program dalam mendeteksi objek pada masing masing sudut. Dari grafik terlihat bahwa akurasi rendah terjadi pada sudut 1 dan sudut 5. Kegagalan program untuk mendeteksi terjadi karena objek yang memiliki ukuran besar dan ukuran kecil. Pada kondisi sudut satu, objek dari kecil ke besar, sedangkan pada sudut 5 sebaliknya. Proses deteksi *Haar Cascade* tidak fokus pada ukuran piksel tertentu dan terjadi banyak ketidakakuratan proses deteksi. Pembatasan parameter *Haar Cascade* untuk membuat program meminimalisir objek terdeteksi menjadi sulit dilakukan.

### 5. Kesimpulan

Dari hasil proses pengujian dan analisa yang telah dilakukan pada sistem, dapat diambil kesimpulan sebagai berikut :

1. Program dapat mengenali objek dengan metode pengenalan objek *Haar Cascade Classifier*. Tingkat akurasi maksimal untuk mengenali objek yaitu sebesar 62,25%.
2. Program dapat mengidentifikasi kecepatan sebagai parameter memperoleh informasi kemacetan. Tingkat akurasi maksimal untuk memperoleh informasi kemacetan yaitu 70,37%.

### 6. Daftar Pustaka

- [1] Gonzales, R.C. dan R. E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company, 1992
- [2] Gupta, K., Kulkarni, A. V., "Implementation of an Automated Single Camera Object Tracking System Using Frame Differencing and Dynamic Template Matching", Indian Institute of Technology, Kanpur, India, 2008
- [3] Lienhart, Rainer and Jochen Maydt (2002). "An extended set of haar-like features for rapid object detection". In: *IEEE ICIP 2002*, Vol.1, pp 900-903.
- [4] Munir, Rinaldi, *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Informatika Bandung, 2004.
- [5] Sutoyo, T, Mulyanto. Edy, Suhartono. Vincent, Dwi Nurhayati Oky, Wijanarto, "Teori Pengolahan Citra Digital", Andi Yogyakarta dan UDINUS Semarang, 2009.
- [6] Viola, Paul and Michael Jones (2001). "Rapid object detection using boosted cascade of simple features".

In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition, 2001

[7] Wahyu Setyo Pambudi, Bon Maria Nurintan Simorangkir, (2012), Facetracker Menggunakan Metode Haar Like Feature Dan PID Pada Model Simulasi, Jurnal Teknologi dan Informatika (TEKNOMATIKA), Universitas Internasional Batam, Mei 2012..

[8] Y. Amit, D. Geman, and K. Wilder. "Joint Induction of Shape Features and Tree Classifiers", 1997.

