

SISTEM PENDETEKSI JUMLAH MOBIL DALAM INTELLIGENT TRANSPORT SYSTEM (ITS) MENGUNAKAN METODE VIOLA-JONES

Irmaya Citra Harwendhani^{*1}, Ika Purwanti Ningrum², Muh. Ihsan Sarita³

^{1,2,3}Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari

e-mail: ^{*1}irmayakamari@yahoo.com, ²ika.purwanti.n@gmail.com, ³ihsansarita@yahoo.co.id

Abstrak

Salah satu cara untuk mengatasi kepadatan lalu lintas adalah memanfaatkan teknologi *Intelligent Transport System* (ITS). ITS adalah sistem yang diterapkan untuk mengendalikan dan mengelola lalu lintas kendaraan, distribusi kendaraan dan infrastruktur untuk mencapai sistem transportasi yang lebih aman dan teratur. Perguliran waktu *traffic light* dalam ITS sifatnya fleksibel karena disesuaikan dengan jumlah kendaraan di lalu lintas. Untuk mendapatkan waktu perguliran *traffic light* yang efisien bergantung pada keakuratan perhitungan jumlah kendaraan.

Pada penelitian ini metode yang digunakan adalah metode *Viola-Jones*. Sistem pendeteksi jumlah mobil ini diimplementasikan menggunakan bahasa pemrograman *Java* berbasis *desktop*.

Hasil yang didapatkan dari sistem ini menunjukkan tingkat keakuratan tertinggi metode *viola-jones* dalam mendeteksi jumlah mobil terdapat pada kondisi sampel 1 buah mobil yakni sebesar 92,86%. Sedangkan kesalahan tertinggi metode *viola jones* dalam menentukan jumlah mobil terdapat pada kondisi sampel acak yakni sebesar 47,22%.

Kata kunci— ITS, Deteksi Mobil, Viola-Jones

Abstract

One way to cope with the traffic density is leveraging technology Intelligent Transport System (ITS). ITS is a system implemented to control and manage the traffic of vehicles, vehicle distribution and transport system infrastructure to achieve a more secure and organized. Revolving a traffic light in the ITS is flexible because it depends on the number of vehicles in traffic. To obtain the revolving time efficient traffic light depends on the accuracy of the calculation of the number of vehicles.

In this study, the method used is the method of viola-Jones. The number of car detection system is implemented using the Java programming language-based desktop.

Results obtained from these systems demonstrate the highest degree of accuracy viola-Jones method in detecting the number of cars on the condition of the samples contained 1 cars which amounted to 92,86%. While the highest error jones viola method in determining the number of cars found on a random sample conditions which amounted to 47,22%.

Keywords— ITS, Car Detection, Viola-Jones

1. PENDAHULUAN

Salah satu cara untuk mengefisienkan fungsi dari *traffic light* dan menyelesaikan permasalahan yang terjadi di lalu lintas adalah dengan memanfaatkan teknologi *Intelligent Transport Sistem* (ITS). ITS adalah penerapan teknologi informasi dan komunikasi dalam infrastruktur transportasi dan kendaraan sebagai alternatif

solusi untuk masalah kepadatan yang semakin tinggi di kota-kota besar. Sistem ini diterapkan untuk mengendalikan dan mengelola lalu lintas kendaraan, distribusi kendaraan dan infrastruktur untuk mencapai sistem transportasi yang lebih aman, lebih teratur dan perbaikan efisiensi sistem transit dan infrastruktur lalu lintas. Perguliran waktu *traffic light* dalam ITS sifatnya fleksibel karena disesuaikan dengan jumlah kendaraan di jalan.

Untuk mendapatkan keefisienan waktu perguliran *traffic light* bergantung pada keakuratan perhitungan jumlah kendaraan.

Salah satu metode yang dapat digunakan untuk menghitung jumlah kendaraan yaitu metode *Viola-Jones*. Algoritma *Viola-Jones* merupakan algoritma yang paling banyak digunakan untuk mendeteksi objek. Proses pendeteksian objek dilakukan dengan mengklasifikasikan sebuah gambar. Pendekatan metode *Viola-Jones* untuk mendeteksi objek dalam gambar menggunakan fitur *Haar* yang hanya bergantung pada jumlah *pixel* dalam persegi maka kelebihanannya yaitu komputasinya sangat cepat [1]. Penelitian sebelumnya mengenai metode *Viola-Jones* yaitu penelitian yang dilakukan oleh [2] Penelitian tersebut diterapkan pada deteksi wajah dengan hasil citra statis *error* sebesar 28% terjadi pada saat gambar berisi 7 orang dengan jumlah wajah terdeteksi sebanyak 9 wajah. Hal yang berbeda dilakukan oleh [3] dalam studi deteksi wajah metode *viola-jones* pada OpenCV menggunakan pemrograman *Python*. Dalam kesimpulannya, sistem dapat mendeteksi adanya beberapa (lebih dari satu) wajah dalam suatu citra. Sistem juga dapat mendeteksi objek yang menyerupai wajah ketika objek tersebut memiliki kontur yang sama dengan kontur wajah manusia (kontur wajah pada template), misalnya, wajah boneka dan topeng Hulk.

2. METODE PENELITIAN

2.1 Data

Data yang digunakan dari penelitian tugas akhir ini adalah data yang langsung berkaitan dengan objek penelitian, dalam hal ini data diperoleh melalui pengambilan data langsung di lapangan yaitu di kawasan lampu merah Jl. tol di kota Makassar dan kawasan sekitar lampu merah eks MTQ di Kota Kendari

Jumlah data yang digunakan sejumlah 62 sampel uji yang terdiri dari :

1. Dua belas (12) sampel posisi 45°
2. Empat belas (14) sampel uji 1 mobil
3. Dua belas (12) sampel uji 2 mobil
4. Dua belas (12) sampel uji lebih dari 3 mobil
5. Dua belas (12) sampel uji acak (posisi belakang, samping kiri, samping kanan)

2.2 ITS (Intelligent Transport Sistem)

Menghadapi kemajuan teknologi dalam transportasi terdapat pilihan, yaitu memperbaiki teknologi yang ada sekarang atau pembangunan teknologi baru. Pembangunan teknologi baru membutuhkan tersedianya dana yang sangat besar [4]. *Traffic light* saat ini dituntut untuk dapat menyelesaikan permasalahan perguliran waktu yang sesuai pada kebutuhan persimpangan jalan serta hal terkait selanjutnya kemampuan nalar manusia sehingga solusi paling tepat yang ditawarkan yaitu penerapan *artificial intelligence (AI)* pada *traffic light* tersebut.

Oleh [5] menyatakan bahwa AI berusaha membangun entitas-entitas cerdas yang sesuai dengan pemahaman manusia, dan entitas tersebut ternyata sangat menarik dan mempercepat proses pemahaman terhadap kecerdasan manusia. Oleh [6] menambahkan bahwa Teknologi ITS (*Intelligent Transport Sistem*) adalah salah satu cabang AI di bidang transportasi yang baru berkembang beberapa tahun terakhir untuk mengatasi kemacetan lalu lintas di beberapa negara maju.

2.3 Objek Dalam Citra

Mengenali suatu objek dalam citra dapat dilakukan dengan cara mengidentifikasi pola objek tersebut. Secara umum, pengertian pola (*pattern*), atau dikenal dengan istilah ciri/fitur, adalah komposit, gabungan atau himpunan dari fitur yang merupakan sifat dari suatu objek. Ciri/ fitur adalah segala jenis aspek pembeda atau cirri-ciri yang membedakan. Ciri inilah yang digunakan dalam melakukan identifikasi terhadap objek yang dikenali. Oleh karena itu pemilihan fitur sangat menentukan keberhasilan dalam pengenalan pola. Ukuran fitur diperoleh dari hasil ekstraksi fitur pada objek. Ukuran fitur berwujud simbolik (misalnya warna) atau *numeric* (misalnya tinggi). Fitur yang bagus adalah fitur yang memiliki daya pembeda yang tinggi, sehingga pengenalan/ pengelompokan pola berdasarkan fitur yang dapat dilakukan dengan keakuratan yang tinggi [6].

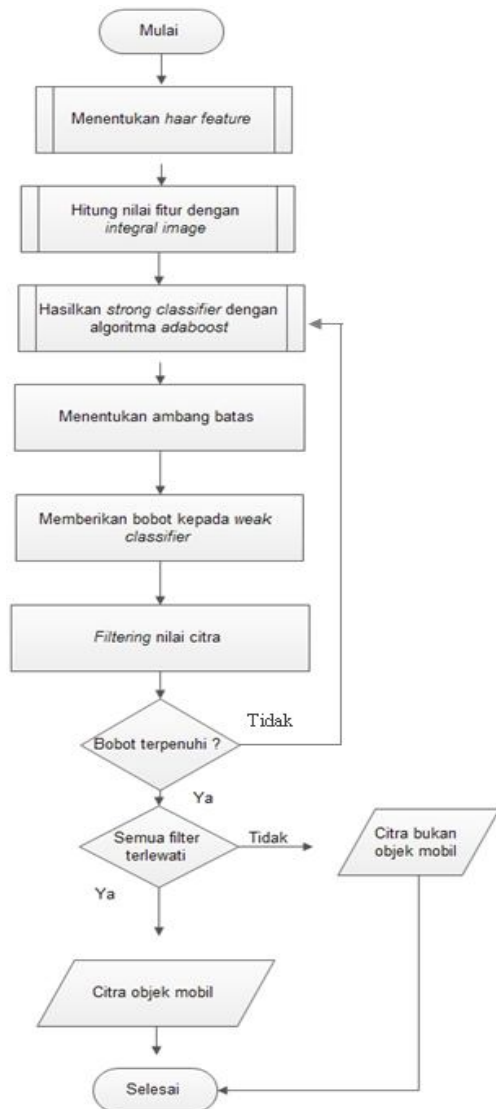
2.4 Viola-Jones

Proses deteksi adanya citra objek dalam sebuah gambar dapat dilakukan dalam OpenCV (*Open Computer Vision*) yang diintegrasikan dengan software pemrograman. Salah satu metode pendeteksi objek yang

umum saat ini yaitu menggunakan sebuah metode yang dipublikasikan oleh Paul Viola dan Michael Jones tahun 2001. Umumnya disebut metode *Viola-Jones*. Pendekatan untuk mendeteksi objek dalam gambar menggabungkan empat konsep utama :

1. Fitur segi empat sederhana yang disebut fitur *Haar*.
2. *Integral* gambar untuk pendeteksian fitur secara cepat.
3. Metode *machine learning adaboost*.
4. Pengklasifikasi bertingkat (*cascade classifier*) untuk menghubungkan banyak fitur secara efisien.

Gambar 1 menunjukkan *flowchart* seleksi fitur Haar dengan *Adaboost*.



Gambar 1 *Flowchart* seleksi fitur *haar* dengan *adaboost*

Sebuah *fitur haar* diawali dengan meng-input citra lalu citra *image* di *prerprocessing* menjadi citra *grayscale* yang akan dicari fitur positif dengan *AdaBoost* dan *Cascade Classifier*. Proses konversi *image* dari RGB ke *Grayscale* menggunakan persamaan dibawah ini :

Misalkan suatu citra objek memiliki nilai $R = 100, G = 100, B = 100$, maka nilai *grayscale* dari citra tersebut dapat dihitung menggunakan Persamaan (1).

$$X = (0,299R) + (0,587G) + (0,114B) \quad (1)$$

diperoleh

$$X = (0,299 * 100) + (0,587 * 100) + (0,114 * 100) = 99,99$$

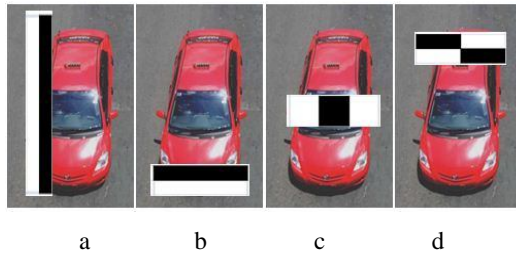
Gambar 2 adalah contoh citra hasil *grayscale*ing.



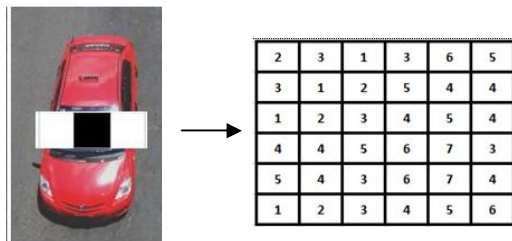
Gambar 2 Contoh konversi citra RGB menjadi citra *grayscale*

Setelah melakukan proses *grayscale* maka proses selanjutnya adalah *scaling* dan *sliding* pada *window*. Jika objek terdeteksi, akan dilakukan penggambaran garis persegi pada gambar mobil tersebut. Pendeteksian objek menggolongkan gambar berdasarkan pada nilai dari fitur sederhana. Operasi dasar dari suatu fitur jauh lebih cepat dibandingkan dengan pengolahan *pixel*. Sejumlah Fitur *haar* mewakili wilayah persegi pada citra dan menjumlahkan semua *pixel* pada daerah tersebut.

Viola-jones mengklasifikasikan citra dari nilai fitur-fitur sederhana dan menggunakan tiga jenis fitur, yaitu fitur persegi, fitur tiga persegi, dan fitur empat persegi. Nilai dari fitur-fitur tersebut adalah selisih antara daerah hitam dan putih. Gambar 3 menunjukkan sample perhitungan *integral image*.

Gambar 3. Sampel perhitungan *integral image*

Pertama, ditentukan terlebih dahulu area yang akan dideteksi apakah terdapat obyek atau tidak. Proses berikutnya adalah melakukan pendeteksian obyek menggunakan *Haar Cascade Clasifier* dengan langkah-langkah yang akan dijelaskan sebagai berikut. *Integral image* adalah sebuah citra yang nilai tiap *pixel*-nya merupakan penjumlahan nilai *pixel* atas dan kirinya. Sebagai contoh pada Gambar 3 (c) sebuah daerah persegi yang akan di-scan menggunakan persegi gelap terang memiliki nilai yang ditunjukkan oleh Gambar 4.



Gambar 4. Citra Masukan

Pada citra masukan yang diberi persegi pada Gambar 4 terlihat pada Gambar 5.

2	3	1	3	6	5
3	1	2	5	4	4
1	2	3	4	5	4
4	4	5	6	7	3
5	4	3	6	7	4
1	2	3	4	5	6

Gambar 5. Persegi *haar like* dari citra masukan pada Gambar 4

Proses perhitungan nilai gelap dan nilai terang dinyatakan dalam Persamaan (2).

$$F_{Haar} = |Total\ pixel\ hitam - Total\ pixel\ putih| \quad (2)$$

Untuk menghitung Fitur nilai *Haar* menggunakan *Summed Area Table* atau yang

dikenal sebagai *Integral image*, pertama dibentuk matriks nilai *integral image* dimana $x - 1$ dan $y - 1$ adalah pixel tetangga dari pixel yang akan dihitung *integral image* dari citra masukan pada Gambar 5 terlihat pada Gambar 6.

2	3	1	3	6	5
3	1	2	5	4	4
1	2	3	4	5	4
4	4	5	6	7	3
5	4	3	6	7	4
1	2	3	4	5	6

2	5	6	9	15	20
5	9	12	20	30	39
6	12	18	30	45	58
10	20	31	49	71	87
15	29	43	67	96	116
16	32	49	77	111	137

Gambar 6. Matriks *Integral Image* dari Citra Masukan

Nilai fitur *haar* dari area matriks pada Gambar 4 dapat dihitung menggunakan Persamaan (2) sehingga didapatkan hasil sebagai berikut :

$$\begin{aligned}
 F_{Haar} &= |Total\ pixel\ hitam \\
 &\quad - Total\ pixel\ putih| \\
 &= |(32 + 2 - 16 - 5) - (49 + 5 - 32 - 6) \\
 &\quad + (77 + 6 - 49 - 9)| \\
 &= |13 - 16 + 25| \\
 &= 22
 \end{aligned}$$

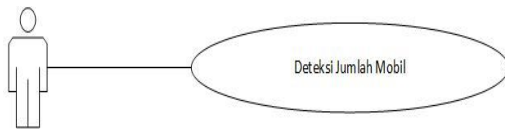
Hasil perhitungan secara manual menggunakan Persamaan (2). Nilai 22 tersebut kemudian dibandingkan dengan *threshold* yang sudah ditentukan sebagai pendeteksian obyek. Apabila nilai fitur *haar* lebih tinggi daripada *threshold*, maka dapat dikatakan pada area tersebut memenuhi filter *haar*. Sesuai *flowchart* pada Gambar 1, 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 obyek yang diamati.

2.5 Perancangan Sistem

Rancangan sistem pendeteksi jumlah mobil dalam *Intelligent Transport Sistem (ITS)* menggunakan metode *viola-jones* disajikan menggunakan *Unified Modelling Language (UML)* berupa diagram *use case*, diagram *activity*, dan diagram.

A. Use Case Diagram

Adapun diagram *use case* yang dapat menggambarkan rancangan sistem pendeteksi jumlah mobil ditunjukkan oleh Gambar 7.

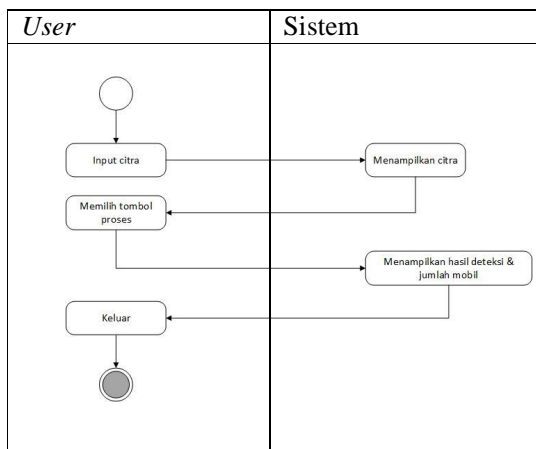


Gambar 7 Diagram *use case* sistem pendeteksi jumlah mobil

Berdasarkan Gambar 7, aplikasi sistem pendeteksi jumlah mobil dapat diakses oleh seorang *user*.

a) Activity Diagram

Diagram activity adalah diagram yang menggambarkan semua aktifitas yang dilakukan oleh *user*, ditunjukkan oleh Gambar 8..



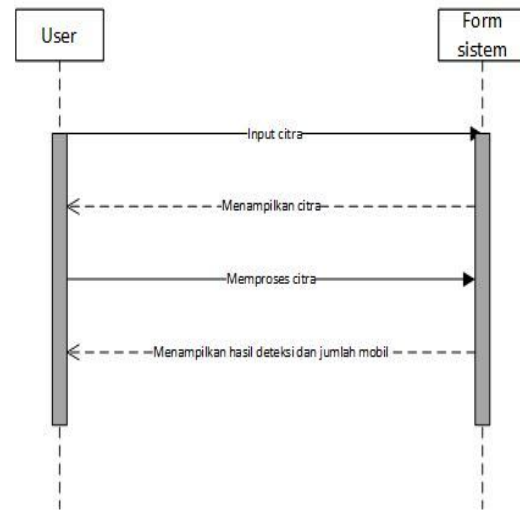
Gambar 8 Diagram *activity* sistem pendeteksi jumlah mobil

Diagram *activity* pada Gambar 8 menunjukkan bahwa seorang *user* dapat meng-*input*-kan sebuah citra lalu sistem akan menampilkan citra yang telah di-*input* oleh *user*. Selanjutnya, *user* memilih tombol proses dan sistem akan menampilkan hasil deteksi dan jumlah mobil. Apabila *user* telah melakukan pendeteksian maka *user* dapat menekan tombol keluar untuk mengakhiri proses.

b) Sequence Diagram

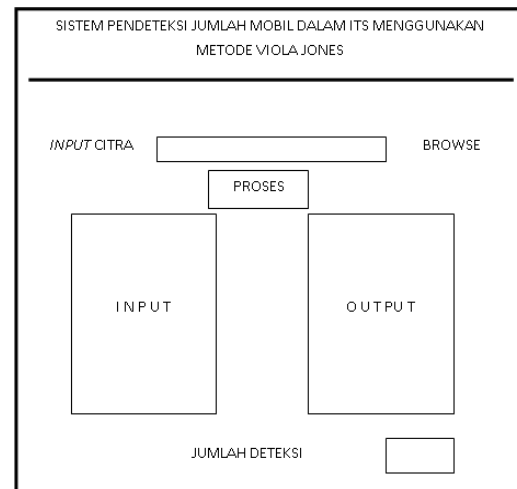
Gambar 9 menunjukan urutan kinerja sistem diawali dengan *user* meng-*input* citra lalu sistem akan menampilkan citra dan memproses citra hingga menampilkan hasil deteksi dan jumlah mobil.

dapat dilakukan oleh pengguna dalam sistem. Adapun perancangan antarmuka ini diuraikan sebagai berikut.



Gambar 9 *Sequence* diagram sistem pendeteksi jumlah mobil

Pada sistem pendeteksian jumlah mobil, sistem ini menggunakan 1 *form* dimana terdapat tombol proses untuk menampilkan jumlah mobil. Adapun rancangan *interface* sistem pendeteksi jumlah mobil dalam *Intelligent Transport System* (ITS) menggunakan metode *Viola – Jones* ditunjukkan oleh Gambar 10.



Gambar 10 *Interface* sistem pendeteksi jumlah mobil

Gambar 10 merupakan tampilan aplikasi sistem pendeteksi jumlah mobil. *User* memilih citra *image* yang akan diproses. Pada *form* tersebut terdapat tombol proses digunakan untuk memproses citra *image* yang telah di-

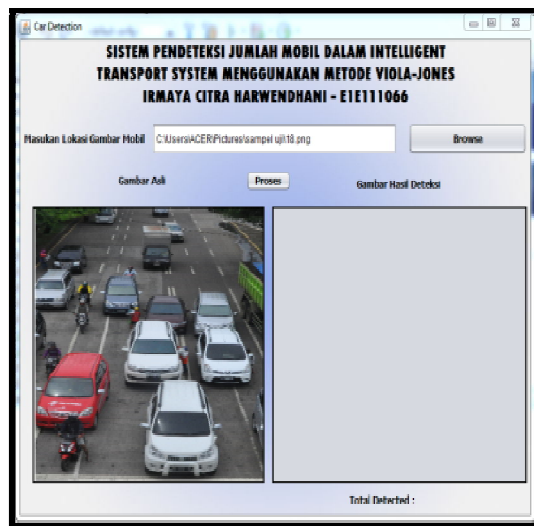
B. Perancangan Antarmuka

Perancangan antarmuka pengguna atau *design user interface* merupakan penggambaran tampilan yang digunakan secara langsung oleh pengguna, dan interaksi yang

input sebelumnya untuk mengetahui berapa jumlah mobil yang terdapat pada citra uji. Setelah *user* menekan tombol proses, maka sistem akan menampilkan jumlah mobil hasil deteksi.

3. HASIL DAN PEMBAHASAN

Pada aplikasi pendeteksi jumlah mobil untuk melakukan proses deteksi mobil, *user* terlebih dahulu harus memilih citra/gambar yang akan diuji. Setelah citra telah di-*input* maka akan muncul tampilan *input* citra yang ditunjukkan oleh Gambar 7.



Gambar 7. Tampilan *input* citra

Pada aplikasi deteksi jumlah mobil, hanya terdapat 1 tombol proses, tombol ini berfungsi untuk melakukan proses deteksi mobil menggunakan metode *Viola-Jones*. Ketika *user* menekan tombol proses maka akan muncul tampilan seperti pada Gambar 8.



Gambar 8. Tampilan proses deteksi jumlah mobil

Tabel 1 menunjukkan bahwa sampel yang diuji berupa sampel dengan jumlah 1 buah mobil pada setiap sampel. Sampel uji terdiri dari 14 sampel, jumlah nilai aktual pada tabel di atas sebanyak 14 mobil. Terdapat 14 mobil yang terdeteksi total pada semua sampel uji. Namun hanya 12 sampel yang benar terdeteksi hanya mobil. Untuk mengetahui tingkat keakuratan jumlah mobil menggunakan metode *Viola-Jones* ini adalah dengan mengetahui nilai RMSE dan nilai keakuratan. *Root Mean Square Error* (RMSE) dapat dihitung :

$$RMSE = \frac{\sqrt{\frac{1}{N} \sum_{t=h}^N (A-M)^2}}{A_{max} - A_{min}}$$

$$RMSE = \frac{\sqrt{\frac{1}{14} \sum_{t=h}^N (14 - 15)^2}}{3 - 1}$$

$$RMSE = \frac{\sqrt{\frac{1}{14} * 1}}{2}$$

$$RMSE = 0,13$$

Nilai keakuratan dapat dihitung menggunakan persamaan :

$$\% \text{ keakuratan} = \frac{\sum \text{Deteksi Mobil}}{\sum \text{Aktual}} * 100\%$$

$$\% \text{ keakuratan} = \frac{13}{14} * 100\%$$

$$\% \text{ keakuratan} = 92,85\%$$

$$\% \text{ kesalahan non objek} = 100 - \% \text{ keakuratan}$$




















$$\% \text{ kesalahan} = 100 - 92,85\%$$









$$\% \text{ kesalahan} = 7,14\%$$

Berdasarkan hasil perhitungan tersebut, nilai keakuratan yang diperoleh dari sampel uji 1 buah mobil, adalah 92,85% dan tingkat kesalahan pengenalan non-objek yang dideteksi sebagai objek dalam sistem ini yaitu sebesar 7,14%.

Hasil pengujian tersebut memperlihatkan hubungan antara hasil perhitungan aktual dengan jumlah mobil terdeteksi sebagai acuan perhitungan RMSE. Sedangkan untuk hubungan antara kondisi sampel dengan hasil pengamatan sebagai acuan perhitungan keakuratan sistem ditunjukkan oleh Tabel 2.

Tabel 1 Pengujian Sistem

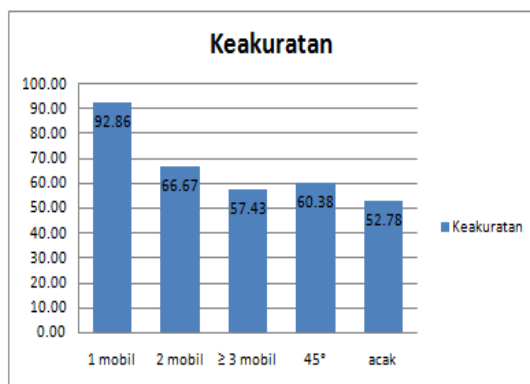
Sampel	Sebelum deteksi	Setelah deteksi	Aktual	Deteksi total	Deteksi hanya mobil
Uji 1			1	1	1
Uji 2			1	1	1
Uji 3			1	1	1
Uji 4			1	1	1
Uji 5			1	0	0
Uji 6			1	1	1
Uji 7			1	1	1
Uji 8			1	1	1
Uji 9			1	1	1
Uji 10			1	1	1

Uji 11			1	1	1
Uji 12			1	1	1
Uji 13			1	1	1
Uji 14			1	3	1
Jumlah			14	15	13

Tabel 2 Hubungan antara kondisi sampel dan hasil pengamatan.

Hasil Pengamatan	Kondisi Sampel				
	1 Mobil	2 Mobil	≥ 3 Mobil	45°	Acak
Aktual	14	24	101	53	36
Deteksi Total	15	22	95	44	103
Deteksi Mobil	13	16	58	32	19
Keakuratan	92,86	66,67	57,43	60,38	52,78
Kesalahan	7,14	33,33	42,57	39,62	47,22

Tabel 2 merupakan rangkuman hasil pengamatan seluruh pengujian sampel, sedangkan nilai keakuratan terbesar pengujian ditunjukkan oleh Gambar 10.



Gambar 10 Grafik nilai keakuratan

Gambar 10 menunjukkan hubungan kondisi sampel di lapangan dengan nilai keakuratan pada pengamatan. Gambar 10 memperlihatkan bahwa nilai keakuratan tertinggi diperoleh oleh pengujian sampel dengan kondisi 1 buah mobil yaitu 92,86 %, sedangkan keakuratan terendah terjadi pada kondisi sampel pengujian secara acak yaitu 52,78 %. Dapat diartikan bahwa keakuratan pendeteksian sistem sangat baik jika kondisi sampel dengan jumlah mobil sedikit yaitu 1 mobil.

4. KESIMPULAN

Berdasarkan implementasi dan evaluasi sistem pendeteksian jumlah mobil dalam *Intelligent Transport Sistem (ITS)* menggunakan metode *Viola-Jones* dapat disimpulkan bahwa keakuratan tertinggi metode *Viola-Jones* dalam menentukan jumlah mobil sebesar 92,86% yaitu terdapat pada kondisi sampel 1 buah mobil. Sedangkan keakuratan terendah sebesar 52,78% diperoleh pada kondisi sampel acak. Hal ini berarti bahwa keakuratan pendeteksian sistem pendeteksian jumlah mobil menggunakan metode *Viola-Jones* sangat baik dalam mendeteksi jumlah mobil dalam kondisi sampel 1 buah mobil

5. SARAN

Saran yang dapat dipertimbangkan untuk pengembangan sistem lebih lanjut terkait dengan penelitian ini adalah pada pengembangan selanjutnya, dapat menambahkan fitur sistem absensi dari Siakad.

DAFTAR PUSTAKA

- [1] Bharata, A., 2012, *Sistem pendeteksi wajah dengan menggunakan metode Viola-Jones*, Universitas Gajah Mada: Yogyakarta
- [2] Kusumanto, R.D., 2012, *Aplikasi Sensor Vision Untuk Deteksi Multiface dan Menghitung Jumlah Orang*, Universitas Internasional Batam: Batam.
- [3] Prasetya, 2012, *Deteksi Wajah Metode Viola-Jones pada OpenCV menggunakan pemrograman python*. Universitas Muhamadiyah Surakarta : Kartasura
- [4] Adisasmita, S.A., 2011, *Perencanaan Pembangunan Transportasi*, Yogyakarta, Graham Ilmu.
- [5] Suyanto, 2011, *Artificial Intelligence, Searching, Reasoning, Planning, dan Learning*, Informatika Bandung, Jakarta.
- [6] Suyuti, R., 2012, Implementasi Intelligent Transportation System (ITS) Untuk Mengatasi Kemacetan Lalu Lintas di DKI Jakarta, *Jurnal Konstruksia Vol. 3. No. 2. 2012*, Universitas Muhammadiyah, Jakarta.

