

APLIKASI PENGENALAN WAJAH MENGGUNAKAN METODE *EIGENFACE* DENGAN BAHASA PEMROGRAMAN JAVA

Anita T. Kurniawati dan Afrilyan Ruli Dwi Rama

Teknik Informatika-ITATS, Jl. Arief Rahman Hakim 100 Surabaya

Email: anitateku@yahoo.com

ABSTRAK

Salah satu bagian dari sistem biometrika adalah *Face recognition* (pengenalan wajah yang banyak digunakan untuk identifikasi personal pada penggunaan mesin absensi, akses kontrol, keamanan dan lain-lain. Pada penelitian ini, dibuat suatu aplikasi untuk mengenali wajah dengan metode *Eigenfaces*. Sedangkan untuk mendeteksi wajah digunakan metode Viola Jones. Pengambilan citra wajah sebanyak 120 image dari 6 orang dengan menggunakan webcam. Melalui tahap pemrosesan citra wajah RGB diubah menjadi citra wajah *histogram qualization* untuk selanjutnya dilakukan tahap pengenalan menggunakan metode *eigenface*. Metode *eigenface* ini terdapat beberapa tahapan inti, yaitu citra wajah diubah menjadi matriks, dihitung *eigenvalue* dan *eigenvector*, sehingga diperoleh nilai *eigenface*-nya yang selanjutnya dilakukan proses pengenalan wajah dengan mencari nilai *eigenface* yang mendekati. Hasil pengujian untuk mendeteksi wajah dengan Viola Jones adalah akurasi ketika wajah dalam posisi frontal dengan webcam sebesar 100% dengan waktu deteksi kurang dari 1 detik. Sedangkan batas kemiringan maksimum ± 70 dan jaraknya 20 – 120 cm. Pengujian sistem menggunakan metode *Eigenface* diperoleh nilai akurasi sebesar 90%.

Kata Kunci: Pengenalan Wajah, Viola Jones, *Eigenface*.

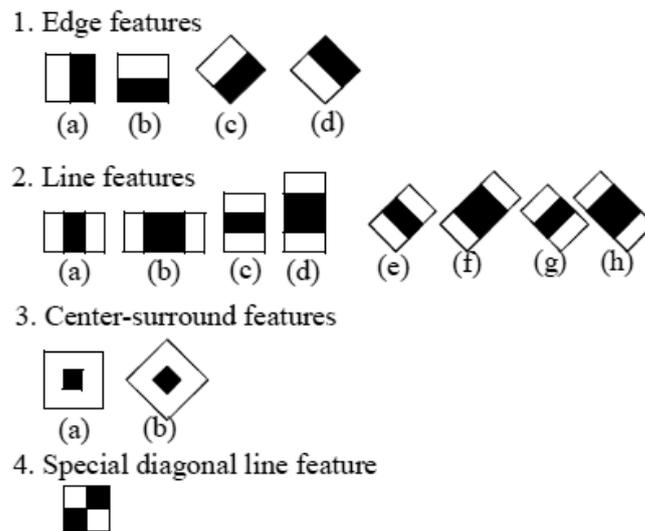
PENDAHULUAN

Sistem biometrika merupakan teknologi pengenalan diri dengan menggunakan bagian tubuh atau perilaku manusia yang memiliki keunikan. Salah satu bagian sistem biometrika adalah *face recognition* (pengenalan wajah) yang banyak digunakan untuk identifikasi personal pada penggunaan mesin absensi, akses kontrol, keamanan dan lain-lain. Banyak penelitian merancang sistem pengenalan wajah dengan berbagai metode yang ada. Seperti Prasetya melakukan deteksi wajah dengan metode Viola Jones menggunakan bahasa pemrograman Python. Jika dilakukan pendeteksian wajah dalam keadaan frontal, tingkat akurasi sistem sebesar 100%. [5] Putra merancang aplikasi absensi deteksi wajah dengan menggunakan metode *eigenface*. Aplikasi yang dirancang dengan menggunakan program matlab. [6]

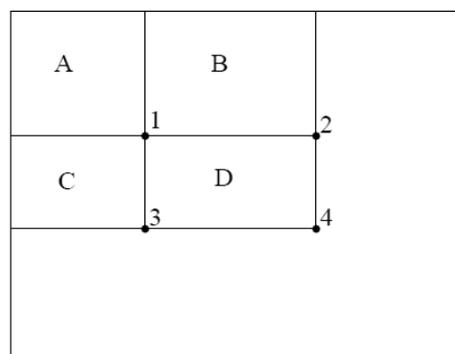
Pada penelitian ini akan dibuat aplikasi pengenalan wajah dengan menggunakan metode *Eigenface*. Sedangkan pendeteksian wajah dengan metode Viola Jones. Aplikasi yang dibuat dengan menggunakan software NetBeans 8.02 dan bahasa pemrograman Java.

DASAR TEORI

Sistem pengenalan wajah secara umum dibagi menjadi dua tahap yaitu sistem deteksi wajah yang merupakan tahap awal (*pre-processing*) dan dilanjutkan dengan sistem pengenalan wajah (*face recognition*). [1] Tahap pendeteksian wajah digunakan metode Viola Jones dimana metode ini ada 4 tahapan. Tahap pertama adalah penggunaan fitur *Haar* dengan konsep seperti pada Gambar 1. [4] Tahap kedua adalah perhitungan *Integral Image* seperti pada Gambar 2. [7]



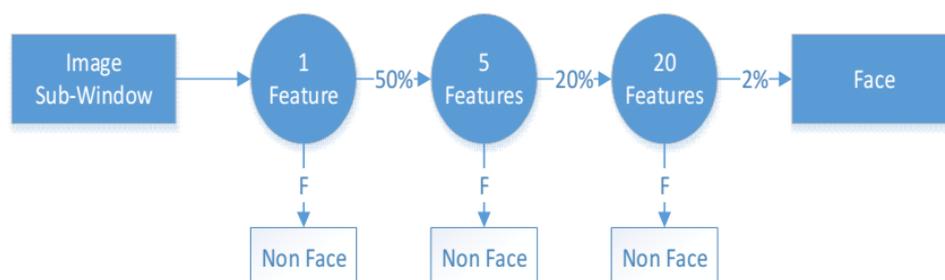
Gambar 1 Fitur Haar (Lienhart, et. al, 2002)



Gambar 2. Perhitungan *integral image* (Viola, et. al, 2001)

Tahap ketiga adalah proses *Adaptive Boosting* atau *AdaBoost*, yaitu mencari fitur-fitur yang memiliki tingkat pembeda yang tinggi. Menurut Viola, et. al, (2011), *AdaBoost* merupakan algoritma pertama yang dapat beradaptasi dengan *weak learner*. Tahap keempat adalah kombinasi *Classifier of Cascade*, dengan cara memusatkan perhatian pada daerah-daerah dalam *image* yang berpeluang saja. Hal ini dilakukan untuk menentukan di mana letak objek yang dicari pada suatu *image*. [7]

Karakteristik dari algoritma Viola Jones adalah adanya klasifikasi bertingkat. Klasifikasi pada algoritma ini terdiri dari tiga tingkatan di mana tiap tingkatan mengeluarkan subimage yang diyakini bukan objek. Hal ini dilakukan karena lebih mudah untuk menilai subimage tersebut bukan objek yang ingin dideteksi ketimbang menilai apakah subimage tersebut merupakan objek yang ingin dideteksi. [2] Alur kerja klasifikasi bertingkat dapat dilihat pada Gambar 3.



Gambar 3. Alur kerja klasifikasi bertingkat (Dzulkamain, et. al, 2011)

Pembelajaran prosedur *AdaBoost* yang disajikan dalam bagian ini, berfungsi untuk meminimalkan kesalahan, dan tidak secara khusus dirancang untuk mencapai deteksi tingkat tinggi dengan mengorbankan besar tingkat kesalahan positif. Skema untuk perkiraan dari kesalahan ini adalah untuk menyesuaikan *threshold* dari *perceptron* yang di produksi oleh *AdaBoost*. Menurut Viola, et. al, (2001) *thresholds* yang lebih tinggi menghasilkan *classifiers* dengan tingkat kesalahan positif yang lebih sedikit, dan tingkat deteksi yang lebih rendah. Sedangkan kebalikannya *thresholds* yang lebih rendah menghasilkan *classifiers* dengan tingkat kesalahan positif yang lebih tinggi.[7]

Salah satu algoritma pengenalan wajah adalah *Eigenface*, dengan prosesnya sebagai berikut:[5]

- a. Langkah pertama adalah menyiapkan data dengan membuat suatu himpunan S yang terdiri dari seluruh *training image* (I_1, I_2, \dots, I_m) yaitu $S = (I_1, I_2, \dots, I_m)$.
- b. Langkah kedua adalah mengambil nilai tengah atau mean (ψ).

$$\Psi = \frac{1}{M} \sum_{n=1}^M I_n \quad (1)$$

- c. Langkah ketiga kemudian mencari selisih (ϕ) antara *training image* (I_i) dengan nilai tengah (ψ).
- d. Langkah keempat adalah menghitung nilai matriks *covariant* (C).

$$C = \frac{1}{M} \sum_{n=1}^M \phi_n \phi_n^T = AA^T \quad (2)$$

$$A = [\phi_1, \phi_2, \dots, \phi_M] \quad L = AA^T \quad (3)$$

$$L = AA^T \quad L = \phi_n \phi_n^T \quad (4)$$

- e. Langkah kelima adalah menghitung *eigenvalue* (λ) dari matriks *covariant*(C).
 $\lambda - L_i = 0 \quad (5)$
- f. Langkah keenam adalah menghitung *eigenvector* (v) dari matriks *covariant* (C).
 $L \cdot v_i = \lambda_i \cdot v_i \quad (6)$
- g. Langkah ketujuh, setelah *eigenvector* (v) diperoleh, maka *eigenface* (μ) dapat dicari dengan :

$$\mu_i = \sum_{k=1}^M v_{ik} \phi_k \quad i = 1, \dots, M \quad (7)$$

Untuk mencari jarak (*distance*) terpendek antara nilai *eigenface* dari *training image* dalam *database* dengan *eigenface* dari *image test face* digunakan rumus *Euclidean Distance* seperti pada persamaan 8.[5]

$$\varepsilon = || \Omega - \Omega_k || \quad (8)$$

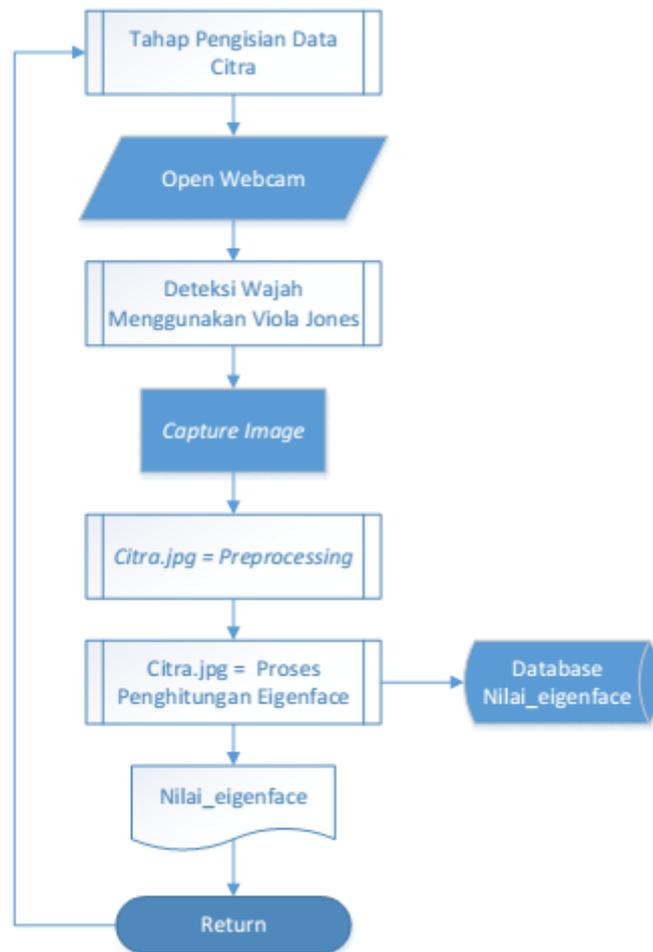
METODE PENELITIAN

Aplikasi yang dibuat dengan menggunakan software Netbeans 8.02 dan bahasa pemrograman Java. Alur proses utama dari sistem terdiri dari dua tahap yaitu pengisian data citra wajah ke dalam database dan proses pengenalan wajah. Hal ini dapat dilihat pada Gambar 4. Sedangkan pengisian data citra wajah ke dalam database dapat dilihat pada Gambar 5. Mulai dari input gambar citra wajah dari kamera *webcam*, pendeteksian wajah pada *window* menggunakan metode Viola Jones, dilakukan secara langsung (*real time*) menggunakan *webcam*. Setelah terdeteksi adanya gambar wajah pada tampilan *window*, maka dilakukan penangkapan citra wajah (*Capture Image*). Tahap berikutnya adalah *preprocessing* yang meliputi proses *resize* untuk membuang bagian selain wajah sehingga hanya bagian wajah saja yang diproses dan normalisasi pencahayaan ketika mengambil citra, RGB ke *grayscale*, *histogram equalization* untuk memperbaiki kualitas *input* citra agar memudahkan proses pengenalan tanpa menghilangkan informasi utamanya. Kemudian dilakukan proses pencarian nilai *eigen* dengan metode *eigenface* sehingga didapatkan *eigenvector* dan *eigenvalue* dari citra tersebut. Setelah dilakukan

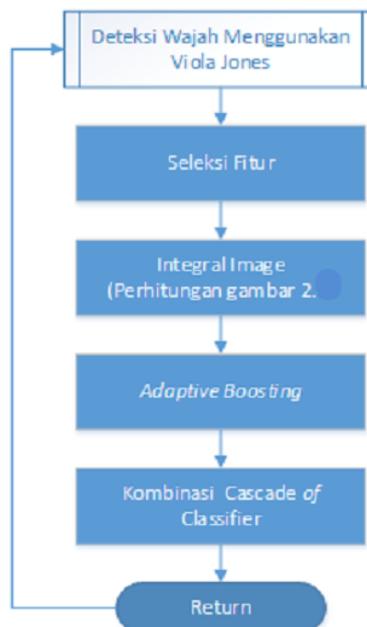
proses penghitungan dengan *eigenface* maka didapatkan hasil nilai dari *eigenface* tersebut kemudian disimpan ke dalam *database*.



Gambar 4. Diagram Alir Sistem



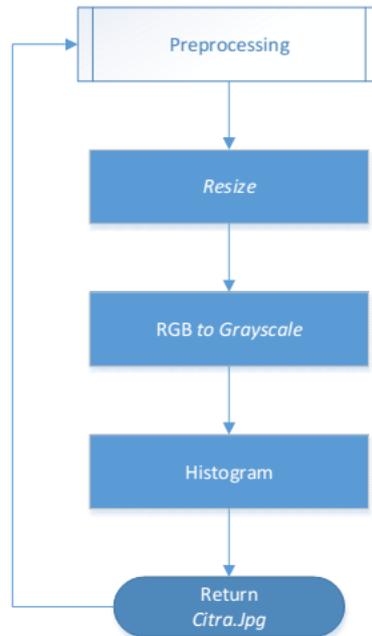
Gambar 5. Diagram Alir Tahap Pengisian Data Citra Wajah
Proses Viola Jones digunakan untuk mendeteksi objek wajah dapat dilihat pada Gambar 6.



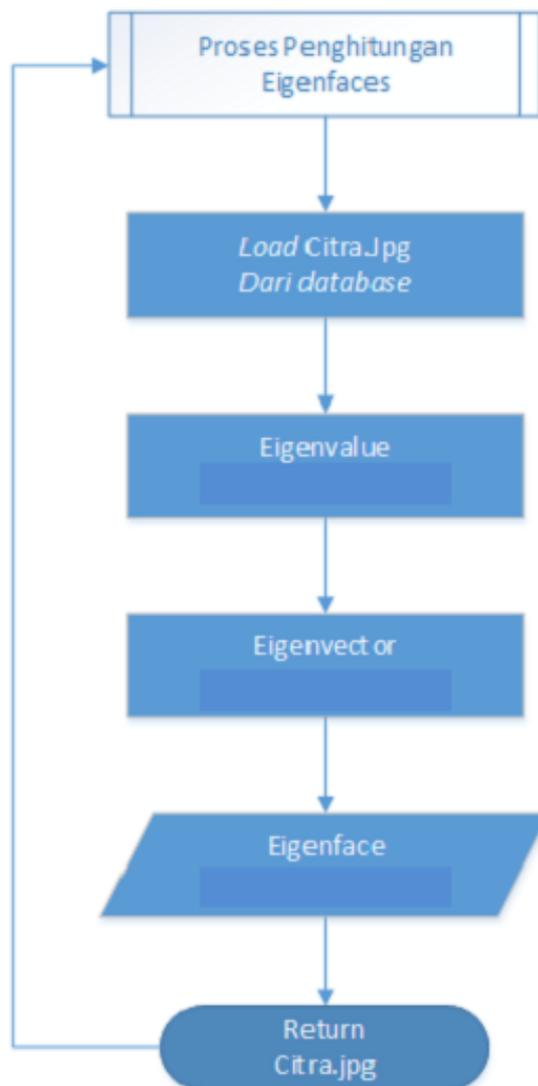
Gambar 6. Diagram Alir Tahap Proses *Viola Jones*

Pada Gambar 6, objek wajah dicari menggunakan Viola Jones dimana citra yang di-*capture* dari *webcam* akan diseleksi fitur kemudian dilakukan proses perhitungan integral *image*. Dengan *Adaptive Boosting* dan kombinasi *Cascade of Classifier* akan mempercepat pendeteksian objek wajah. Jika wajah terdeteksi, akan dilakukan penggambaran garis persegi pada wajah tersebut.

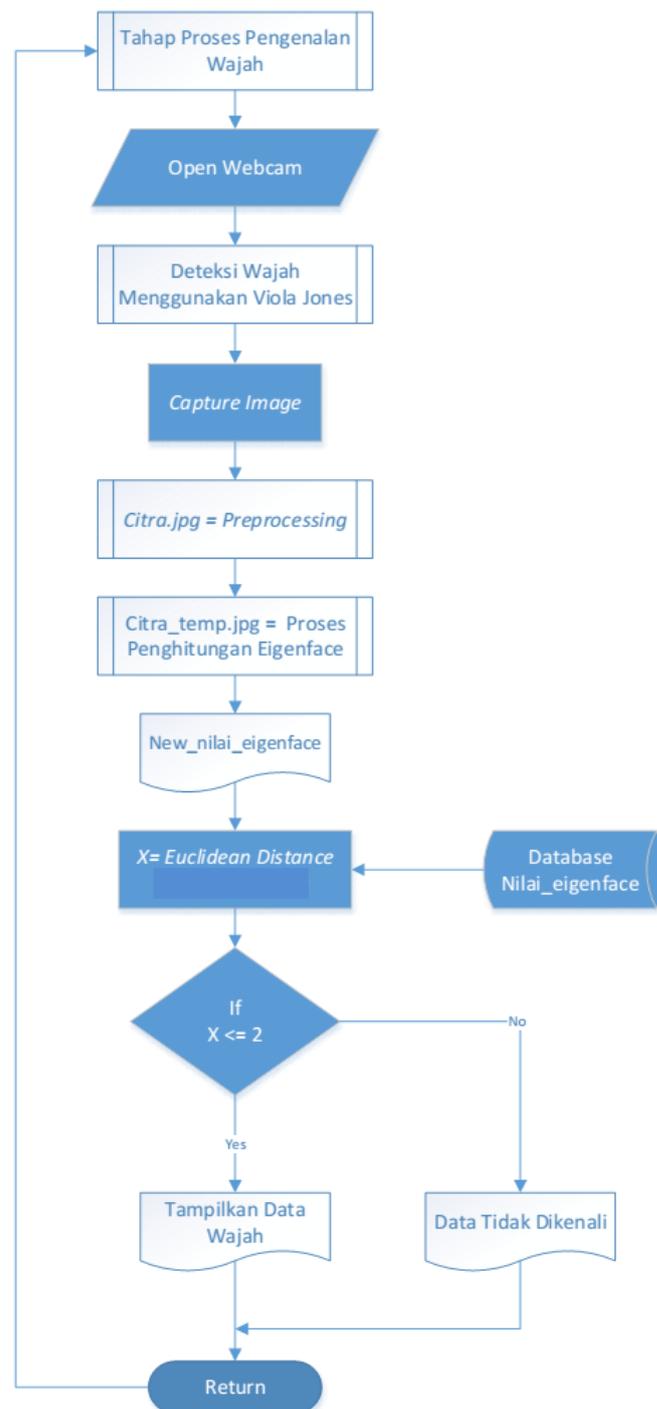
Tahap berikutnya adalah tahap *preprocessing* yang digunakan untuk *resize* atau normalisasi ukuran citra, RGB ke grayscale, *histogram equalization* untuk memperbaiki kualitas citra *input* agar memudahkan proses pengenalan tanpa menghilangkan informasi utamanya. Dari proses tersebut menghasilkan citra wajah yang diambil dalam bentuk *.jpg. Proses *preprocessing* dijelaskan pada Gambar 7.



Gambar 7. Diagram Alir Tahapan *Preprocessing*.



Gambar 8. Diagram Alir Tahapan *Eigenface*



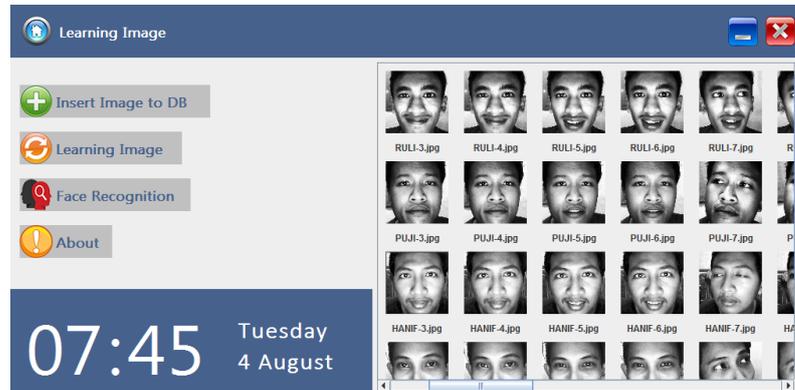
Gambar 9. Diagram Alir Tahap Proses Pengenalan Wajah

Diagram alir pada Gambar 9 menggambarkan tahap proses pengenalan wajah. Pada tahap ini terdapat tiga proses utama. Proses pertama adalah tahap pendeteksian objek dari *scan* wajah menggunakan metode *Viola Jones*. Proses kedua adalah *preprocessing*, tujuannya mempersiapkan gambar untuk mendapatkan citra yang akan diproses dalam penghitungan *Eigenface* dengan optimal. Proses ketiga adalah tahap penghitungan dengan metode *Eigenface*, tujuannya adalah menghitung *eigenvalue*, *eigenvektor*. Selanjutnya dilakukan proses pengenalan dengan mencari jarak terpendek antara nilai *eigenfaces* yang tersimpan dalam nilai *database* dengan nilai *eigenfaces* dari perhitungan secara *realtime* menggunakan *Euclidean Distance*. Jika nilai jarak

(*Euclidean Distance*) kurang dari atau sama dengan dua, maka akan menampilkan data wajah, jika tidak maka wajah tidak dikenali.

HASIL DAN PEMBAHASAN

Pada Gambar 10 adalah tampilan menu *learning image* merupakan tampilan *database* dari *image* yang sudah dimasukkan dari menu sebelumnya. Terdapat 120 *image* wajah dari 6 orang yang tersimpan didalam *database*.



Gambar 10. Menu *Learning Image*

Pada Gambar 11 adalah tampilan menu pengenalan wajah ini merupakan implementasi dari metode *Eigenface*. Citra yang diambil secara *realtime* ini akan dicocokkan dengan nilai citra yang jarak nilainya paling pendek berdasarkan perhitungan dari *Euclidean Distance*.

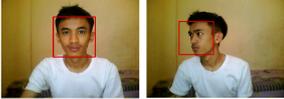
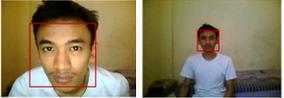


Gambar 11. Menu Pengenalan wajah

Pengujian yang dilakukan ada 2, yaitu:

- a. Pengujian untuk mendeteksi wajah dengan metode Viola Jones
Pengujian dilakukan berdasarkan pencahayaan, jarak, kemiringan, wajah terhalang objek lain, beberapa wajah, dan berdasarkan karakter objek yang menyerupai wajah.

Tabel 1. Pengujian deteksi wajah dengan Viola Jones

No	Gambar	Jenis pengujian	Hasil	Kesimpulan
1		Kemiringan	Terdeteksi	Ketika wajah pada posisi frontal dengan <i>webcam</i> akurasinya adalah 100% dengan waktu deteksi kurang dari 1 detik, sedangkan batas kemiringan maksimum $\pm 70^\circ$
			Tidak terdeteksi	
2		Pencahayaan	Terdeteksi	Pada pencahayaan gelap ataupun terang masih mampu mendeteksi adanya obyek
3		Jarak	Terdeteksi	Jarak terdekat 20cm, pada jarak 200cm masih bisa terdeteksi.
4		Wajah terhalang obyek	Terdeteksi	mampu mendeteksi objek wajah meskipun terhalang selamanya konturnya sama dengan kontur wajah.
			Tidak terdeteksi	
5		Beberapa wajah	Terdeteksi	mampu mendeteksi objek wajah lebih dari satu
6		Berdasarkan karakter yang menyerupai wajah	Terdeteksi	mampu mendeteksi adanya wajah selama konturnya masih menyerupai wajah, terbukti dengan wajah pada tokoh kartun Masha

b. Pengujian aplikasi yang dibuat

Pengujian dilakukan 50 kali percobaan pada sebuah citra yang berukuran 80x80 piksel. Pengujian dilakukan dalam berbagai ekspresi wajah. Dari 50 percobaan hanya ada 5 citra wajah yang tidak dikenali. Sehingga tingkat akurasi sistem sebesar 90%. Hal ini dikarenakan wajah tersebut tidak terdapat didalam database.

KESIMPULAN

1. Akurasi sistem ketika wajah pada posisi frontal dengan *webcam* sebesar 100% dengan waktu deteksi kurang dari 1 detik, sedangkan batas kemiringan maksimum $\pm 70^\circ$ dan jaraknya 20 – 120 cm.

2. Faktor-faktor yang mempengaruhi hasil pengenalan pada sistem ini adalah pencahayaan, jarak *capture* antara objek wajah dan *webcame* serta *distorsi* (kemiringan) wajah pada saat proses *capture* dilakukan.
3. Dari pengujian sebanyak 50 kali didapatkan sebanyak 45 citra wajah berhasil dikenali sesuai dengan *database*. Sedangkan 5 citra wajah tidak berhasil dikenali. Tingkat akurasi sistem sebesar 90%.
4. Wajah yang tidak berhasil dikenali dikarenakan wajah tersebut tidak terdapat dalam *database*.

DAFTAR PUSTAKA

- [1.] Al Fatta, Hanif. 2009. *Rekayasa Sistem Pengenalan Wajah*. Yogyakarta : ANDI.
- [2.] Dzulkamain, A. D., Dewantara, B. S., & Besari, A. A. 2011. *Pengendalian Robot Lengan Beroda Dengan Kamera Untuk Pengambilan Obyek*.
- [3.] Shih, F.Y. 2010. *Image Processing and Pattern Recognition*.
- [4.] Lienhart, R., Kuranove, A., & Pisarevsky, V. 2002. *Emperical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection*.
- [5.] Prasetya, Dedi Ary dan Nurviyanto, Imam. 2012. *Deteksi Wajah Metode Viola Jones Pada OpenCV Menggunakan Pemrograman Python. (Simposium Nasional RAPI XI FT UMS. 18)*
- [6.] Putra, Riyan Syah. 2013. *Perancangan Aplikasi Absensi dengan Deteksi Wajah Menggunakan Metode Eigenface*.
- [7.] Viola & Jones. 2001. *Robust Real-time Object Detection*.

Halaman ini sengaja dikosongkan