

# Aplikasi Screen Lock pada Smartphone Menggunakan Identifikasi Wajah dengan Menerapkan Pointwise

Andi Widiyanto\*<sup>1</sup>, Bintang Dian Mahardika<sup>2</sup>

<sup>1,2</sup>Teknik Informatika Universitas Muhammadiyah Magelang

E-mail: \*<sup>1</sup>[andi.widiyanto@ummgl.ac.id](mailto:andi.widiyanto@ummgl.ac.id), <sup>2</sup>[dekha\\_clickers@gmail.com](mailto:dekha_clickers@gmail.com)

## Abstrak

Penerapan identifikasi wajah (*face recognition*) telah diterapkan pada komputer, laptop atau alat-alat lain yang memang dikhususkan untuk identifikasi wajah. Perkembangan smartphone khususnya android berkembang dengan cepat. Untuk menjaga keamanan supaya hanya dapat digunakan oleh pemilik telah disediakan dengan PIN, phone code, pola geser titik sentuh layar. Aplikasi identifikasi wajah digunakan sebagai pengganti PIN atau code phone pada smartphone android dibutuhkan sebagai proteksi supaya hanya pemiliknya saja yang dapat menggunakannya. Supaya proses identifikasi wajah pemilik lebih mudah perlu dilakukan konversi dari gambar true color ke grayscale proses yang digunakan adalah pointwise. Aplikasi face recognition yang dibangun membutuhkan training wajah pemilik dengan 6 pose wajah yang disimpan, kemudian akan dibandingkan dengan identifikasi wajah saat aplikasi digunakan. Hasil pengujian menunjukkan bahwa tingkat keberhasilan antara 70% - 90%. Jarak antara wajah dan kamera serta tingkat kecerahan cahaya mempengaruhi hasil dari identifikasi wajah. Tingkat keberhasilan identifikasi wajah ditentukan oleh pengambilan image, pemrosesan image, dan perhitungan dengan PCA (*eigenface*).

**Kata Kunci**— *Face recognition, smartphone, android, pointwise, eigenface*

## Abstract

Face recognition has been implemented on a computer, laptop or other device tool which is dedicated for face identification. Developments in particular android smartphones growing rapidly. To maintain the security that can only be used by owners have been provided with a PIN, phone code, pattern shear point touch screen. Face recognition application used as a substitute for or a PIN code on the phone android smartphone needed as protection so only the owner who can use it. So that the process of identification of the owner's face needs to be done easier conversion of true color images into grayscale process used is pointwise. Face recognition application that is built requires owners face training with 6 face pose saved, then will be compared with the face identification when the application is used. The test results showed that the success rate of between 70% - 90%. The distance between the face and the camera and the brightness of light affect the results of face identification. The success rate is determined by identifying the face image capture, image processing, and computation with PCA *eigenface*.

**Keywords**— *Face recognition, smartphone, android, pointwise, eigenface*

## 1. PENDAHULUAN

Sistem operasi *smartphone* yang paling banyak digunakan adalah *android*. *Android* adalah sebuah sistem *open source* yang berkembang dengan pesat begitu pula dengan program aplikasi. Untuk menjaga keamanan supaya hanya dapat digunakan oleh pemilik telah disediakan dengan PIN, *phonecode*, pola geser titik sentuh layar.

Penerapan identifikasi wajah telah diterapkan pada komputer, laptop atau alat-alat lain yang memang dikhususkan untuk identifikasi wajah. Namun saat ini pengguna lebih cenderung menggunakan *handphone* atau *smartphone* daripada menggunakan laptop ataupun komputer dengan berbagai alasan misalnya kepraktisannya.

Pada penelitian ini rumusan masalah yang diangkat adalah bagaimana membangun aplikasi identifikasi wajah dengan pada *smartphone android* sebagai proteksi supaya hanya pemiliknya saja yang dapat menggunakannya.

Aplikasi identifikasi wajah (*face recognition*) digunakan sebagai pengganti PIN atau *code phone* pada *smartphone android* supaya hanya wajah pemilik saja yang dapat digunakan untuk membuka kunci (*lock*) *smartphone* yang terkunci oleh pemiliknya.

Image processing atau pengolahan gambar adalah bidang sendiri yang sudah cukup berkembang sejak orang mengerti bahwa komputer tidak hanya dapat menangani data teks, tetapi juga data gambar. Bidang – bidang yang termasuk dalam image processing meliputi penajaman gambar, penonjolan fitur tertentu dari suatu gambar, komponen gambar dan koneksi gambar yang tidak fokus atau kabur, pewarnaan pada gambar *gray scale* dan sebagainya [1].

Menurut Ridwan (2007) *Point process* atau *pointwise* merupakan salah satu teknik untuk melakukan pengolahan citra digital dengan cara mengubah nilai pixel yang asli berdasarkan nilai pixel tersebut. *Pointwise* juga dapat diterapkan dalam pengolahan citra *truecolor* menjadi *gray scale*[2].

*Gray scale* adalah konversi citra *true color* menjadi citra keabuan. Suatu istilah untuk menyebutkan satu citra yang memiliki warna abu-abu, hitam dan putih. *Gray scale* menunjukkan jumlah warna (dari abu-abu, hingga hitam - putih) yang ada dalam satu citra. Operasi konversi citra *true color* ke *gray scale* dapat Untuk mengubah citra berwarna yang memiliki nilai matrik masing-masing R, G, dan B menjadi citra *gray scale* dengan membagi jumlah ketiga layer, dengan persamaan 1.

$$f_0(x, y) = \frac{f_i^R(x,y) + f_i^G(x,y) + f_i^B(x,y)}{3} \quad (1)$$

Gambar diinput kemudian melalui perulangan untuk x (lebar) dan y (tinggi). x dan y adalah koordinat dari pixel. Kemudian dilanjutkan dengan mencari nilai dari *red*, *green*, *blue* dalam setiap pixel dan dioperasikan dalam penjumlahan kemudian dibagi oleh tiga. Hasilnya akan didapat nilai dari *gray scale* seperti pada gambar 1.



Gambar 1. Perubahan Citra *True Color* (kiri) menjadi citra *gray scale* (kanan)

Secara umum sistem pengenalan wajah dibagi menjadi 2 jenis, yaitu sistem *feature base* dan sistem *image-based*. Pada sistem pertama fitur lokal didapat dari proses ekstraksi komponen citra wajah seperti mulut, hidung, mata, pipi dan lain-lain yang kemudian di modelkan secara geometris hubungan antar fitur-fitur tersebut, sedangkan untuk sistem yang kedua menggunakan informasi mentah dari pixel citra yang kemudian direpresentasikan dalam metode tertentu, misalnya *Principal Component Analysis (PCA)*, transformasi *Wavelet*, dan sebagainya, yang kemudian digunakan untuk pelatihan dan klasifikasi identitas citra [3].

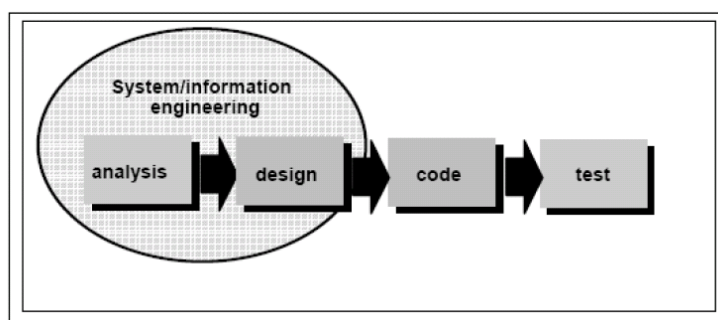
*Eigenface* adalah salah satu algoritma pengenalan wajah yang didasarkan pada *Participal Component Analysis (PCA)* yang dikembangkan Matthew Turk dan Alex Pentland dari *Vision and Modeling Grup*, The Media Laboratory, MIT (*Massachusetts Institute of Tecnology*) pada tahun 1987. Algoritma *eigenface* secara keseluruhan cukup sederhana. *Training image* direpresentasikan ke dalam sebuah *vectorflat* (gabungan vektor) dan digabung secara bersama-sama menjadi sebuah matriks tunggal. *Eigen Vector* kemudian diekstraksi dan disimpan dalam *file temporary* atau *database*. *Training image* kemudian diproyeksikan dalam *feature space*, dinamai *face space* yang ditentukan oleh *eigen vector* [3].

Android adalah sebuah sistem operasi untuk perangkat lunak mobile berbasis *Linux* yang mencakup sistem operasi, *middleware* dan aplikasi. Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman *Java*. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang direlease oleh Google [4].

JavaCV merupakan *wrapper* untuk pustaka-pustaka yang umum digunakan untuk fungsi *real time computer vision* yang terdapat pada aplikasi OpenCV, FFmpeg, *OpenKinect*, dan *ARToolKitPlus*. Dengan menggunakan JavaCV maka pengembangan aplikasi yang menggunakan pustaka yang disebutkan diatas (umumnya *C based*) dapat dilakukan di environment Java [5].

## 2. METODE PENELITIAN

Penelitian ini metodologi penelitian menggunakan Model Sekuensial Linier menurut Roger S. Pressman. Model ini mengembangkan perangkat lunak dengan pendekatan yang sistematis dan sekuensial yang dimulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan bagannya seperti pada gambar 2.



Gambar 2. Model Sekuensial Linier

Aplikasi identifikasi wajah (*face recognition*) yang digunakan sebagai pengganti PIN atau *code phone* pada *smartphone android* yang akan dibangun dengan menerapkan pointwise untuk pengolahan citra adalah pengubah citra RGB menjadi *gray scale*.

Sebuah gambar (*true color*) terdiri dari tiga layer yaitu, *R-layer*, *G-layer*, dan *B-layer* diubah menjadi satu *layer gray scale*. Untuk mengubah citra berwarna yang memiliki nilai matriks masing-masing R, G, dan B menjadi citra *gray scale* dengan membagi jumlah ketiga layer seperti pada persamaan 1.

Proses konversi RGB ke gray scale melalui perhitungan manual, Misalkan sebuah gambar memiliki citra berukuran 3 x 4 pixel dengan nilai - nilai RGB-nya terlihat pada tabel 1.

Tabel 1. RGB dengan pixel 3x4

(x , y)	1	2	3	4
1	60	70	85	90
	63	66	55	86
	30	20	88	73
2	70	86	76	74
	40	75	55	84
	46	61	49	55
3	50	96	55	115
	100	54	100	117
	60	57	61	128

Berikut ini adalah contoh pengolahan RGB ke *gray scale* pada kordinat fo(1,1) dan fo(3,4)

$$f_0(1,1) = \frac{60 + 63 + 30}{3} = 51$$

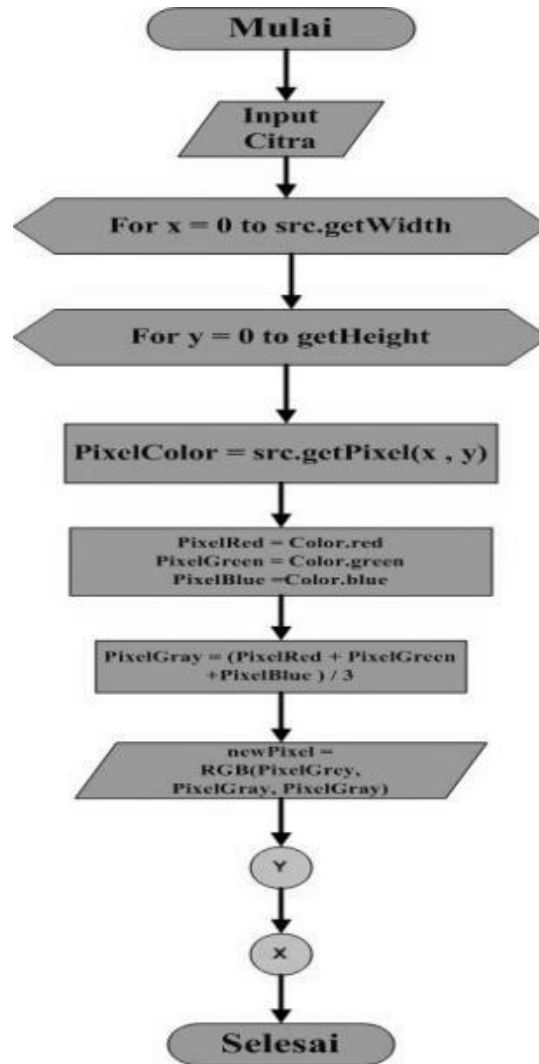
$$f_0(3,4) = \frac{115 + 117 + 128}{3} = 120$$

Perhitungan tersebut dilakukan untuk setiap koordinat, sehingga matriks citra hasil konversi seperti pada tabel 2.

Tabel 2. *Gray scale* dengan pixel 3x4

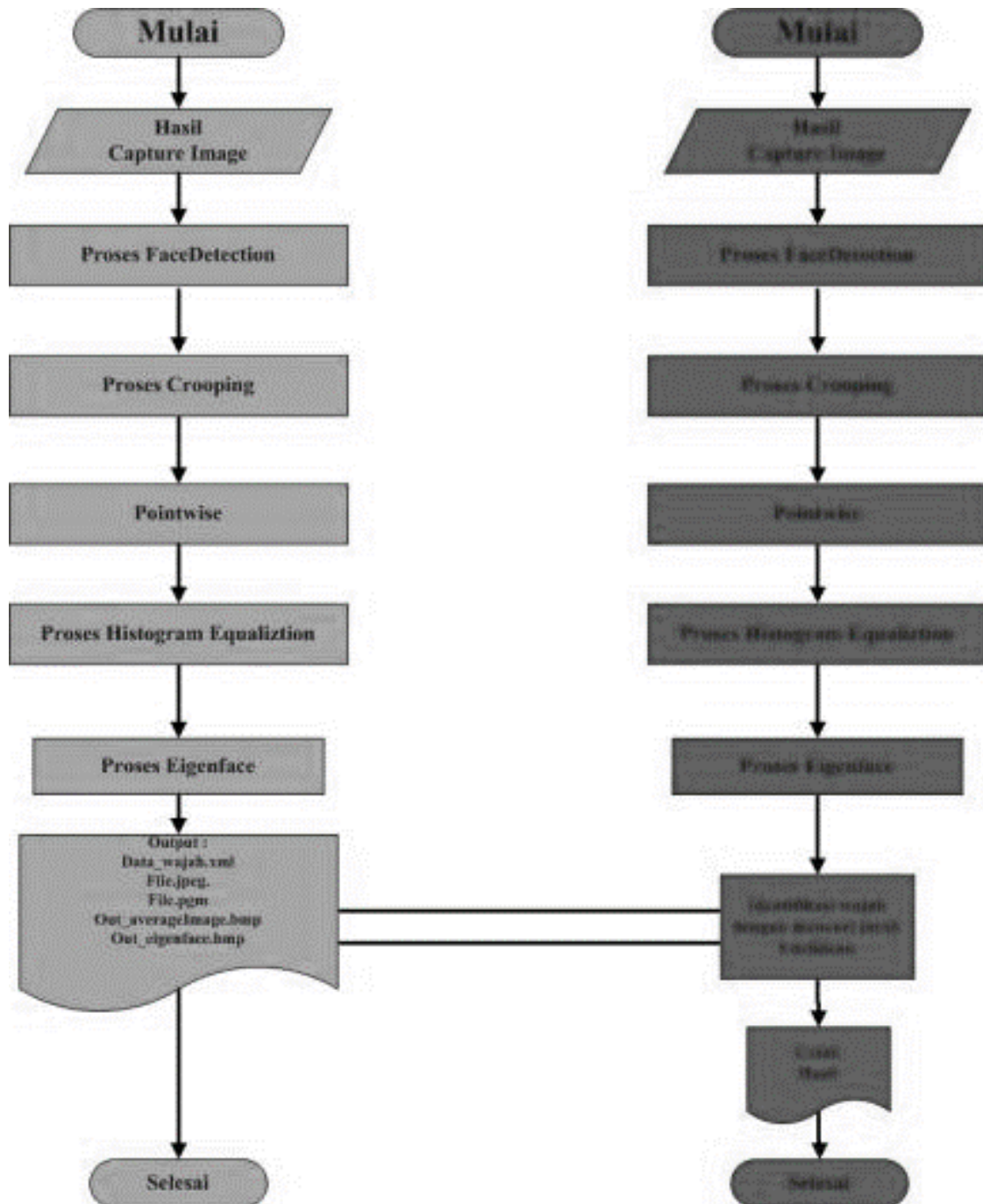
(x , y)	1	2	3	4
1	51	52	76	83
2	52	74	60	71
3	70	69	72	120

Hasilnya akan didapatkan nilai dari *gray scale*. Untuk lebih jelasnya digambarkan dengan *flowchart* gambar 3.



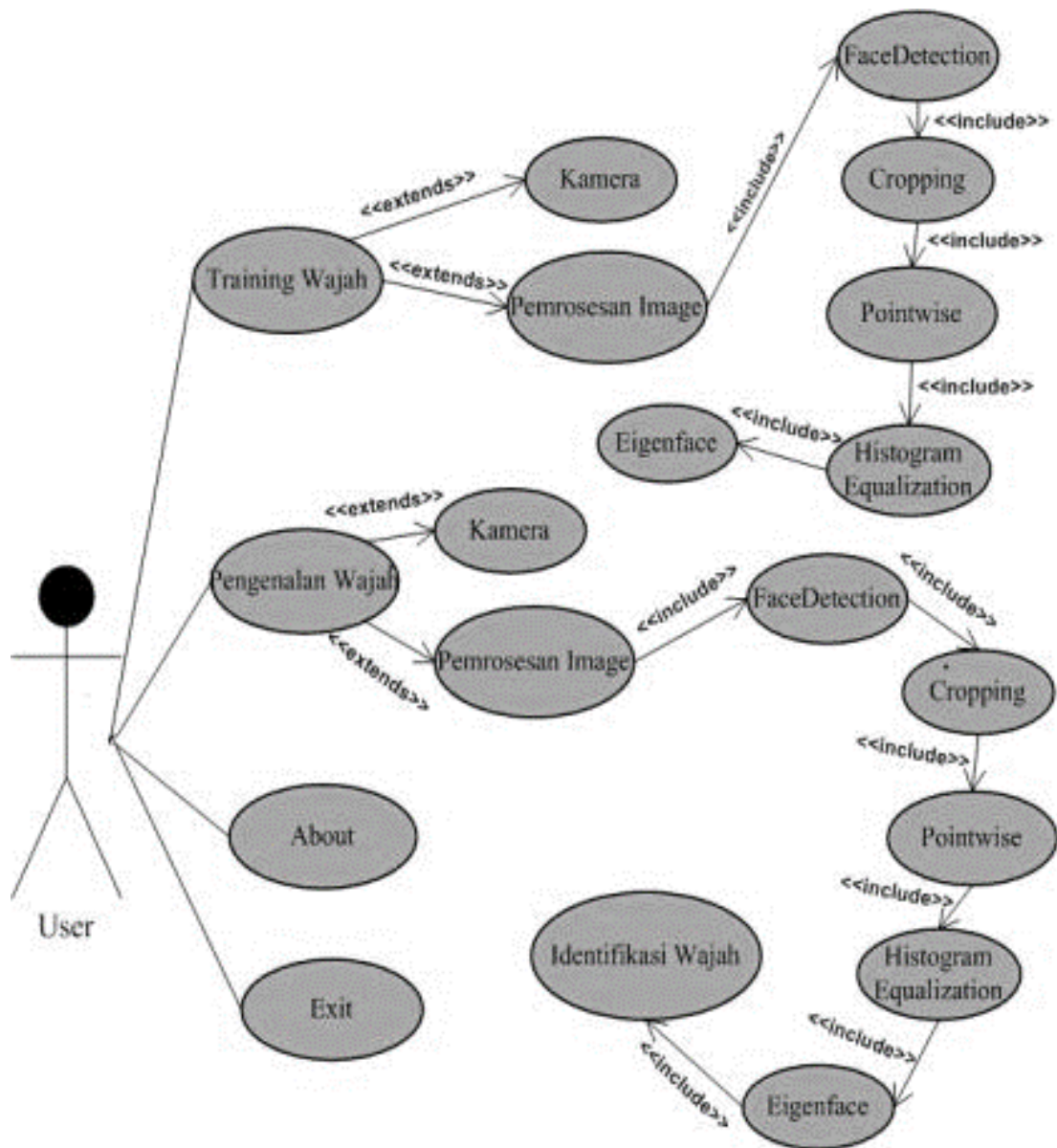
Gambar 3. Flowchart konversi ke *gray scale*

Proses selanjutnya adalah *training* dan identifikasi wajah untuk memproses menggunakan Algoritma *eigenface*. Image training wajah dari hasil *capture* yang telah diolah berupa gambar *gray scale*. Image diproyeksikan ke *facespace* dengan mengalikan di basis *eigenface* proyeksi *vector* wajah akan dibandingkan dengan *vector* yang sesuai. Ekstrak PCA image tes, Cari distance minimum dari image tes dan image training. Penjumlahan dengan hasil terkecil dengan nilai yang tidak bernilai negatif adalah wajah yang cocok dengan wajah tes digambarkan dalam bentuk flowchart yang dapat dilihat pada gambar 4.



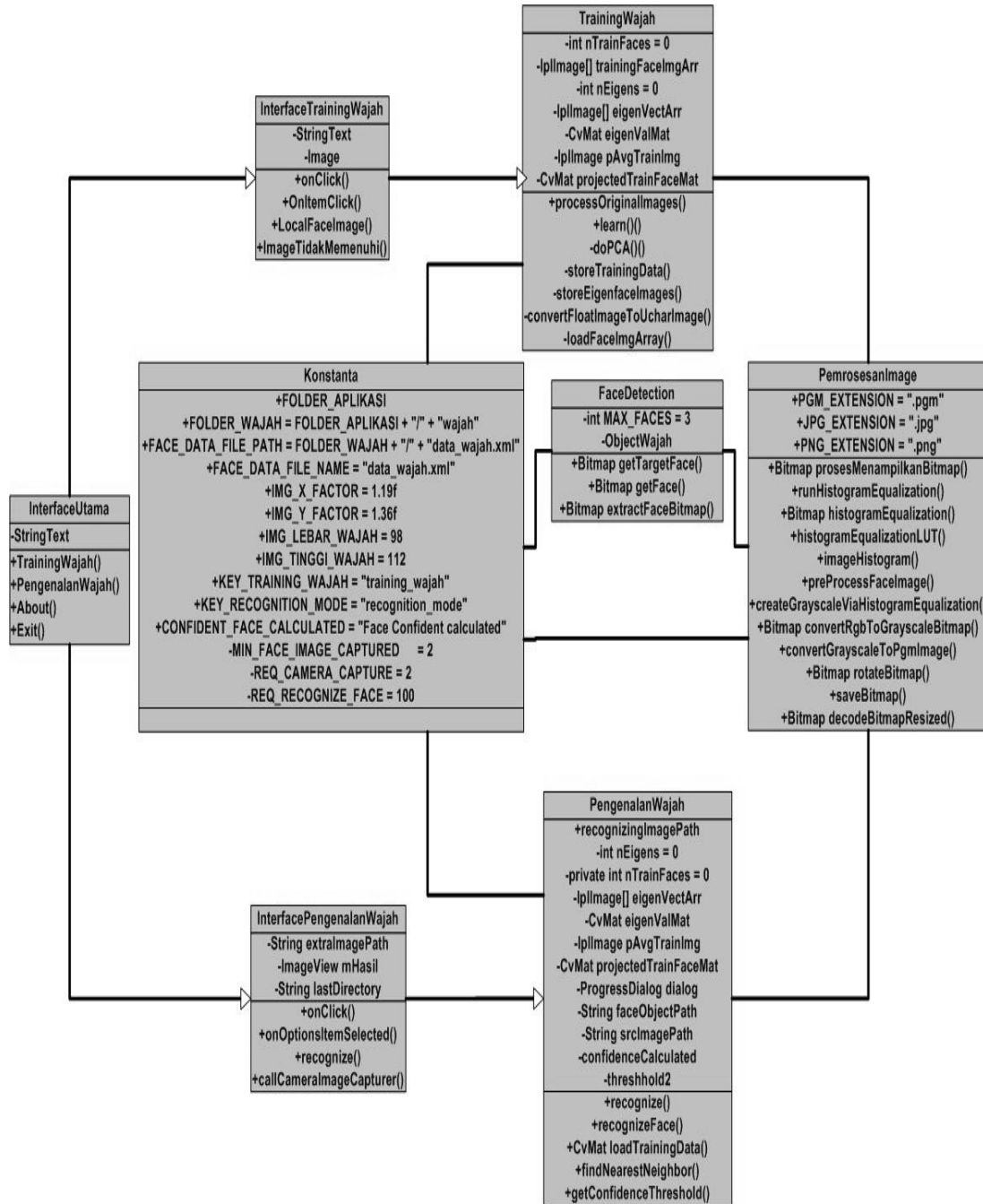
Gambar 4. Model Algoritma *training* wajah (kiri) dan identifikasi wajah (kanan)

Perancangan sistem yang digunakan untuk membangun aplikasi adalah dengan UML (*Unified Modelling Language*). UML adalah bahasa standar yang digunakan untuk menjelaskan dan memvisualisasikan artefak dari proses analisis dan desain berorientasi obyek. Pemodelan aplikasi digambarkan dengan diagram *use case*. Pengembangan aplikasi identifikasi wajah menggunakan metode *eigenfae* hanya terdapat satu actor seperti pada gambar 5.



Gambar. 5 Use case Diagram aplikasi identifikasi wajah

*Class diagram* digunakan untuk menunjukkan hubungan antara *class* dalam aplikasi yang sedang dibangun dan bagaimana setiap *class* saling berkolaborasi untuk mencapai suatu tujuan seperti pada gambar 6.



Gambar 6. Class Diagram aplikasi identifikasi wajah



### 3. HASIL DAN PEMBAHASAN

Rancangan sistem diaplikasikan dengan spesifikasi *hardware* dan *software* sebagai berikut:

1. Prosesor Intel core 2 duo E7500 @ 2.93Ghz
2. RAM 1GB
3. Harddisk 320GB
4. VGA Card Nvidia GeForce 9500GT
5. Monitor
6. Sistem Operasi Windows XP
7. Eclipse versi 3.7.0 sebagai tools pengembangan aplikasi
8. Android SDK versi 20.0.3 sebagai library dari android
9. JDK untuk compiler versi 1.6.
10. Library JavaCV
11. CorelDRAW X4 untuk pembuatan *icon*

Aplikasi *face recognition* yang dihasilkan diinstal pada *smartphone Android* versi 2.3.4 sehingga pada layar *smartphone* akan muncul icon aplikasi seperti pada gambar 7.



Gambar 7. Icon Aplikasi face recognition

Saat aplikasi dijalankan pertama kali akan muncul menu utama terdiri dari *button training*, *button pengenalan*, *button about* dan *button exit* seperti pada gambar 8.



Gambar 8. Menu Utama

*Training* wajah digunakan untuk menyimpan wajah pemilik *smartphone* yang terdiri dari *button* kamera yang digunakan untuk menampilkan fungsi kamera handphone, *button* proses digunakan untuk memproses *image*, *imageview* digunakan untuk menampilkan hasil *image* yang telah diambil, dan *gallery* yang berfungsi menampilkan beberapa gambar yang telah diambil seperti pada gambar 9.

Pemilik *smartphone* mengambil 6 posisi wajah yang berbeda. Hal ini supaya proses *unlock phone* posisi wajah menjadi lebih mudah, tidak terpaku pada posisi tertentu saja akan tetapi dapat melalui beberapa posisi seperti pada gambar 10.



Gambar 9. Tampilan training wajah

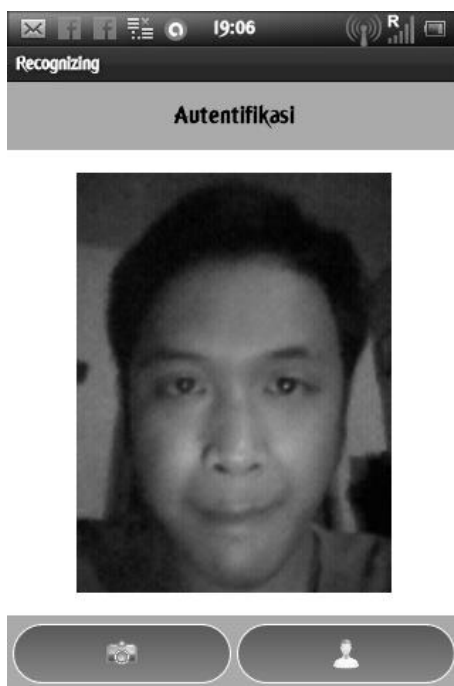


Gambar 10. Proses *training* wajah

Pengenalan Wajah adalah tampilan yang akan muncul pada saat button pengenalan dipilih, atau pada saat *smartphone* dalam keadaan terkunci (*lock*). Menu pengenalan wajah terdapat beberapa komponen antara lain adalah *button* kamera yang digunakan untuk menampilkan fungsi kamera, *button recognition* yang berfungsi untuk memproses pengenalan wajah seperti pada gambar 11 dan gambar 12.

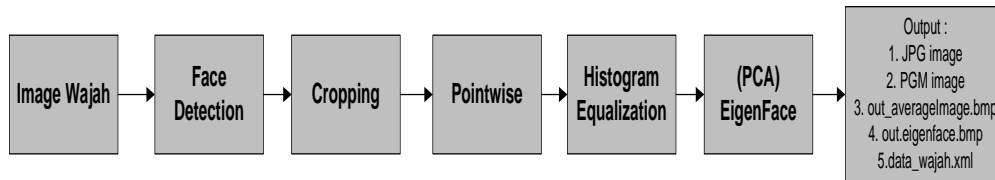


Gambar 11. Tampilan pengenalan wajah

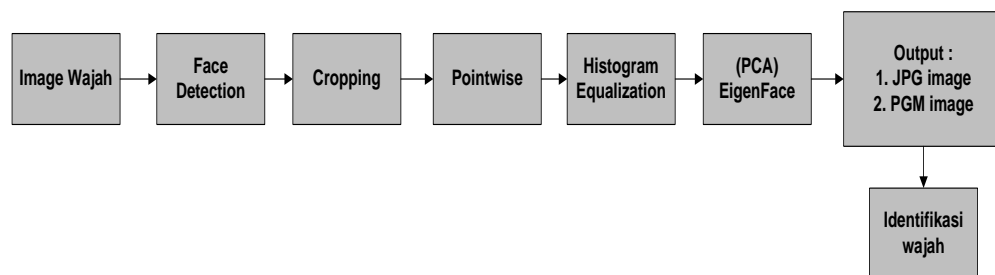


Gambar 12. Proses pengenalan wajah

Penerapan *pointwise* pada penelitian ini digunakan untuk merubah image berwarna (*true color*) menjadi *gray scale*. Proses ini terjadi pada saat proses training dan identifikasi wajah pengguna melalui proses yang sama. Perbedaan proses training wajah dengan identifikasi wajah adalah proses *training* wajah pada penyimpanan image (*data\_wajah.xml*), sedangkan proses identifikasi wajah adalah proses pencocokan data (*face recognition*). Proses training wajah dan *face recognition* untuk lebih jelas dapat digambarkan dengan flowchart pada gambar 13 dan gambar 14.



Gambar 13. Proses Training Wajah



Gambar 14. Proses Identifikasi Wajah

Pengujian aplikasi *face recognition* dilakukan dengan cara pengguna melakukan *training* wajah dengan 6 pose berbeda dengan cahaya normal dan jarak kamera dengan wajah konstan. Posisi wajah saat training dilakukan dengan beberapa kriteria yaitu:

1. Wajah menghadap lurus ke arah kamera.
2. Wajah dengan pose yang unik atau lucu menghadap lurus ke arah kamera.
3. Wajah 5 derajat ke arah samping kanan dari depan kamera.
4. Wajah 5 derajat ke arah samping kiri dari depan kamera.
5. Wajah 5 derajat ke arah atas dari depan kamera.
6. Wajah 5 derajat ke arah bawah dari arah kamera.

Identifikasi wajah dilakukan dengan pengambilan 40 image wajah, dimana 20 adalah wajah pengguna yang memiliki hak masuk dan 20 adalah pengguna yang tidak memiliki hak masuk. Pengujian juga dilakukan dengan tingkat pencahayaan dan jarak yang berbeda agar bisa diambil kesimpulan, hasilnya tercantum pada tabel 3, tabel 4, tabel 5 dan tabel 6.

Tabel 3. Pengujian dengan tingkat cahaya normal dan jarak berbeda

Pengujian Dengan Tingkat Cahaya Normal dan Jarak Berbeda				
Wajah Tes	Banyaknya Input Wajah Tes	Dikenali	Tidak Dikenali	Gagal Saat Identifikasi
Wajah pengguna yang memiliki hak masuk	20	15	3	2
Wajah Pengguna yang tidak memiliki hak masuk	20	3	14	3

Hasil pengujian pada tabel 3 menunjukkan bahwa:

1. Persentase kecocokan antara wajah training dengan wajah yang memiliki hak masuk adalah sebesar  $\frac{15}{20} \times 100\% = 75\%$ .
2. Persentase Keberhasilan ketidakcocokan antara wajah training dengan wajah yang tidak memiliki hak masuk adalah sebesar  $\frac{14}{20} \times 100\% = 70\%$ .

Tabel 4. Pengujian dengan Cahaya Normal dan Jarak Konstan

Pengujian Dengan Tingkat Cahaya Normal dan Jarak Konstan				
Wajah Tes	Banyaknya Input Wajah Tes	Dikenali	Tidak Dikenali	Gagal Saat Identifikasi
Wajah pengguna yang memiliki hak masuk	20	18	1	1
Wajah Pengguna yang tidak memiliki hak masuk	20	1	17	2

Hasil pengujian pada tabel 4 menunjukkan bahwa:

1. Persentase kecocokan antara wajah training dengan wajah yang memiliki hak masuk adalah sebesar  $\frac{18}{20} \times 100\% = 90\%$ .
2. Persentase Keberhasilan ketidakcocokan antara wajah training dengan wajah yang tidak memiliki hak masuk adalah sebesar  $\frac{17}{20} \times 100\% = 85\%$ .

Tabel 5. Pengujian dengan Tingkat Cahaya Terang dan Jarak Konstan

Pengujian Dengan Tingkat Cahaya Terang dan Jarak Konstan				
Wajah Tes	Banyaknya Input Wajah Tes	Dikenali	Tidak Dikenali	Gagal Saat Identifikasi
Wajah pengguna yang memiliki hak masuk	20	14	4	2
Wajah Pengguna yang tidak memiliki hak masuk	20	1	17	2

Hasil pengujian pada tabel 5 menunjukkan bahwa:

1. Persentase kecocokan antara wajah training dengan wajah yang memiliki hak masuk adalah sebesar  $\frac{14}{20} \times 100\% = 70\%$ .
2. Persentase Keberhasilan ketidakcocokan antara wajah training dengan wajah yang tidak memiliki hak masuk adalah sebesar  $\frac{17}{20} \times 100\% = 85\%$ .

Tabel 6. Pengujian dengan Tingkat Cahaya Rendah dan Jarak Konstan

Pengujian Dengan Tingkat Cahaya Rendah dan Jarak Konstan				
Wajah Tes	Banyaknya Input Wajah Tes	Dikenali	Tidak Dikenali	Gagal Saat Identifikasi
Wajah pengguna yang memiliki hak masuk	20	14	4	2
Wajah Pengguna yang tidak memiliki hak masuk	20	1	17	2

Hasil pengujian pada tabel 6 menunjukkan bahwa:

1. Persentase kecocokan antara wajah training dengan wajah yang memiliki hak masuk adalah sebesar  $\frac{14}{20} \times 100\% = 70\%$ .
2. Persentase Keberhasilan ketidakcocokan antara wajah training dengan wajah yang tidak memiliki hak masuk adalah sebesar  $\frac{17}{20} \times 100\% = 85\%$ .

#### 4. KESIMPULAN

Berdasarkan Implementasi dan penjelasan yang telah dipaparkan sebelumnya, maka dapat diambil kesimpulan dari aplikasi identifikasi wajah menggunakan metode eigenface berbasis android sebagai berikut:

1. Aplikasi ini dapat mengidentifikasi wajah menggunakan metode eigenface berbasis android yang diterapkan pada face unlock sebagai autentifikasi user untuk menggantikan PIN dan password dengan tingkat keberhasilan antara 70% sampai 90%.
2. Jarak antara wajah dan kamera serta tingkat kecerahan cahaya mempengaruhi hasil dari identifikasi wajah. Tingkat keberhasilannya ditentukan oleh pengambilan image, pemrosesan image, dan perhitungan dengan PCA (*eigenface*).

#### 5. SARAN

Aplikasi *face recognition* akan lebih optimal jika digunakan pada *smartphone* yang menggunakan kamera statis dan kondisi ruangan dengan tingkat pencahayaan maupun jarak stabil.

#### DAFTAR PUSTAKA

- [1] Ahmad, U., 2005, *Pengolahan Citra Digital dan Teknik Pemrogramannya*, Edisi 1, Graha Ilmu, Yogyakarta.
- [2] Ridwan, F., Mengenal Lebih Jauh Apa Itu Point Process, <http://ilmukomputer.org/2011/03/14/mengenal-lebih-jauh-apa-itu-point-process>, diakses tanggal 5 Desember 2012.
- [3] Fatta, H. A., 2009, Pengenalan Wajah dengan Menggunakan Algoritma Eigenface, *Jurnal Penelitian*.
- [4] Siregar, I. M., Yusuf, R., Siendow, W., Wino, W. W., 2010, *Mengembangkan Aplikasi Enterprise Berbasis Android*, Gava Media, Yogyakarta.
- [5] Rifqi, Instalasi JavaCV, <http://rifqithokz.wordpress.com>, diakses 2 Januari 2013.