

Metode Haar-Cascade Classification Menggunakan Raspberry Pi

MUHAMMAD ICHWAN, MILDA GUSTIANA HUSADA, ALDRI HELMAPUTRA

Jurusan Teknik Informatika, Fakultas Teknologi Industri
Institut Teknologi Nasional Bandung

Email: ichwan@itenas.ac.id

ABSTRAK

Proses untuk mendeteksi objek dalam citra menggabungkan empat kunci utama yaitu *Haar like feature*, *Integral Image*, *Adaboost learning* dan *Cascade Classifier*. Metode *haar-cascade classification* dapat dimanfaatkan sebagai alternatif dalam pengaturan *traffic light*. Cara kerjanya yaitu, citra yang telah di-*capture* akan dicocokkan dengan data latih, apabila dalam citra tersebut terdapat objek yang sesuai dengan data latih maka objek tersebut di tandai, sehingga objek akan terhitung jumlahnya dengan cara melewati sebuah garis virtual yang dibuat sistem. Dari hasil jumlah objek terdeteksi, sistem dapat mengatur berapa lama lampu merah dan hijau menyala atau mati sesuai dengan jumlah objek yang telah ditentukan. Oleh karena itu, alternatif pengaturan *traffic light* menggunakan metode *Haar-cascade classification* digunakan untuk pendeteksian objek. Hasil dari penelitian menggunakan kamera 18 megapiksel 60 FPS jumlah mobil terhitung di garis lebih akurat menggunakan resolusi 480p atau 720p sebesar 79.51%.

Kata Kunci: *Pendeteksian Objek, Visi Computer, traffic light.*

ABSTRACT

The process to detect objects in the image grouping four main key Haar like features, Integral Image, Adaboost learning and Cascade Classifier. Haar-cascade method of classification can be utilized as an alternative to the traffic light settings. The way it works is, the image that has been captured will be matched with the training data, if there is an object in the image that correspond to the training data of the object on the mark, so the object will be numberless by crossing a virtual line that created the system. From the results the number of objects are detected, the system can set how long the red and green lights on or off according to the number of objects that have been determined. Therefore, alternative traffic light settings using Haar-Cascade Classification methods used for object detection. The results of the study using 18 megapixel camera 60 FPS counted the number of cars on the line more accurately using a resolution of 480p or 720p at 79.51%.

Keywords: *Detection Object, Computer Vision, Traffic Lights.*

1. PENDAHULUAN

Salah satu metode pendeteksian objek adalah metode *haar-cascade classification*. Metode *haar-cascade classification* bekerja dengan cara mengenali objek berdasarkan nilai sederhana dari fitur yang berupa jumlah piksel dalam persegi, sehingga objek yang ada dalam citra dapat dikenali dan ditandai oleh sistem. *Haar-cascade classification* merupakan metode yang menggunakan statistikal model (*classifier*). Proses untuk mendeteksi objek dalam citra menggabungkan empat kunci utama yaitu *haar like feature*, *Integral Image*, *adaboost learning* dan *cascade classifier*. Metode pendeteksian objek dapat dimanfaatkan sebagai metode alternatif untuk melakukan pengaturan durasi *traffic light*.

Pengaturan *traffic light* dilakukan dengan mendeteksi objek mobil melalui citra yang diambil dari kamera yang dipasang sebelum mobil mendekati ke *traffic light*. Objek yang terdeteksi dihitung jumlahnya dan ditandai ketika mengenai garis virtual yang akan mengatur nyala dan mati lampu merah atau lampu hijau dari jumlah objek yang terdeteksi.

Pada penelitian ini *device* yang digunakan berbasis Raspberry Pi menggunakan metode *haar-cascade classification* untuk proses pendeteksian kendaraan roda 4 atau lebih, kamera yang digunakan untuk pengambilan citra menggunakan resolusi 240p, 360p, 480p, 720p dan masing-masing resolusi menggunakan 30 FPS dan 60 FPS.

2. METODOLOGI PENELITIAN

Metode *haar-cascade classification* memiliki dua proses yaitu proses perhitungan *Haar-Like Feature* sebagai pendeteksian objek dan *Cascade Classification* sebagai pengklasifikasian.

2.1 Haar-like feature

Deteksi objek kendaraan merupakan pemodifikasian sistem *Haar-like features* dari deteksi wajah yang pertama kali dilakukan oleh Viola dan Jones kemudian dikembangkan oleh Lienhart (Lienhart, Rainer, & Maydt, 2002). Metode yang diusulkan Viola dan Jones menggabungkan empat kunci utama untuk mendeteksi sebuah objek [11]:

- Fitur persegi sederhana, disebut fitur *Haar*.
- Integral image* untuk pendeteksian fitur dengan cepat.
- Metode *AdaBoost machine-learning*.
- Cascade classifier* untuk mengkombinasikan banyak fitur.

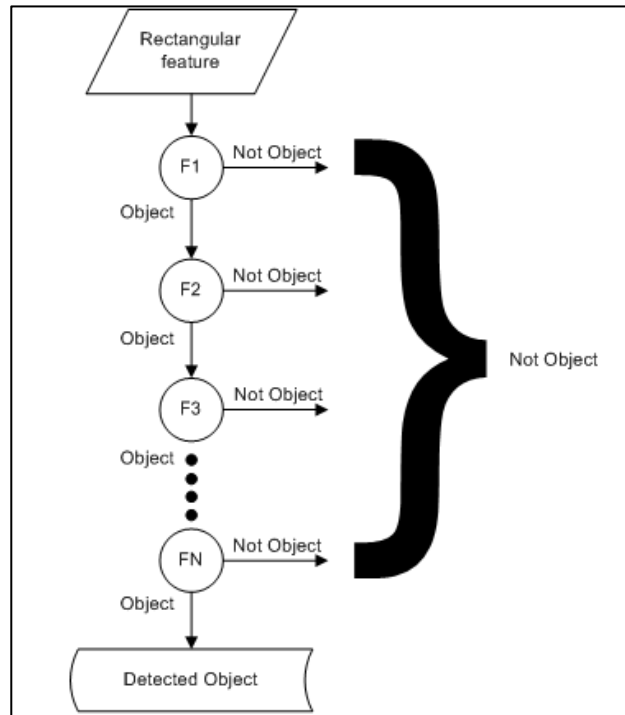
Haar-like feature memproses citra dalam sebuah kotak persegi dengan ukuran tertentu, misalnya 25 x 25 piksel.



Gambar 1. Rectangular Feature Haar Cascade

Di dalam kotak objek yang ditandai proses *filtering* objek dilakukan untuk diketahui apakah ada atau tidak objek yang akan dideteksi. Sebuah citra masukan melalui proses filterisasi

pertama (F1) dan proses filterisasi ini dilakukan secara bertingkat sampai objek terdeteksi dalam filterisasi yang ke-n (FN) hingga objek terdeteksi yang menyebabkan metode disebut sebagai *Haar-Cascade Classifier* [7].



Gambar 2. Skema Pendeteksi Objek

2.2 Cascade classification

Cascade classification merupakan proses pembelajaran berdasarkan gabungan dari beberapa pengklasifikasi (*multi-stages*), menggunakan semua informasi yang dikumpulkan dari data hasil diuji yang kemudian diberikan sebagai informasi tambahan untuk pengklasifikasian. Proses ini bertujuan untuk mendapatkan sebuah pohon keputusan yang dapat digunakan dalam pendeteksian objek. Metode *Haar-cascade classification* membutuhkan beberapa persiapan, yaitu:

- a. Pengumpulan gambar dan membuat deskripsi untuk sampel positif. Sampel positif merupakan objek yang akan dideteksi, contohnya ditunjukkan pada Gambar 3.



Gambar 3. Sampel Positif

- b. Pengumpulan gambar dan membuat deskripsi untuk sampel negatif. Sampel negatif merupakan background yang digunakan sebagai latar belakang dari objek yang ingin dideteksi, contohnya ditunjukkan pada Gambar 4.



Gambar 4 Sampel Negatif

- c. Membuat sampel vektor menggunakan sampel positif dan sampel negatif. Sampel positif dan sampel negatif yang telah terkumpul digabungkan menjadi sampel vektor. Contohnya ditunjukkan pada Gambar 5.



Gambar 5. Hasil Pembuatan Sampel Vektor

- d. Menggabungkan seluruh sampel vektor. Seluruh sampel vektor yang telah dibuat digabungkan menjadi satu *file* sampel vektor.
- e. Melatih sampel vektor untuk pengklasifikasian dan mengkonversi hasil. *File* sampel vektor yang telah menjadi satu dilatih menggunakan OpenCV *training cascade* menjadi bentuk *file* *.xml.
- f. Menggunakan *file* *.xml untuk mendeteksi objek. *File* *.xml digunakan untuk proses pendeteksian.



Gambar 6. Hasil Deteksi

2.3 Pendeteksian objek

Sebuah citra digital memiliki komponen nilai RGB (kombinasi dari warna merah, hijau dan biru). Dari nilai RGB dapat diketahui nilai *grayscale* (derajat keabu-abuan) yang dihitung menggunakan Persamaan (1) [3].

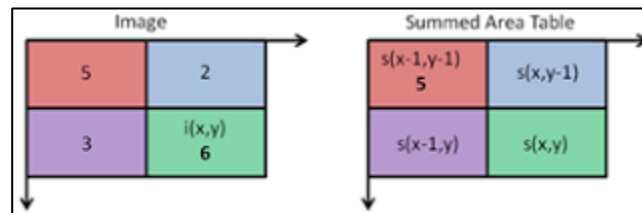
$$\text{Grayscale_pixel} = 0.2989R + 0.5870G + 0.1140B \tag{1}$$



Gambar 7 Grayscale

Proses *integral image* adalah sebuah citra yang nilai tiap pikselnya merupakan penjumlahan nilai piksel atas dan kirinya, dimana $x-1$ dan $y-1$ merupakan piksel tetangga dari piksel yang akan dihitung [3] menggunakan Persamaan 2.

$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x',y') \quad (2)$$



Gambar 8. Piksel Tetangga pada Proses *Integral Image*

Pada integral image matriks yang diperoleh dilakukan penghitungan nilai fitur *haar* yang disisipkan [3] menggunakan Persamaan 3 untuk mengetahui nilai objek yang bergerak.

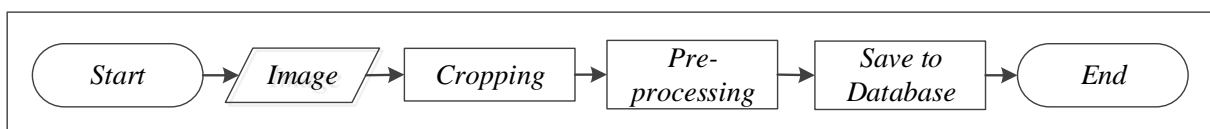
$$i(x',y')=s(A)+s(D)-s(B)-s(C) \quad (3)$$

Perhitungan nilai fitur haar menghasilkan nilai *threshold*. Apabila nilai fitur *Haar* lebih tinggi dari pada *threshold*, maka dapat dikatakan pada area tersebut memenuhi filter *Haar*. Proses ini akan dilanjutkan untuk menguji kembali area tersebut dengan filter *Haar* yang lain dan apabila seluruh filter *Haar* terpenuhi maka dikatakan pada area tersebut terdapat objek yang diamati.

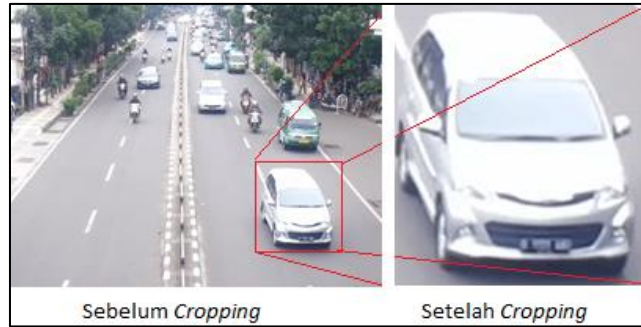
3. ANALISIS DAN PEMBAHASAN

Proses pendeteksian objek dibagi menjadi 2 yaitu proses data latih dan pendeteksian.

3.1 Pembuatan data latih



Gambar 9. Flowchart Data Latih



Gambar 10. pembuatan sampel positif

Cropping dilakukan dengan cara manual menggunakan *software* Adobe Photoshop untuk memisahkan sampel positif dan sampel negatif. Hasil dari citra yang telah di *cropping* dilakukan proses *grayscale*. Menggunakan Persamaan 1.

Contoh sampel positif yang ingin diubah menjadi *grayscale* memiliki nilai RGB yaitu, *red*=100, *green*=100, *blue*=100. Proses perhitungannya menggunakan Persamaan 1.

$$\begin{aligned} \text{diketahui} &= \text{Red}(R) = 100, \text{Green}(G) = 100, \text{Blue}(B) = 100 \\ \text{Grayscale} &= (0.2989 * 100) + (0.5870 * 100) + (0.1140 * 100) \\ &= 29.89 + 58.7 + 11.4 \\ &= 99.99 \end{aligned}$$



Gambar 11. Contoh Hasil *Grayscale*

Citra *grayscale* berupa sampel vektor yang berformat *file* *.xml digunakan untuk pendeteksian objek, dimana proses pendeteksian dijelaskan sebagai berikut.

Kamera mengambil citra dengan format *.avi, sebagai masukan pada pendeteksian objek sehingga objek yang ada dalam citra dapat dikenali sistem. Objek yang telah terdeteksi diberi tanda (*masking*) dengan sebuah titik hijau. Jika titik hijau mengenai garis maka dilakukan perhitungan jumlah objek. Objek yang telah terdeteksi dapat dilihat pada Gambar 12.



Gambar 12. Objek yang Telah di Masking

7	6	2	8
5	7	9	9
8	4	3	1
9	7	6	2

Gambar 13. nilai piksel pada citra

Pada pendeteksian objek dilakukan perhitungan secara manual. Citra masukkan disisipkan persegi citra baru yang di asumsikan sebagai citra dari objek yang bergerak dan dilakukan proses *grayscale*.

7	6	2	8
5	7	9	9
8	4	3	1
9	7	6	2

Gambar 14. Contoh Penyisipan Citra

Didapatkan hasil proses *grayscale*, dilakukan perhitungan *integral image* dengan menggunakan Persamaan 2.

		Sumbu X			
		7	6+7	2+6+7	8+2+6+7
Sumbu Y	5+7	7+7+6+5	9+7+5+2+6+7	9+9+7+5+8+2+6+7	
	5+8+7	4+7+6+8+5+7	3+4+8+9+7+5+2+6+7	1+3+4+8+5+7+9+9+7+6+2+8	
	9+8+5	7+4+7+6+9+8+	6+7+9+3+4+8+9+7+5+	7+6+7+9+1+3+4+8+9+9+7+5+8	
	+7	5+7	2+6+7	+2+6+7	

Gambar 15. Perhitungan *Integral Image*

7	13	15	23
12	25	36	53
20	37	51	69
29	53	73	98

Gambar 16. Hasil *Integral Image*

Didapatkan hasil perhitungan dengan menggunakan Persamaan 2 nilai *matriks* dilakukan penghitungan nilai fitur *haar* yang disisipkan menggunakan Persamaan 3 untuk mengetahui nilai objek yang bergerak.

		Sumbu X			
		7	13	15	23
Sumbu Y	12	(25+7)-12-13	(36+13)-25-15		53
	20	(37+12)-20-25	(51+25)-37-36		69
	29	(53+20)-29-37	(73+37)-53-51		98

Gambar 17 Perhitungan Nilai Fitur *Haar*

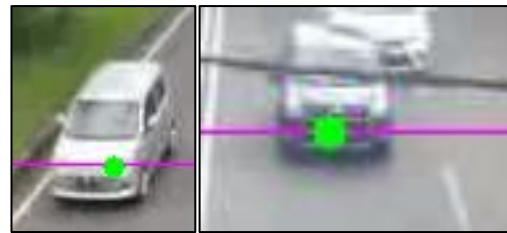
7	13	15	23
12	7	9	53
20	4	3	69
29	7	6	93

Gambar 18 Hasil Perhitungan Nilai Fitur Haar

Hasil dari perancangan sistem kemudian diimplementasikan menjadi sistem pendeteksian objek.



Gambar 19 Tampilan Pendeteksian



Gambar 20 Mobil Terhitung di Garis

3.2 Pengujian Sistem

Pada sistem pendeteksian objek dilakukan uji fungsi (*blackbox*) dan dilakukan uji *alpha*. Pengujian dilakukan pada 32 sampel video di jalan Merdeka dan jalan Soekarno Hatta. Pengujian dilakukan untuk melihat tingkat akurasi pendeteksian objek.

3.2.1 Pengujian tingkat akurasi

Pengujian tingkat akurasi bertujuan untuk menguji kemampuan dan ketepatan metode *haar-cascade classification* pada saat sistem melakukan pendeteksian terhadap benda yang bergerak, pengujian ini dilakukan dengan beberapa parameter sebagai pengukur tingkat ketepatannya dan menggunakan dua buah jenis kamera. Parameter yang digunakan dalam pengujian ini diantaranya adalah intensitas cahaya, posisi kamera, kualitas resolusi video.

- **Kamera 8 Megapiksel 30 Fps**
Pengujian Terhitung di Garis

Tabel 1 Pengujian Mobil Terhitung di Garis

No.	Jumlah Pengujian	Posisi Kamera		Cahaya (Lux)	Resolusi	Persentase
		Derajat (o)	Tinggi (m)			
1	8	5-10	5,1	5458 - 8351	240p 30fps	62.58 %
2	8	5-10	5,1	5458 - 8351	360p 30fps	67.54 %
3	8	5-10	5,1	5458 - 8351	480p 30fps	69.86 %
4	8	5-10	5,1	5458 - 8351	720p 30fps	67.54 %

Dari pengujian yang telah dilakukan didapatkan hasil menggunakan kamera 8 Megapiksel 30 fps mendapatkan nilai persentase tertinggi sebesar 67.54% menggunakan resolusi 360p dan 720p.

- **Kamera 18 Megapiksel 60 Fps**
Pengujian Terhitung di Garis

Tabel 5. Pengujian Mobil Terhitung di Garis

No.	Jumlah Pengujian	Posisi Kamera		Cahaya (Lux)	Resolusi	Persentase
		Derajat (o)	Tinggi (m)			
1	8	10	5,1	6091-9813	240p 60fps	73.27 %
2	8	10	5,1	6091-9813	360p 60fps	77.58 %
3	8	10	5,1	6091-9813	480p 60fps	79.31 %
4	8	10	5,1	6091-9813	720p 60fps	79.31 %

Dari pengujian yang telah dilakukan didapatkan hasil menggunakan kamer 18 Megapiksel 60 fps mendapatkan nilai persentase tertinggi sebesar 79.31% menggunakan resolusi 480p dan 720p.

3.3 Pebandingan Kamera 8 Megapiksel dan 18 Megapiksel

Tabel 6. Perbandingan Akurasi Kamera 8 Megapiksel dan 18 Megapiksel

Kamera	Resolusi	Posisi Kamera		Cahaya (Lux)	Hasil
		Derajat (o)	Tinggi (m)		
8 Megapiksel	240p 30fps	5-10	5,1	5458-8354	62.58 %
	360p 30fps	5-10	5,1	5458-8354	67.54 %
	480p 30fps	5-10	5,1	5458-8354	69.86 %
	720p 30fps	5-10	5,1	5458-8354	67.54 %
18 Megapiksel	240p 60fps	5	5,1	6091-9813	73.27 %
	360p 60fps	5	5,1	6091-9813	77.58 %
	480p 60fps	5	5,1	6091-9813	79.31 %
	720p 60fps	5	5,1	6091-9813	79.31 %

Dari pengujian yang telah dilakukan didapatkan hasil akurasi terbaik menggunakan kamera 18 megapiksel, resolusi 720p 60fps dengan persentase 79.31 %.

4. KESIMPULAN

Berdasarkan hasil analisis pengujian dapat disimpulkan :

Penelitian yang sudah dilakukan menggunakan metode *haar-cascade classification* berbasis Raspberry Pi mempunyai tingkat akurasi terbaik menggunakan kamera 18 megapiksel yang menggunakan 60 FPS dengan nilai persentase 79.31% berdasarkan pebandingan kamera 8 megapiksel 30 fps dan 18 megapiksel 60 fps.

DAFTAR PUSTAKA

- [1] Andriyani. "Konsep dasar *Pythor*". Universitas Gunadarma. Depok.
- [2] Booby Nicholas, Muhammad. 2016. "Deteksi Gender pada *Single Board Computer* sebagai Kamera CCTV". Institut Teknologi Nasional Bandung.

- [3] Derek, 2014. "Deteksi dan Pengenalan Rambu Pembatas Kecepatan Menggunakan Perangkat *Mobile*". Universitas Atma Jaya. Yogyakarta.
- [4] Hakim, M.A.I. 2013. "Pemanfaatan Mini PC Raspberry PI sebagai Pengontrol Lampu Jarak Jauh Berbasis WB pada Rumah". Universitas Komputer Indonesia. Bandung.
- [5] Piter Sanjaya, 2014. "CCTV (*Closed Circuit Television*)".
- [6] Prabowo Laksa Agung, Kurniawan Ricky. "Penerapan Teknologi *Computer Vision* dalam Bidang Kedokteran dan Biomedika". Universitas Gunadarma Depok.
- [7] Sajati, Haruno. 2015. "Deteksi Objek Menggunakan *Haar Cascade Classifier*".
- [8] Santana, Muhammad. 2015 "Aplikasi Pendeteksian Wajah Manusia untuk Menghitung Jumlah Manusia Menggunakan Metode Viola-Jones. Universitas Sumatra Utara.
- [9] Santosa Duwi, 2013. "Lampu Lalu Lintas (*Traffic Signal*)". Diunduh 26 September 2016 pukul 15.02
- [10] Syarif, Muhammad. 2015. "Deteksi Kedipan Mata Dengan Haar Cascade Classifier Dan Contour Untuk Password Login Sistem". Universitas Dian Nuswantoro Semarang.
- [11] Viola, Paul, & Jones, M. 2001. *Rapid object detection using boosted cascade of simple features*. In: *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*.