



UNIVERSITAS INDONESIA

**IMPLEMENTASI SISTEM PENGENALAN WAJAH SEBAGAI
PENGHUBUNG JEJARING SOSIAL:**

**PENERAPAN *AUGMENTED REALITY* SEBAGAI PENAMPIL
INFORMASI HASIL PENGENALAN WAJAH PADA
PERANGKAT ANDROID**

SKRIPSI

**Slamet Budiayatno
0806459904**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK KOMPUTER
DEPOK
JULI 2012**



UNIVERSITAS INDONESIA

**IMPLEMENTASI SISTEM PENGENALAN WAJAH SEBAGAI
PENGHUBUNG JEJARING SOSIAL:
PENERAPAN *AUGMENTED REALITY* SEBAGAI PENAMPIL
INFORMASI HASIL PENGENALAN WAJAH PADA
PERANGKAT ANDROID**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana

**Slamet Budiayatno
0806459904**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK KOMPUTER
DEPOK
JULI 2012**

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.

Nama : Slamet Budiayatno
NPM : 0806459904
Tanda Tangan : 
Tanggal : 2 Juli 2012



HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Slamet Budiayatno
NPM : 0806459904
Program Studi : Teknik Komputer
Judul : Implementasi Sistem Pengenalan Wajah Sebagai Penghubung Jejaring Sosial: Penerapan *Augmented Reality* Sebagai Penampil Informasi Hasil Pengenalan Wajah Pada Perangkat Android

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Komputer, Fakultas Teknik, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing : Prof. Dr. Ir. Riri Fitri Sari M.M., M.Sc.

()

Penguji : Dr. Ir. Dodi Sudiana M.Eng.

()

Penguji : Yan Maraden ST. MSc.

()

Ditetapkan di : Depok

Tanggal : 2 Juli 2012

UCAPAN TERIMA KASIH

Puji dan syukur penulis panjatkan kehadiran Allah SWT atas segala nikmat dan karunianya sehingga penulis dapat menyelesaikan skripsi ini.

Selain itu penulis ingin mengucapkan terima kasih juga kepada:

1. Ibu Riri Fitri Sari selaku dosen pembimbing, yang telah memberikan banyak waktu, petunjuk, saran dan bimbingan dalam penulisan skripsi maupun selama masa studi di Teknik Komputer;
2. Kedua orang tua saya, serta kakak-kakak yang senantiasa memberikan doa, semangat dan motivasi;
3. Teman setim (Nur Muhammad Ridho dan Prasetyawidi Indrawan) serta teman-teman Teknik Elektro dan Teknik Komputer angkatan 2008 yang setia memberi dukungan dan menjalin kerjasamanya.
4. Teman-teman yang telah bersedia menjadi relawan untuk diambil wajahnya sebagai pengujian.

Demikian yang bisa penulis sampaikan. Jika ada kekurangan dan kesalahan dalam penulisan skripsi ini, mohon dimaafkan. Penulis mengharapkan kritik dan saran yang membangun dalam melengkapi kekurangan yang ada pada skripsi ini.

Depok, 2 Juli 2012

Penulis,



Slamet Budiayatno

NPM. 0806459904

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademika Universitas Indonesia, saya yang bertanda tangan dibawah ini :

Nama : Slamet Budiayatno

NPM : 0806459904

Program : Teknik Komputer

Departemen : Teknik Elektro

Fakultas : Teknik

Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Non-eksklusif (Non-exclusive Royalty Free Right)** atas karya ilmiah saya yang berjudul :

IMPLEMENTASI SISTEM PENGENALAN WAJAH SEBAGAI
PENGHUBUNG JEJARING SOSIAL:

**PENERAPAN *AUGMENTED REALITY* SEBAGAI PENAMPIL
INFORMASI HASIL PENGENALAN WAJAH PADA PERANGKAT
ANDROID**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non Eksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan tugas akhir saya selama mencantumkan nama saya sebagai penulis/pencipta sebagai pemegang Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 2 Juli 2012

Yang menyatakan



Slamet Budiayatno

ABSTRAK

Nama : Slamet Budiayatno
Program Studi : Teknik Komputer
Judul : Implementasi Sistem Pengenalan Wajah Sebagai Penghubung Jejaring Sosial: Penerapan *Augmented Reality* Sebagai Penampil Informasi Hasil Pengenalan Wajah Pada Perangkat Android.

Sejak kehadiran jejaring sosial belakangan ini setiap orang dapat dengan mudah memperoleh informasi siapapun. Bahkan persaingan industri IT semakin marak dengan dikembangkannya aplikasi dan layanan yang terhubung dengan jejaring sosial. Oleh karena itu, muncul sebuah ide untuk mengembangkan sistem pengenalan wajah sebagai identitas penghubung jejaring sosial. Selain menggunakan wajah sebagai identitas utama dalam perancangan sistem, wajah juga digunakan sebagai marker untuk menampilkan informasi hasil pengenalan wajah berbasis *Augmented Reality*. Sistem utuh ini terdiri dari pengenalan wajah pada perangkat mobile Android, pengenalan wajah pada layanan komputasi awan dan tambahan informasi hasil pengenalan wajah berupa *Augmented Reality*. Modul pengenalan wajah ditanamkan pada layanan *Cloud Computing Google App Engine* berbasis *Python* dengan memanfaatkan *Face.com API* sebagai pengolahan citra wajah. Hasil informasi dari layanan tersebut dikembalikan dalam format JSON. Response JSON itu dimanfaatkan sebagai tambahan informasi yang akan ditampilkan dengan konsep *Augmented Reality*. Kehadiran *Augmented Reality* pada sistem ini bertujuan untuk memberikan interaksi yang ramah dengan pengguna. Berdasarkan hasil pengujian, *Augmented Reality* bekerja dengan cepat ketika menjadikan wajah sebagai marker untuk menampilkan informasi hasil pengenalan wajah, dengan respon rata-rata sebesar 1025.42 ms untuk mendapat informasi lengkap dan 697.7 ms untuk mendapat sedikit informasi dari orang yang dikenal.

Kata kunci:

Jejaring Sosial, Pengenalan Wajah, Cloud, Face.com API, dan Augmented Reality

ABSTRACT

Name : Slamet Budiayatno
Study Program : Computer Engineering
Title : Design of Face Recognition System as an Identity Connector between Social Networking Applications : The Application of Augmented Reality as an information viewer for Face Recognition System with OpenCV on Google App Engine.

Recently, since the presence of social networking, anyone can easily receive information of anyone, anytime and anywhere. The competition for IT industry increased with the development of connected applications and services with social networking. Therefore, we proposed a face recognition system as a connector to social networking application. In addition to using face as a primary identity in the system design, face is also used as a marker to display information of the result of the face recognition-based on Augmented Reality. This complete system consists of face recognition on Android mobile devices. Face recognition on cloud computing services and additional information on the results of face recognition in the form of Augmented Reality. Face recognition module is embedded in the Cloud Computing using Google App Engine services based on Python, and also using Face.com API for facial image processing. The results of the service information is returned in JSON format. Given JSON response used as additional information to be displayed with the concept of Augmented Reality. The presence of Augmented Reality in this system aims to provide a friendly interaction with the user. Based on the results of test, Augmented Reality works quickly when used faces as a marker, with the average response time of 1025.42 ms to get complete information and 697.7 ms to get a little information from people who are known.

Keyword

Social networking, face recognition, cloud computing, face.com API and augmented reality

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN	iii
UCAPAN TERIMA KASIH	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penulisan.....	3
1.4 Metodologi Penelitian	3
1.5 Sistematika Penulisan.....	4
BAB 2 Sistem Pengenalan Wajah Berbasis <i>Cloud</i> dan <i>Augmented Reality</i> pada Perangkat Android.	6
2.1 Konsep Sistem Pengenalan Wajah.....	6
2.1.1 Wajah	6
2.1.2 Klasifikasi Wajah	6
2.1.3 Sistem Pengenalan Wajah	7
2.2 Teknologi <i>Augmented Reality</i>	10
2.2.1 Jenis-jenis <i>Augmented Reality</i>	12
2.2.2 Perangkat Keras Teknologi <i>Augmented Reality</i>	13
2.3 Cloud Computing.....	14
2.3.1 Latar Belakang Munculnya Cloud Computing.....	15
2.3.2 Karakteristik Cloud Computing	15
2.3.3 Layanan Cloud Computing	16
2.3.4 Penggunaan Google App Engine sebagai PaaS	17
2.4 Platform Android Sebagai Media Pengembangan Sistem.....	18
2.5 Facebook API Sebagai Media Penghubung Jejaring Sosial.....	22
BAB 3 Perancangan Sistem Pengenalan Wajah dan Tampilan Informasi Berbasis <i>Augmented Reality</i>.....	24
3.1 Deskripsi Umum Sistem.....	24
3.2 Alur Kerja Sistem	26
3.3 Diagram Unified Modelling Language (UML)	27
3.3.1 Use Case Diagram.....	28

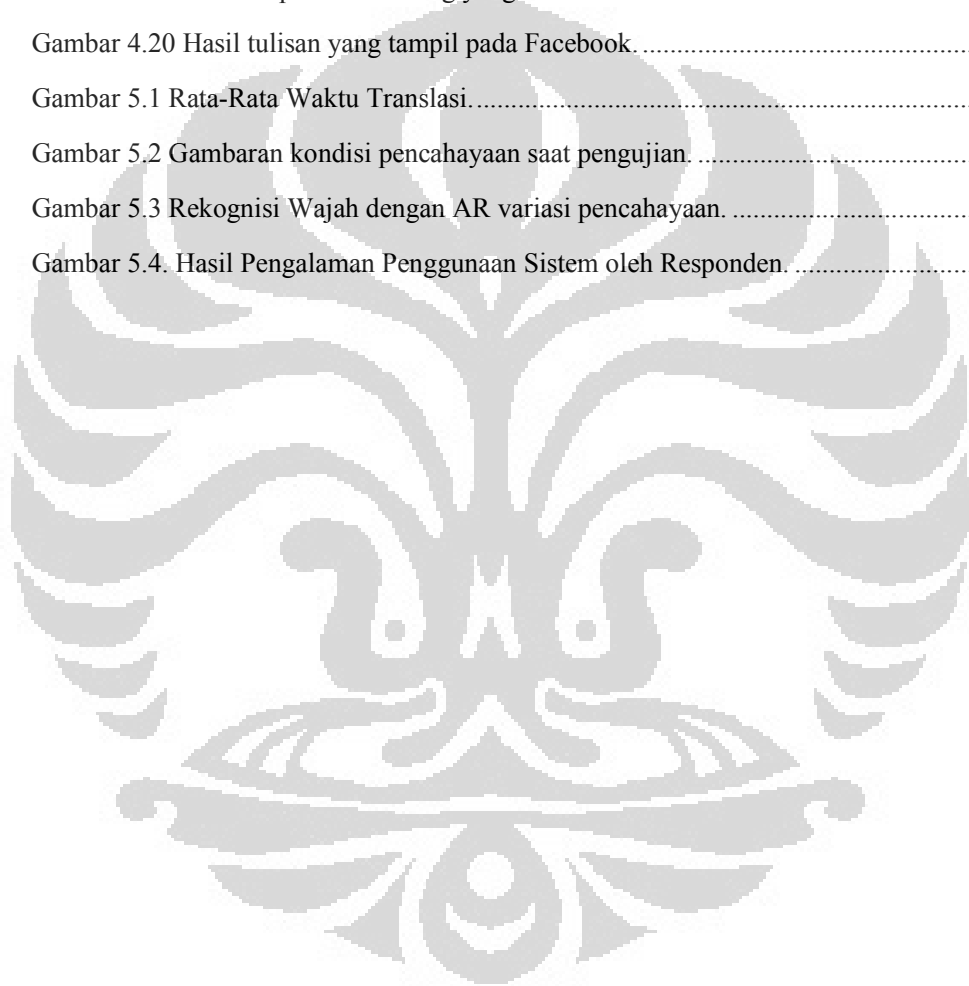
3.3.2 Activity Diagram	29
3.3.3 Sequence Diagram	31
3.3.4 Deployment Diagram	32
BAB 4 Implementasi Modul <i>Augmented Reality</i> dan Integrasi Pada Sistem Pengenalan Wajah	33
4.1 Perancangan Modul <i>Augmented Reality</i>	33
4.1.1 Perangkat Pendukung Pengembangan.....	33
4.1.2 Perancangan Modul <i>Augmented Reality</i> Sistem Terpisah	34
4.1.3 Pemrograman Modul <i>Augmented Reality</i>	34
4.2 Integrasi Modul <i>Augmented Reality</i> Pada Sistem Utuh.	42
BAB 5 Pengujian dan Analisa Modul <i>Augmented Reality</i> Pada Proses Pengenalan Wajah	47
5.1 Pengujian dan Analisis <i>Translation Time</i> pada <i>Augmented Reality</i>	47
5.2 Pengujian dan Analisa Performa <i>Augmented Reality</i> berdasarkan Tingkat Pencahayaan.....	55
5.3 Pengujian Kualitatif.....	58
BAB 6 PENUTUP	61
DAFTAR REFERENSI.....	63



DAFTAR GAMBAR

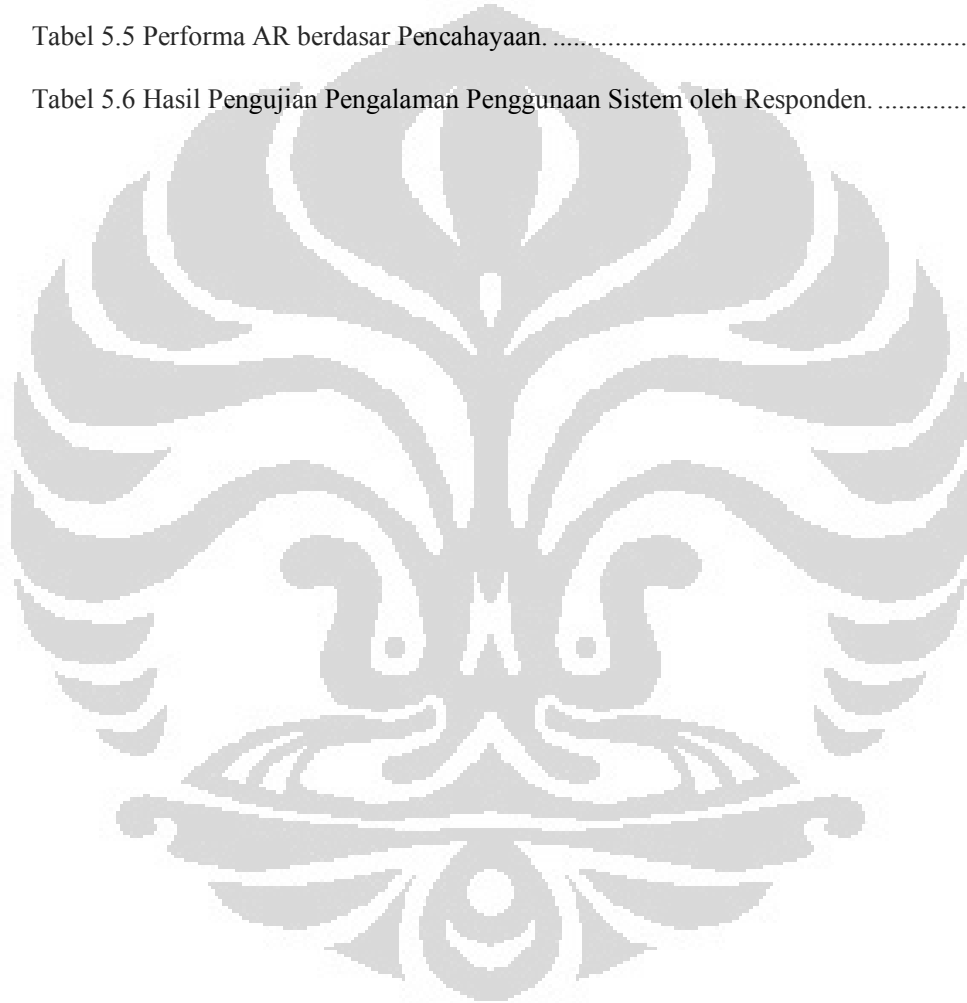
Gambar 2.1	Ilustrasi <i>Face Recognition</i>	7
Gambar 2.2	<i>AR Building Block</i>	11
Gambar 2.3	Augmented Reality dengan Marker.....	12
Gambar 2.4	<i>Markerless AR</i> pada <i>face tracking</i>	12
Gambar 2.5	Perangkat bersama untuk aplikasi <i>Augmented Reality</i>	13
Gambar 2.6	<i>Cloud Computing</i>	14
Gambar 2.7	Distribusi Platform Android.....	19
Gambar 2.8	Arsitektur sistem operasi Android.....	20
Gambar 3.1	Arsitektur Sistem Pengenalan Wajah.....	25
Gambar 3.2	Diagram Alir Umum Sistem Pengenalan Wajah berbasis Cloud.....	26
Gambar 3.3	Use Case Diagram Registrasi Pengguna.....	28
Gambar 3.4	Use Case Diagram Pengguna Terdaftar.....	29
Gambar 3.5	<i>Activity Diaram</i>	30
Gambar 3.6	Sequence Diagram.....	31
Gambar 3.7	Deployment Diagram.....	32
Gambar 4.1	Algoritma Modul <i>Pre-processing</i>	34
Gambar 4.2	Tampilan modul <i>Augmented Reality</i> terpisah.....	35
Gambar 4.3	Fungsi Deteksi Wajah.....	37
Gambar 4.4	<i>Layer</i> pada <i>Augmented Reality</i>	38
Gambar 4.5	Kode untuk pengaturan tampilan Kamera.....	38
Gambar 4.5	Kode kelas <i>PreviewView</i>	39
Gambar 4.6	Kode pembuatan <i>Overlay</i> pada Android.....	39
Gambar 4.7	Kode untuk pilihan <i>mode</i>	40
Gambar 4.8	Kode untuk meletakkan gambar di kanvas.....	41
Gambar 4.9	Antarmuka dari tiap lapisan <i>Augmented Reality</i>	42
Gambar 4.10	Penggambaran objek pada <i>canvas</i>	43
Gambar 4.11	Kode penentuan posisi objek pada <i>canvas</i>	44
Gambar 4.12	Kelas <i>onTouchEvent</i> untuk aksi aktivitas rekognisi wajah.....	45

Gambar 4.13 Antarmuka Sistem Pengenalan Wajah berbasis <i>Augmented Realit</i>	46
Gambar 4.14 Contoh respon JSON hasil rekognisi dengan <i>Facebook</i>	46
Gambar 4.15 Respon JSON hasil translasi Graph API.	47
Gambar 4.16 Proses koneksi dengan Graph API.	47
Gambar 4.17 Pengambilan beberapa informasi.....	48
Gambar 4.18 Informasi lengkap hasil pengenalan wajah.....	48
Gambar 4.19 Menulis pada wall orang yang dikenal.....	49
Gambar 4.20 Hasil tulisan yang tampil pada Facebook.....	49
Gambar 5.1 Rata-Rata Waktu Translasi.....	48
Gambar 5.2 Gambaran kondisi pencahayaan saat pengujian.....	53
Gambar 5.3 Rekognisi Wajah dengan AR variasi pencahayaan.....	55
Gambar 5.4. Hasil Pengalaman Penggunaan Sistem oleh Responden.....	59



DAFTAR TABEL

Tabel 5.1 Waktu Translasi 9 akses informasi.....	51
Tabel 5.2 Waktu Translasi 1 akses informasi.....	52
Tabel 5.3 Rataan Waktu Translasi JSON.....	52
Tabel 5.4 Hasil Pengujian Variasi Pencahayaan pada Sistem Pengenalan Wajah.....	56
Tabel 5.5 Performa AR berdasar Pencahayaan.....	56
Tabel 5.6 Hasil Pengujian Pengalaman Penggunaan Sistem oleh Responden.....	58



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dewasa ini tidak bisa dipungkiri masyarakat Indonesia maupun dunia sedang menikmati fenomena jejaring sosial. Sebuah fenomena yang mampu menyentuh berbagai kalangan, karena keberadaannya memberikan kesempatan kepada siapapun untuk mengenal lebih banyak orang di semua belahan dunia. Tidak hanya itu, sejumlah informasi pribadi seseorang pun bisa diperoleh melalui jejaring sosial.

Seiring pesatnya perkembangan teknologi, hadirilah sistem pengenalan wajah (*face recognition*) yang juga semakin banyak dimanfaatkan pada perangkat *mobile*. Melalui sistem pengenalan wajah tersebut, identitas diri seseorang dapat diketahui dengan mudah hanya dengan memanfaatkan sistem kamera yang terdapat pada perangkat *mobile* yang diarahkan ke wajah seseorang tak dikenal. Sistem pengenalan wajah bukanlah hal yang baru dalam ranah komputasi citra atau pemrosesan wajah. Pada bidang pemrosesan wajah (*face processing*), pendeteksian wajah manusia (*face detection*) adalah salah satu tahap awal yang sangat penting dalam proses pengenalan wajah (*face recognition*). Sistem pengenalan wajah digunakan untuk membandingkan satu citra wajah masukan dengan suatu *database* wajah dan menghasilkan wajah yang paling cocok dengan citra tersebut. Hanya saja kelemahan yang sering terjadi pada penelitian pengenalan wajah yang telah ada sebelumnya [1] adalah sistem yang berjalan tidak *real-time* [1], ekspresi wajah yang harus sama, serta proses komputasi yang terlalu berat [2].

Berawal dari proses komputasi yang berat, diperlukan terobosan baru yang mampu menyederhanakan permasalahan tersebut. Salah satunya dengan memanfaatkan layanan komputasi awan (*cloud computing*). Sangatlah cocok apabila dikombinasikan antara sistem pengenalan wajah dan *cloud computing* dikarenakan keduanya memiliki relasi yang jelas, dimana *cloud computing* dapat

menjadi solusi atas kendala yang dihadapi terkait proses komputasi yang terjadi pada sistem pengenalan wajah.

Cloud Computing [3] (komputasi awan) menawarkan suatu konsumsi dan model *deployment* pada sebuah layanan teknologi informasi berbasis internet. Dengan model ini, suatu *resource* akan digunakan secara bersama-sama melalui jaringan internet sebagai suatu layanan bagi para pengguna. Dengan model komputasi awan, pengguna jasa layanan tidak lagi direpotkan dengan media penyimpanan maupun proses pengolahan data karena semuanya telah diatur oleh server luar (*cloud*). Pembelian dan perawatan mesin *server* dapat dihindari karena seluruhnya menjadi tanggung jawab layanan yang disediakan penyedia jasa.

Hal inilah yang menjadi latar belakang perancangan sistem pengenalan wajah berbasis *cloud computing*. Kemudian menengok pada kondisi bahwa sistem pengenalan wajah yang telah ada belum berjalan dengan *real-time*, penerapan *Augmented Reality* sebagai penampil informasi identitas diri dapat menjadi solusi yang ramah untuk menyesuaikan kondisi *real time* pada sistem pengenalan wajah.

Augmented Reality (AR) merupakan suatu lingkungan yang memasukkan objek *virtual* kedalam lingkungan nyata. Oleh karena itu, unsur *reality* lebih diutamakan sehingga interaksi secara *real-time* pada sebuah sistem dapat memudahkan *end-user* dalam penggunaannya. Dengan demikian pemanfaatan teknologi AR pada sistem pengenalan wajah berbasis *cloud computing* dapat memberikan informasi yang lebih menarik dan memberikan warna tersendiri dalam perkembangan teknologi.

1.2 Rumusan Masalah

Permasalahan yang menjadi perhatian dalam utama pada tulisan ini adalah perancangan dan implementasi sebuah sistem pengenalan wajah pada perangkat *mobile* memanfaatkan layanan komputasi awan sebagai media pemrosesan, hasil pengenalan wajah tersebut ditampilkan dalam bentuk *augmented reality*. Lanjutan atas tampilnya informasi wajah yang dikenal yakni terhubungnya jejaring sosial (*facebook*) pemilik wajah. Melalui jejaring sosial tersebut informasi mengenai data diri dapat diperoleh lebih lengkap.

1.3 Tujuan Penulisan

Tujuan dari penulisan penelitian ini adalah merancang sebuah sistem pengenalan wajah yang terdiri dari sistem pengolahan citra berbasis komputasi awan dan sistem penampil informasi dengan konsep *augmented reality* yang secara langsung akan menjadi penghubung dengan jejaring sosial. Sistem ini ditujukan untuk membantu mengenali wajah, mendapatkan informasi terkait wajah yang sesuai pada pemrosesan yang ditampilkan dalam bentuk *augmented reality* dan menjadi penghubung jejaring sosial.

1.4 Metode Penelitian

Metode penelitian yang digunakan dalam penulisan skripsi ini antara lain :

1. Studi Literatur

Pada tahap ini dilakukan pendalaman literatur yang berhubungan dengan pembuatan aplikasi android, dasar dari teknologi *augmented reality*, dasar sistem pengenalan wajah, penggunaan *augmented reality* pada android, koneksi aplikasi android dengan jejaring sosial (*facebook*) dan berbagai literatur lain yang mendukung pembuatan sistem.

2. Perancangan Perangkat Lunak.

Perancangan perangkat lunak dilakukan dan digambarkan dalam *Unified Modelling Language* (UML) sebagai bahasa standar pemodelan rancangan. Pendahuluan tahap ini dengan menyediakan *requirement* pendukung dalam pengembangan *augmented reality* pada android seperti perangkat *mobile* android, *Software Development Kit* (SDK) dan *Native Development Kit* (NDK). Pada proses pengembangan, dilakukan perancangan model tampilan *augmented reality* memanfaatkan wajah sebagai marker melalui pendeteksian wajah. Selanjutnya, menampilkan realitas tertambah dengan tanda berupa masker yang menutupi area wajah saat kamera android diarahkan pada wajah seseorang.

3. Integrasi Keseluruhan Perangkat Lunak.

Berdasarkan perancangan yang telah dibuat, dilakukan integrasi dengan sistem keseluruhan yaitu pengenalan wajah, layanan komputasi awan dan jejaring

sosial. *Augmented Reality* yang diterapkan pada sistem ini memanfaatkan hasil pengenalan wajah berupa format JSON yang dilayani *cloud* dan ditampilkan pada layar perangkat Android.

4. Pengujian dan Analisa Perangkat Lunak.

Pada tahap ini dilakukan pengujian terhadap sistem apakah sudah sesuai dengan rancangan awal yang telah dibuat sebelumnya atau tidak. Jika terdapat hal yang tidak sesuai akan dilakukan analisa kesalahan dan selanjutnya akan ditindak lanjuti dengan melakukan pembenaran pada sistem dengan mengulangi proses pengerjaan pada tahap tertentu sesuai dengan kesalahan yang terjadi.

5. Penulisan Skripsi

Tahap ini merupakan tahap terakhir yaitu melakukan penulisan dan detail tentang sistem yang telah dibuat.

1.5 Sistematika Penulisan

Sistematika dari penulisan skripsi ini meliputi beberapa bab, yaitu :

Bab 1 Pendahuluan

Bab ini berisi Latar Belakang, Rumusan Masalah, Tujuan Penulisan, Metode Penelitian, dan Sistematika Penulisan.

Bab 2 *Augmented Reality* dan Sistem Pengenalan wajah berbasis *Cloud*.

Bab ini memberikan pengantar mengenai berbagai teori yang digunakan dalam perancangan sistem, yang terdiri dari bahasan mengenai Pendeteksian dan Pengenalan Wajah, *Augmented Reality* dan Perkembangannya, *Platform* Android, dan Sistem Kerja Komputasi Awan (*Cloud Computing*).

Bab 3 Perancangan Sistem Pengenalan Wajah dan Tampilan Informasi berbasis *Augmented Reality*

Bab ini membahas deskripsi umum sistem, rancang bangun salah satu bagian dari sistem (*Augmented Reality* sebagai penampil informasi), Diagram Alir Program dan Diagram *Unified Modelling Language* (UML).

Bab 4 Implementasi Modul *Augmented Reality* dan Integrasi Pada Sistem Pengenalan Wajah

Bab ini membahas mengenai implementasi modul *augmented reality* dalam sistem tunggal dan integrasi modul *augmented reality* dalam sistem pengenalan wajah keseluruhan. Bahasan implementasi modul *augmented reality* mencakup pada rancang bangun pada perangkat android.

Bab 5 Pengujian dan Analisis Modul *Augmented Reality* pada Proses Pengenalan Wajah

Berdasarkan hasil implementasi yang dilakukan, tahapan selanjutnya adalah pengujian dengan hasil yang dibahas pada Bab ini. Pengujian dan analisis dilakukan melalui hasil pengujian teknis dengan berbagai variabel meliputi waktu translasi dan kondisi pencahayaan serta pengujian kualitatif yang dilakukan beberapa responden.

Bab 6 Penutup

Bab ini berisi penutup dari penulisan skripsi, yang berisi kesimpulan dan saran.

BAB 2

SISTEM PENGENALAN WAJAH BERBASIS *CLOUD* DAN *AUGMENTED REALITY* PADA PERANGKAT ANDROID

2.1 Konsep Sistem Pengenalan Wajah

Sistem pengenalan wajah merupakan konsep pengembangan ilmu yang berada dalam kategori biometrik. Biometrik adalah salah satu teknik pengenalan pola kebenaran identitas seseorang dengan menggunakan fisiologi atau perilaku manusia melalui teknologi komputer seperti pengenalan sidik jari, telapak tangan, wajah, iris dan lain-lain. Sistem pengenalan wajah menjadi salah satu diantaranya yang bertujuan untuk melakukan identifikasi wajah dengan membandingkan wajah dengan *database* wajah yang telah ada.

2.1.1 Wajah

Wajah adalah bagian terpenting dari siapa pun untuk dapat mengenali, kecuali pada kasus kembar identik. Wajah tergolong sebagai karakteristik fisik paling unik yang dimiliki setiap orang. Sementara itu manusia memiliki kemampuan bawaan untuk mengenali dan membedakan wajah yang berbeda, sedangkan teknologi komputer saat ini sedang mengujarnya untuk dapat melakukan hal serupa melalui perancangan sistem pengenalan wajah.

2.1.2 Klasifikasi Wajah

Pada sebuah sistem yang berkaitan dengan pengenalan wajah, ada beberapa hal penting yang perlu diperhatikan yakni klasifikasi wajah. Terdapat tiga jenis hasil klasifikasi wajah [4] sebagai informasi hasil deteksi sistem :

- Bukan Wajah.

Klasifikasi jenis ini menyatakan bahwa sistem melakukan proses deteksi terhadap suatu objek. Proses deteksi tersebut ditemukan sebagai sebuah pola yang memberikan informasi bahwa hasil deteksi bukan sebuah wajah, melainkan hanya objek biasa.

- Wajah yang tidak diketahui.

Klasifikasi ini menunjukkan bahwa proses pengujian sistem memberikan informasi hasil deteksi merupakan wajah seseorang. Namun pencocokan wajah hasil deteksi tidak sesuai dengan *image* (wajah) yang terdapat pada sistem. Hasil penafsiran tersebut digolongkan sebagai wajah yang tidak diketahui.

- Wajah yang dikenal

Sedikit berbeda dengan sebelumnya, klasifikasi ini menemukan bahwa hasil deteksi wajah diketahui sebagai wajah yang dikenal atau mirip dengan beberapa wajah yang terdapat pada sistem.

2.1.3 Sistem Pengenalan Wajah

Face Recognition (pengenalan wajah) merupakan riset yang sangat digemari dalam lingkup *Computer Vision* dan *Biometrics* [5]. Aplikasinya kian terasa dalam kehidupan sosial serta banyak dimanfaatkan dalam bidang robotika.

Face Recognition umumnya melibatkan dua tahap [5,6]:

1. *Face Detection* (deteksi wajah), dimana *image* yang diperoleh diminta untuk menemukan wajah (ditampilkan di sini sebagai persegi hijau), maka pengolahan citra membersihkan gambar wajah dan mengabaikan daerah disekitarnya.
2. *Face Recognition* (pengenalan wajah), dimana wajah yang dideteksi kemudian diproses dan dibandingkan dengan *database* wajah-wajah yang dikenal, untuk memutuskan siapa orang tersebut (ditampilkan di sini sebagai teks merah).



Gambar 2.1 Ilustrasi *Face Recognition* [5]

Sebagai tahap awal yang sangat penting dalam proses pengenalan wajah, proses deteksi akan dihadapkan oleh tantangan-tantangan [7] dari berbagai faktor yaitu :

- Posisi wajah. Posisi wajah pada citra dapat bervariasi karena perbedaan posisi bisa tegak, miring, melihat kesamping atau sebagainya.
- Komponen wajah yang tak terduga (bisa ada atau tidak ada) seperti kumis, jenggot dan kacamata.
- Ekspresi wajah. Tampilan wajah sangat dipengaruhi oleh ekspresi seperti tersenyum, tertawa, marah, sedih dan sebagainya.
- Citra wajah terhalang oleh objek lain misalnya pada citra yang berisi sekumpulan orang.
- Kondisi pengambilan citra wajah yang dipengaruhi oleh lingkungan seperti intensitas cahaya, arah sumber cahaya bahkan lensa kamera.

Melihat banyaknya tantangan yang muncul saat proses deteksi wajah, dilakukan sebuah riset [7] yang mengelompokkan metode deteksi wajah menjadi empat kategori antara lain :

1. Knowledge-base method.

Metode ini menggunakan aturan dasar yang umumnya digunakan manusia untuk menentukan apa saja yang membentuk wajah. Sebagai contoh, wajah seseorang pada sebuah citra biasanya memiliki dua buah mata yang simetris, sebuah hidung, dan mulut. Relasi antar organ tersebut dijadikan sebagai sebuah tolak ukur jarak atau posisi.

Masalah yang muncul pada metode ini adalah kesulitan menerjemahkan pengetahuan manusia kedalam aturan yang dipakai. Jika aturan terlalu detail, maka akan sering gagal mendeteksi wajah untuk menyesuaikan aturan tersebut. Jika aturan terlalu umum, akan menghasilkan terlalu banyak *false positive* (salah perkiraan). Kesulitan semakin bertambah jika proses deteksi dilakukan pada posisi wajah yang bervariasi. Oleh karena itu, metode ini umumnya dapat bekerja dengan baik pada wajah frontal dan tegak.

2. *Feature invariant approach*

Metode ini bertujuan untuk menemukan fitur wajah yang tidak berubah (*invariant*) meskipun terdapat variasi pose, sudut pandang dan kondisi cahaya. Fitur tersebut seperti alis, mata, hidung, mulut yang biasanya diekstraksi dengan *edge detector*. Setelah itu dibentuk suatu model matematis yang mendeskripsikan hubungan antar fitur untuk menentukan ada tidaknya wajah. Warna kulit juga dapat membantu memperkirakan wajah, namun perlu dikombinasikan dengan metode lain seperti *shape analysis* dan *motion information*.

3. *Template matching method*

Metode ini akan melakukan penyimpanan pola wajah standar untuk mendeskripsikan wajah secara keseluruhan maupun bagian-bagiannya. Saat pendeteksian, akan dihitung korelasi antara citra input dengan citra pola wajah yang tersimpan sebelumnya.

4. *Appearance-based method*

Pada metode ini, citra wajah dipelajari melalui proses *training* menggunakan satu set data pelatihan yang berisi contoh-contoh wajah. Kemudian hasil *training* akan digunakan untuk mendeteksi wajah. Secara umum metode ini menggunakan teknik analisa statistik dan *machine learning* untuk menemukan karakteristik wajah yang relevan dengan citra wajah inputan. Proses ini dikenal dengan nama metode *eigenfaces*. Pada perancangan sistem pengenalan wajah berbasis *cloud* ini akan digunakan metode tersebut.

Dengan demikian sistem pengenalan wajah merupakan bagian akhir dari sistem deteksi dan pengenalan wajah. Pada sistem deteksi wajah dilakukan proses pengolahan citra dan deteksi wajah. Sistem deteksi bertugas mengolah citra dan menyesuaikan dengan *database* dan hanya melewatkan citra yang terdapat wajah didalamnya, serta melakukan klasifikasi wajah dan non-wajah terhadap citra masukan. Sedangkan sistem pengenalan wajah bertugas mengenali citra masukan sebagai wajah orang yang ada dalam *database* atau tidak. Citra yang merupakan

wajah dalam *database* akan ditentukan sebagai wajah ke-n. Wajah ke-n tersebut akan melalui proses pengenalan yang terdiri dari dua bagian, yaitu bagian ekstraksi fitur dan klasifikasi (pengenalan). Bagian ekstraksi fitur bertujuan mereduksi data sehingga dihasilkan fitur penting yang mewakili citra wajah, pada bagian klasifikasi akan mengenali fitur wajah tersebut sebagai orang yang dikenali (sesuai *database*) atau tidak.

2.2 Teknologi *Augmented Reality*

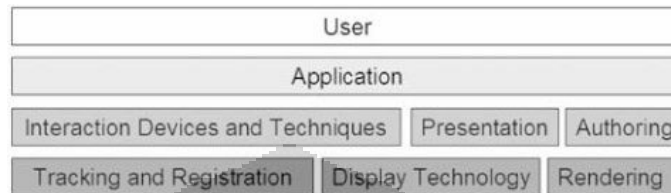
Augmented Reality (AR) merupakan sebuah konsep yang menggabungkan antara objek dunia nyata dan objek dunia maya yang dihasilkan dari sebuah sistem komputer dengan menambah informasi pada objek nyata sehingga batas antara keduanya menjadi sangat tipis. Sistem ini lebih dekat kepada lingkungan nyata, karena itu *reality* lebih diutamakan pada sistem ini. Dengan bantuan teknologi AR, lingkungan nyata disekitar kita akan dapat berinteraksi dengan bentuk digital (*virtual*). Informasi tentang objek dan lingkungan sekitar kita dapat ditambahkan kedalam sistem AR yang kemudian informasi tersebut ditampilkan diatas lapisan dunia nyata secara *real-time* sehingga seolah-olah informasi tersebut menjadi interaktif dan nyata. Konsep AR pertama kali diperkenalkan oleh Thomas Caudell pada tahun 1990 saat ia bekerja di perusahaan Boeing [8].

Ada tiga karakteristik yang menyatakan suatu teknologi menerapkan konsep *Augmented Reality* [9] yaitu :

1. Mampu mengkombinasikan dunia nyata dan dunia maya.
2. Mampu memberikan informasi secara interaktif dan *real-time*.
3. Mampu menampilkan dalam bentuk 3D (tiga dimensi)

Hal tersulit dalam perancangan sistem berbasis *Augmented Reality* adalah membuat seolah-olah tidak ada perbedaan antara lingkungan dunia nyata dan dunia maya (*virtual*), sehingga secara kasat mata keduanya menjadi kesatuan utuh.

Terdapat tiga unsur mendasar dalam pengembangan sistem *Augmented Reality* [10], seperti terlihat dalam *building block* berikut.



Gambar 2.2 AR *Building Block* [10]

Unsur pertama adalah *registration and tracking*. Hal mendasar yang perlu diperhatikan adalah bagaimana menggabungkan dunia nyata dengan objek dunia maya yang akan ditanamkan kedalam dunia nyata tersebut. Inilah yang dimaksud dengan *registration*. Proses penggabungan tersebut diawali oleh respon sistem dalam menangkap sebuah kondisi untuk menggabungkan dunia maya pada dunia nyata.

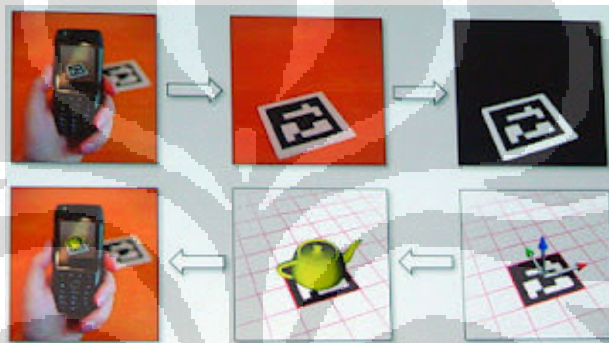
Unsur penting mendasar kedua adalah *display technology*. Teknologi penampil merupakan sebuah alat untuk merepresentasikan antarmuka secara langsung antara pengguna dengan aplikasi yang menerapkan sistem AR.

Unsur terakhir adalah *rendering*. Proses ini terkait bagaimana menggabungkan kondisi nyata dengan objek buatan yang dihasilkan oleh komputer menjadi satu kesatuan utuh. Untuk menghasilkan tampilan AR yang realistis, diperlukan proses *rendering* yang bersifat *real-time*, sehingga apa yang dilihat oleh pengguna mendekati kondisi nyata. Pada perancangan dan implementasi sistem, *augmented reality* digunakan untuk menampilkan informasi hasil pengenalan wajah yang tertangkap oleh kamera pengguna saat sistem dijalankan.

2.2.1 Jenis-jenis Augmented Reality

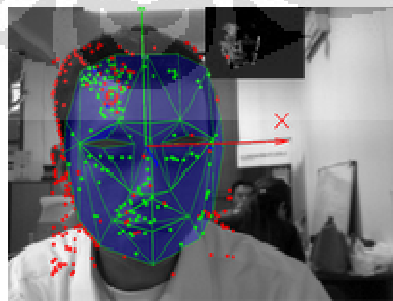
Augmented Reality (AR) terbagi dua macam berdasarkan metode penggunaannya yaitu *Marker Augmented Reality* dan *Markerless Augmented Reality*.

- *Marker Augmented Reality* merupakan sebuah metode yang memanfaatkan marker berupa ilustrasi hitam dan putih berbentuk persegi atau lainnya dengan batas hitam tebal dan latar belakang putih. Melalui posisi yang dihadapkan pada sebuah kamera komputer, maka komputer akan melakukan proses menciptakan dunia *virtual* 2D atau 3D.



Gambar 2.3 *Augmented Reality* dengan Marker [22]

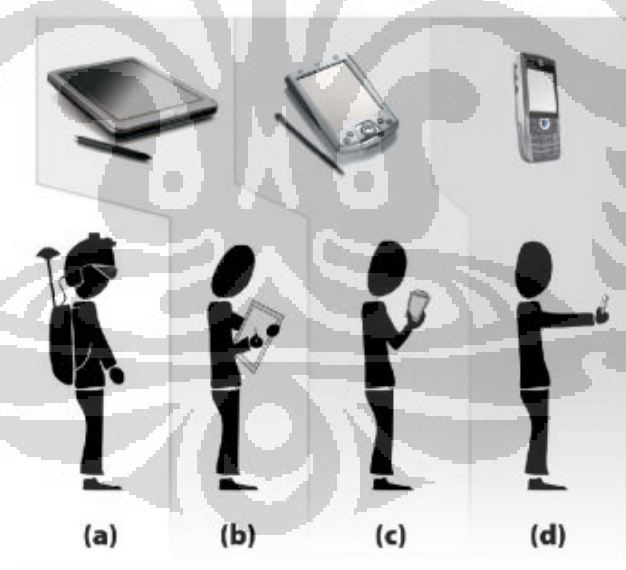
- *Markerless Augmented Reality* yaitu metode AR yang tidak lagi memerlukan marker untuk menampilkan elemen dunia maya (*virtual*) saat menggabungkan dengan lingkungan dunia nyata. Penggunaan metode *markerless* biasa dimanfaatkan untuk *face tracking*, *3D object tracking* dan *motion tracking*. Pada sistem ini diterapkan metode *markerless augmented reality* dengan memanfaatkan deteksi wajah sebagai *trigger* untuk menampilkan informasi pemilik wajah.



Gambar 2.4 *Markerless AR* pada *face tracking*

2.2.2 Perangkat Keras untuk Teknologi Augmented Reality

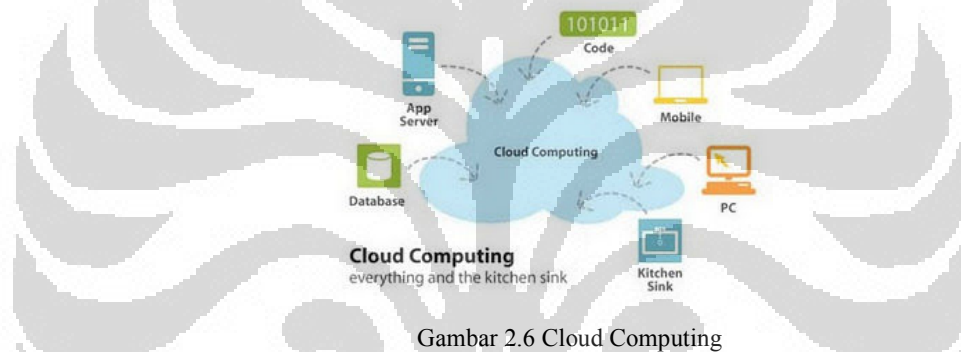
Augmented Reality sebagai teknologi yang mampu memberikan informasi atau gambaran tambahan secara virtual mengenai objek fisik pada dunia nyata sehingga terbentuk kondisi *real-life*, memerlukan sebuah perangkat keras untuk melayani penerapannya. Secara mendasar ada tiga hal terpenting yang dibutuhkan dalam teknologi *Augmented Reality*. Pertama, kamera untuk mengambil gambar dari lingkungan nyata, *display* untuk menampilkan hasil akhir, dan terakhir, perangkat untuk menyediakan proses komputasi. Terdapat berbagai macam perangkat yang memenuhi kebutuhan tersebut. Gambar 2.5 menunjukkan perangkat yang dibutuhkan pada aplikasi *Augmented Reality* [24]. Perangkat tersebut dari kiri ke kanan yakni PC + *Head Mounted Display* (HMD), komputer tablet, PDA, dan *smartphone*. Mereka bukan hanya berbeda dari segi ukuran melainkan juga dari kehebatan CPU yang disediakan. Pada skripsi ini, *augmented reality* lebih difokuskan pada pengembangan di perangkat android.



Gambar 2.5 Perangkat bersama untuk aplikasi *Augmented Reality*
(a). PC + HMD, (b) komputer tablet, (c) PDA, dan (d) *Smartphone*

2.3 Cloud Computing

Proses komputasi yang berkembang pada saat ini terdiri dari beberapa jenis diantaranya yaitu *mobile computing* (komputasi bergerak), *grid computing* (komputasi grid) dan terakhir *cloud computing* (komputasi awan). *Cloud computing* atau komputasi awan merupakan gaya komputasi yang terbilang dinamis yang memberdayakan layanan internet sebagai sumber daya *virtual*. *Cloud computing* digunakan untuk memindahkan layanan yang digunakan sehari-hari ke internet, bukan disimpan pada komputer lokal. Pada saat ini layanan seperti pengolahan kata, *spreadsheet*, dan aplikasi lainnya dipindahkan ke dalam komputasi awan.



Gambar 2.6 Cloud Computing

Cloud computing adalah gabungan pemanfaatan teknologi komputer dan pengembangan berbasis internet. Suatu metode komputasi dimana kemampuan terkait teknologi informasi disajikan sebagai suatu layanan sehingga pengguna dapat mengaksesnya lewat internet tanpa mengetahui apa yang ada didalamnya atau infrastruktur teknologi yang membantunya. Suatu konsep umum yang mencakup *SaaS*, *Web 2.0*, dan tren teknologi terbaru lain yang dikenal luas, dengan tema umum berupa ketergantungan terhadap internet untuk memberikan kebutuhan komputasi pengguna. Sebagai contoh, *Google Apps* menyediakan aplikasi bisnis umum yang dapat diakses melalui suatu penjelajah web dengan perangkat lunak dan data yang tersimpan di *server*.

2.3.1 Latar Belakang Munculnya Cloud Computing

Berawal dari kebiasaan pengguna individual yang tentu cukup apabila menyimpan data-datanya di sebuah laptop atau *personal computer*. Namun tentu berbeda dengan sebuah perusahaan, instansi, atau organisasi tertentu yang memiliki ribuan data penting dan membutuhkan media penyimpanan yang lebih besar dan lebih aman. Jawaban atas segala kondisi tersebut adalah *cloud computing*. Teknologi ini dianggap dapat menekan biaya investasi *server* raksasa, lebih efektif, dan efisien dari jumlah sumber daya manusia.

2.3.2 Karakteristik Cloud Computing

Tidak semua aplikasi berbasis web dapat dimasukkan ke dalam kategori *cloud computing*. Ada lima kriteria yang harus dipenuhi oleh sebuah sistem untuk bisa di masukkan dalam keluarga *Cloud Computing* [11, 12], yaitu :

1. *On Demand Self Service*

Seorang pengguna dimungkinkan untuk secara langsung “memesan” sumber daya yang dibutuhkan, seperti *processor time* dan kapasitas penyimpanan melalui *control panel* elektronis yang disediakan. Jadi tidak perlu berinteraksi dengan *personil customer service* jika perlu menambah atau mengurangi sumberdaya komputasi yang diperlukan.

2. *Broadband Network Access*

Layanan yang tersedia terhubung melalui jaringan pita lebar, terutama untuk dapat diakses secara memadai melalui jaringan internet, baik menggunakan *thin client*, *thick client* ataupun media lain seperti *smartphone*.

3. *Resource Pooling*

Penyedia layanan *cloud*, memberikan layanan melalui sumberdaya yang dikelompokkan di satu atau berbagai lokasi *data center* yang terdiri dari sejumlah *server* dengan mekanisme *multi-tenant*. Mekanisme *multi-tenant* ini memungkinkan sejumlah sumberdaya komputasi tersebut digunakan secara bersama-sama oleh sejumlah *user*, di mana sumberdaya tersebut baik yang

berbentuk fisik maupun *virtual*, dapat dialokasikan secara dinamis untuk kebutuhan pengguna atau pelanggan sesuai permintaan.

Dengan demikian, pelanggan tidak perlu tahu bagaimana dan darimana permintaan akan sumberdaya komputasinya dipenuhi oleh penyedia layanan. Yang penting, setiap permintaan dapat dipenuhi. Sumber daya komputasi ini meliputi media penyimpanan, *memory*, *processor*, pita jaringan dan mesin *virtual*.

4. *Rapid Elasticity*

Kapasitas komputasi yang disediakan dapat secara elastis dan cepat disediakan, baik itu dalam bentuk penambahan ataupun pengurangan kapasitas yang diperlukan. Untuk pelanggan sendiri, dengan kemampuan ini seolah-olah kapasitas yang tersedia tak terbatas besarnya, dan dapat “dibeli” kapan saja dengan jumlah berapa saja.

5. *Measured Service*

Sumber daya *cloud* yang tersedia harus dapat diatur dan dioptimasi penggunaannya, dengan suatu sistem pengukuran yang dapat mengukur penggunaan dari setiap sumber daya komputasi yang digunakan (penyimpanan, *memory*, *processor*, lebar pita, aktivitas *user*, dan lainnya). Dengan demikian, jumlah sumberdaya yang digunakan dapat secara transparan diukur yang akan menjadi dasar bagi *user* untuk membayar biaya penggunaan layanan.

2.3.2 Layanan Cloud Computing

Service model dari layanan *cloud* [13, 14] secara umum dikategorikan ke dalam 3 model, yaitu :

- **Infrastructure-as-a-Service (IaaS)**

IaaS adalah sebuah model layanan dimana penyedia *cloud* menyediakan hardware atau perangkat keras (komputer *server*, penyimpanan data, jaringan, dll) untuk pelanggan. Manajemen perangkat keras menjadi tanggung jawab

penyedia layanan, dan pelanggan mengontrol *operating system* serta aplikasi yang diinstal ke dalam *server*.

- **Platform-as-a-Service (PaaS)**

PaaS adalah model layanan yang menyediakan semua hal yang dibutuhkan untuk mengembangkan sebuah aplikasi pada *cloud*. PaaS menawarkan fasilitas untuk mengembangkan, *testing*, *deployment*, hingga *maintenance* aplikasi tanpa harus membeli infrastruktur dan *software environment (Operating System)*. Contohnya: *Windows Azure Platform*, *Google App Engine*, *VMforce.com* [15].

- **Software-as-a-Service (SaaS)**

SaaS adalah model layanan dimana pelanggan *cloud* menggunakan aplikasi yang sudah disediakan dalam *cloud*. SaaS adalah bentuk *cloud* yang paling umum digunakan saat ini. Contoh: *Office 365*, *Salesforce.com*, *Hosted Exchange*, *Salesforce.com* [15].

2.3.3 Penggunaan Google App Engine (GAE) sebagai Platform as a Service

Google App Engine merupakan jawaban Google.Inc atas kebutuhan *cloud computing* yang semakin mendapat perhatian dari berbagai kalangan dunia IT. *Google App Engine* adalah suatu model cloud berbasis *Platform as a Service*, sebuah sistem operasi di *cloud* yang menyediakan layanan untuk *hosting*, pengaturan, penyimpanan yang *scalable* hingga pengaturan infrastruktur dalam rangka pengadaan dan distribusi layanan yang berbasis *cloud*. *Google App Engine* pertama kali dirilis sebagai versi beta pada bulan April 2008.

Google App Engine platform menawarkan fleksibilitas dan lingkungan pengembangan yang familiar bagi para *developer* untuk membangun aplikasi berbasis *cloud* dan layanan. Dengan GAE, seorang *developer* mampu memangkas waktu pengembangan dan segera beradaptasi dengan kebutuhan pertumbuhan layanan yang dapat disediakan.

Google App Engine sangat cocok digunakan untuk menjalankan kebutuhan proses yang terjadi di *cloud*, membangun, modifikasi dan distribusi aplikasi yang *scalable* dengan *resource* yang minimal, melakukan penyimpanan data skala besar, dan komputasi yang bervolume tinggi. *Google App Engine* gratis sampai tingkat tertentu sumber daya yang digunakan. Biaya yang dikenakan untuk penyimpanan tambahan, *bandwidth* atau siklus CPU yang dibutuhkan aplikasi. Saat ini, *Google App Engine* didukung oleh bahasa pemrograman Python dan Java (dengan perluasan JVM bahasa seperti Groovy, JRuby, Scala, Clojure).

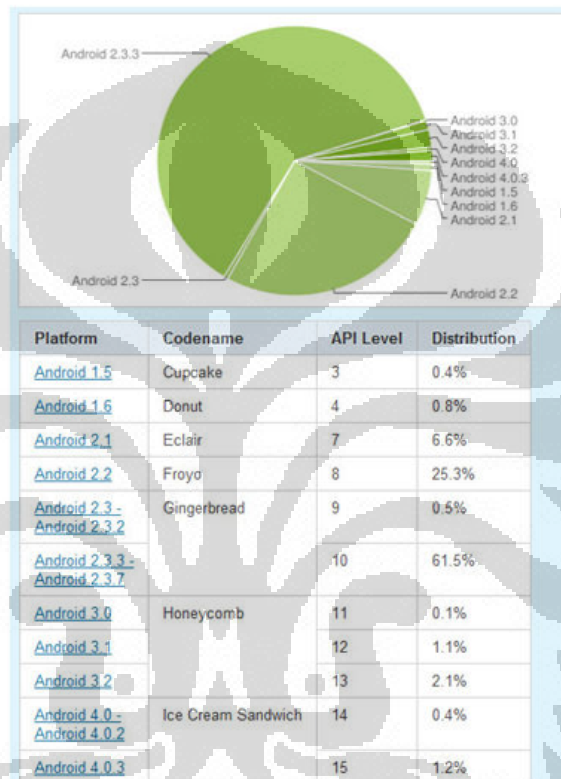
Pada penerapan skripsi ini, *Google App Engine* berperan sebagai *Web Service* yang digunakan sebagai sebuah layanan yang dapat digunakan siapapun dan disebarluaskan pada jaringan internet untuk diintegrasikan pada aplikasi-aplikasi lainnya. Komputasi terdistribusi ini menguntungkan berbagai macam sistem yang tidak mampu melakukan pemrosesan tertentu karena suatu keterbatasan. *Google App Engine* dijadikan sebagai layanan yang dapat diidentifikasi untuk *Unique Resource Identifier* (URI) dan dialamatkan dengan *Unique Resource Locator* (URL) yang menerima permintaan dari client melalui HTTP dan mengembalikannya dengan suatu respon berbasis XML atau JSON. Namun pada pengembangannya kami hanya memanfaatkan respon dengan format JSON. Respon inilah yang digunakan sebagai format data untuk informasi hasil pengenalan wajah yang diterima oleh *client*. Respon JSON tersebut berasal dari *Face.com* API yang merupakan layanan pemrosesan rekognisi wajah.

2.4 Platform Android sebagai Media Pengembangan Sistem

Android adalah sebuah sistem operasi perangkat mobile yang dirilis oleh Google. Android tergolong sistem operasi baru yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri dan digunakan oleh bermacam perangkat *mobile*. Perkembangan teknologi terkait Android yang dirilis oleh perusahaan *Google Inc* tergolong cepat, begitu banyak versi yang telah dapat dimanfaatkan fungsi dan interaksinya saat ini. Versi perdana android diberi nama Android versi 1.1. Berikutnya Android Versi 1.5 (Cupcake), Android Versi 1.6 (Donut), Android versi 2.0/2.1 (Eclair),

Android Versi 2.2 (Froyo: Frozen Yoghurt), Android Versi 2.3 (Gingerbread), Android Versi 3.0 (Honeycomb), dan saat ini Android Versi 4.0 (Ice Cream Sandwich).

Berdasarkan persebaran versi Android dan penggunaannya, berikut statistik tentang distribusi platform Android [20] per Maret 2012.



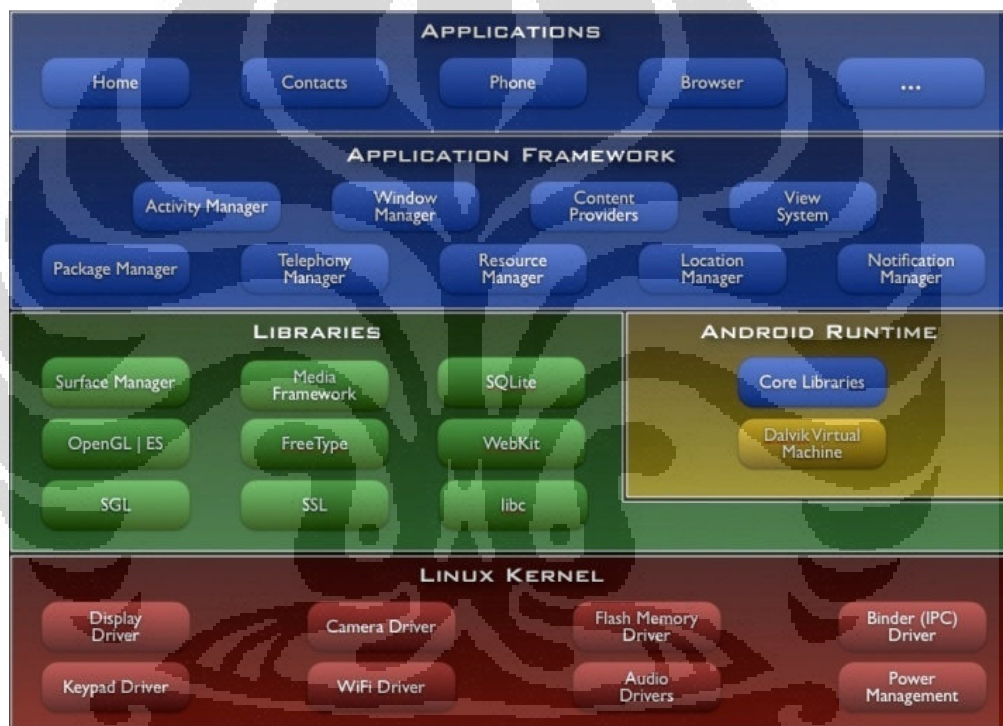
Gambar 2.7 Distribusi Platform Android

Android memiliki beberapa fitur penting, antara lain :

- *Framework* Aplikasi yang mendukung penggantian komponen dan *reusable*.
- Mesin Virtual Dalvik (DVM) dioptimalkan untuk perangkat *mobile*
- *Integrated browser* berdasarkan *engine open source WebKit*
- Grafis yang dioptimalkan dan didukung oleh perpustakaan grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional akselerasi *hardware*)
- SQLite untuk penyimpanan data

- *Media Support* yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF) GSM Telephony (tergantung *hardware*)
- Bluetooth, EDGE, 3G, dan WiFi (tergantung *hardware*)
- Kamera, GPS, kompas, dan *accelerometer* (tergantung *hardware*)
- Lingkungan Development yang lengkap dan kaya termasuk perangkat *emulator*, *tools* untuk *debugging*, profil dan kinerja memori, dan *plugin* untuk Eclipse IDE.

Berikut ini adalah arsitektur dari sistem operasi Android.



Gambar 2.8 Arsitektur sistem operasi Android [23]

Sesuai Gambar 2.5 diatas, terlihat dari lapisan paling bawah dijelaskan bahwa Android menggunakan kernel linux sebagai pengatur keseimbangan atas perangkat keras yang ada pada perangkat Android. Sejak versi 2.6, Linux Kernel menggunakan *Completely Fair Scheduler* (CFS), yang menerapkan kerataan dalam waktu penggunaan CPU untuk *task* (tugas) yang diberikan. Keseimbangan ini menjamin bahwa semua tugas akan mempunyai jatah CPU yang sama. Oleh karena itu, setiap kali terjadi ketidakrataan akan dilakukan penyeimbangan

kembali pada tugas yang sedang berjalan. Lebih jauh lagi, kernel juga menyediakan layanan sistem yang paling dasar seperti sekuritas, manajemen memori, manajemen proses, dan *stack* jaringan.

Lapisan selanjutnya ada *Library* yang menyediakan fitur inti untuk menjalankan fungsi operasional aplikasi. Kemudian terdapat *Android Run Time*. Android menyediakan *Virtual Machine* sendiri yaitu Dalvik. Semua perangkat keras yang berbasis Android dijalankan dengan menggunakan *Virtual Machine* tersebut untuk eksekusi aplikasi sehingga memori yang digunakan menjadi optimal dan sumber daya yang digunakan menjadi terbatas.

Pada dua lapisan teratas terdapat *Application Framework* dan *Application*. *Application Framework* memungkinkan seorang pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi, serta menjalankan layanan yang ada didalamnya. Lapisan *Application* merupakan ruang utama yang terhubung langsung dengan aplikasi seperti SMS, email, kontak, browser, dan lain-lain.

Dalam proses pembangunan aplikasi android, pengembang memiliki beberapa pilihan ketika ingin membuat aplikasi berbasis android. Kebanyakan pengembang menggunakan Eclipse sebagai IDE populer yang tersedia bebas untuk merancang dan mengembangkan aplikasi Android. Bahasa pemrograman yang digunakan android adalah Java. Namun bukan berarti keterlibatan bahasa lain seperti C/C++ tidak bisa disertakan. Terdapat sebuah komponen pendukung yakni *Native Development Kit* (NDK) yang digunakan untuk menggabungkan fungsi yang ditulis dalam C/C++ menjadi sebuah aplikasi Android [21]. Keberadaan NDK sangatlah menguntungkan, selain untuk meningkatkan kinerja aplikasi melalui NDK kita dapat memanfaatkan kode-kode dari bahasa lain untuk penyempurnaan aplikasi android yang akan dibangun. Dasar dalam mengembangkan aplikasi android yang utama adalah penggunaan *Software Development Kit* (SDK) yang harus diintegrasikan kedalam lingkungan Eclipse IDE.

Pengembangan *augmented reality* sebagai penampil informasi hasil pengenalan wajah ini menggunakan Android versi API 9 yaitu *Android Gingerbread*. *Android Gingerbread* tergolong versi yang banyak digunakan pengguna android, selain itu tingkat kompatibilitas yang dimilikinya sangatlah baik bila dibandingkan versi lain sehingga cocok untuk digunakan sebagai wadah pengembangan sistem.

2.5 Facebook API sebagai Media Penghubung Jejaring Sosial

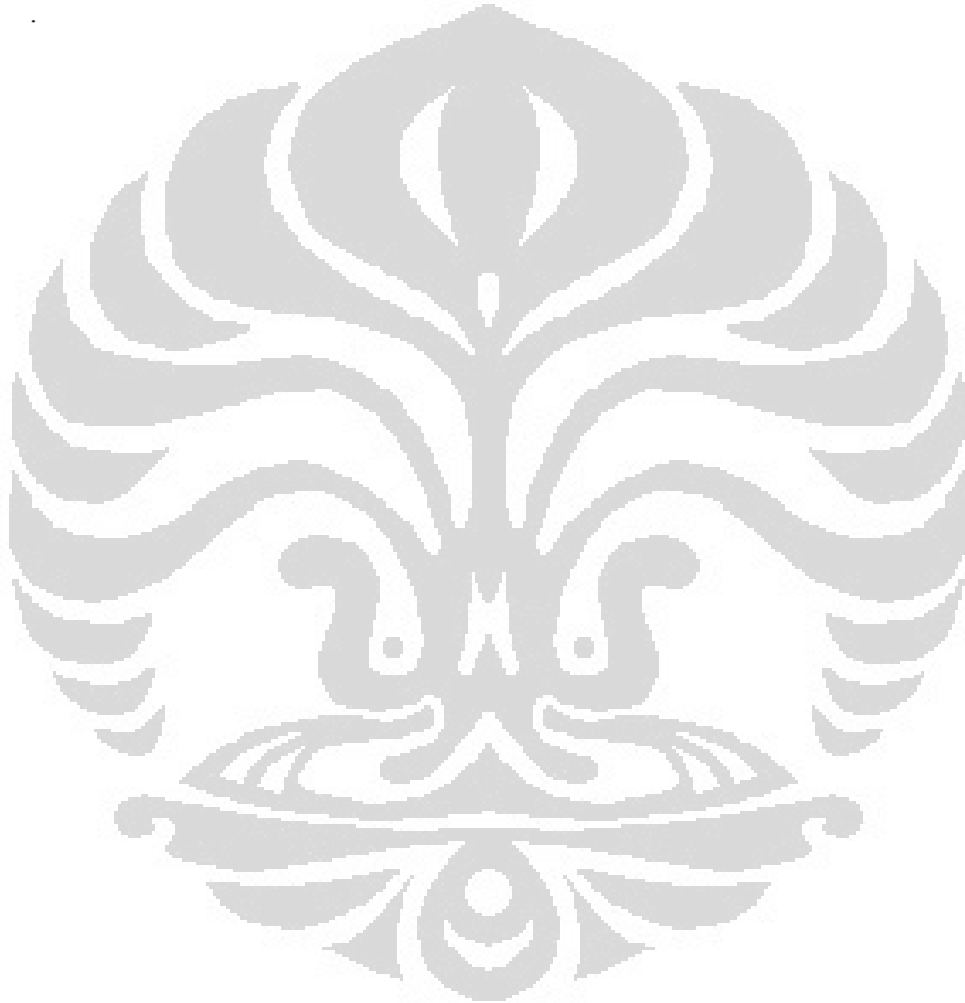
Application Programming Interface (API) merupakan sebuah aplikasi pemrograman yang secara khusus dikembangkan sebagai perantara komunikasi antar komponen perangkat lunak. Sistem pengenalan wajah yang dikembangkan, membawa pengguna untuk dapat terhubung langsung ke sebuah layanan jejaring sosial. Oleh karena itu dibutuhkan sebuah jejaring sosial API. Jejaring sosial API yang digunakan adalah Facebook API.

Facebook API adalah sebuah mekanisme yang memungkinkan para pengembang untuk dapat mengakses informasi yang terdapat pada Facebook *server*. Facebook API dapat menjadi perantara aplikasi yang dikembangkan sehingga secara tidak langsung pengguna dapat membaca, menulis, dan berinteraksi ke Facebook.

Facebook API memiliki lima komponen yang menjadi konsep utama yaitu:

1. **Social Plugins** yang memungkinkan siapapun untuk berbagi informasi pada situs pribadi ke pengguna Facebook yang akan menampilkan urutan informasi yang paling populer berdasarkan jumlah pembaca.
2. **Graph API** merupakan komponen utama dari *Facebook Platform* yang memungkinkan siapapun terutama pengembang aplikasi untuk melakukan proses *read* dan *write* ke situs Facebook. Graph API merupakan API terbaru sebagai pengganti REST API yang lama.
3. **Social Channels** memungkinkan kita untuk mengintegrasikan website atau aplikasi kita dengan *social channels* yang dimiliki Facebook seperti *news feed* untuk mendorong perkembangan website tersebut.

4. **Facebook Authentication** memungkinkan aplikasi kita untuk berinteraksi dengan Graph API dan menyediakan *Single-Sign On* (SSO) menggunakan OAuth 2.
5. **Open Graph Protocol** ini merupakan sebuah protokol yang memungkinkan semua orang dengan mudah mengubah halaman web apapun menjadi sebuah bagian dalam sebuah “*social graph*”..



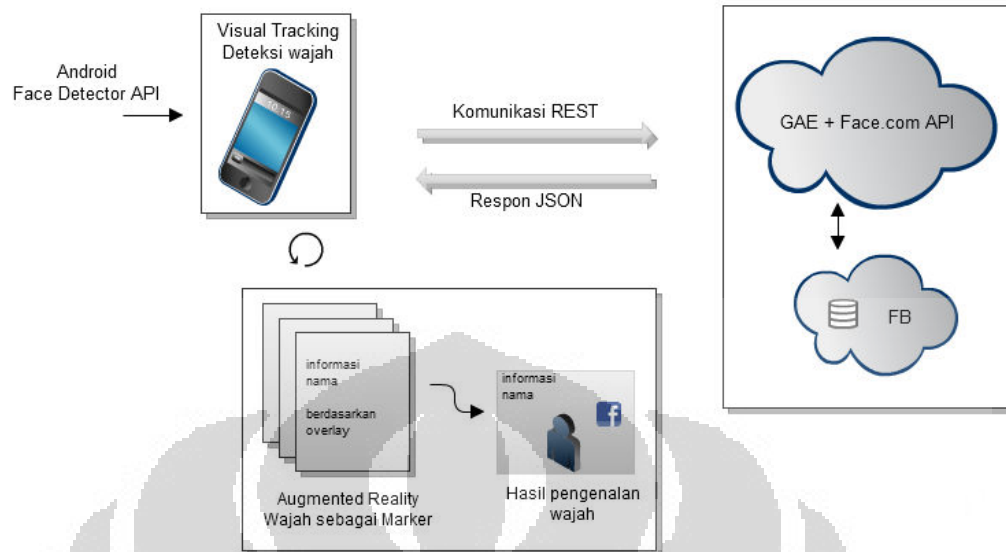
BAB 3

PERANCANGAN SISTEM PENGENALAN WAJAH DAN TAMPILAN INFORMASI BERBASIS *AUGMENTED REALITY*

3.1 Deskripsi Umum Sistem

Pada tulisan dibagian awal, telah dijelaskan bahwa sistem pengenalan wajah berbasis *cloud* ini dirancang sebagai solusi atas masalah proses komputasi yang dinilai berat pada sistem *face recognition* umumnya. Berawal dari hal tersebut dirancnglah sebuah sistem sebagai satu kesatuan utuh yang terdiri dari pemanfaatan layanan komputasi awan untuk proses *Face Recognition*, *Face Recognition* modul yang tertanam pada perangkat *mobile* dan penerapan *Augmented Reality* modul sebagai penampil informasi hasil pengenalan wajah. Perangkat *mobile* yang digunakan pada pengembangan sistem berbasis *platform* Android. Dengan sistem pengenalan wajah tersebut, pengguna akan dapat mengenali wajah seseorang menggunakan kamera yang terdapat perangkat Android hanya dengan mengarahkannya ke area wajah. Pengerjaan sistem utuh ini dilakukan oleh sebuah tim yang terdiri atas Slamet Budiayatno, Nur Muhammad Ridho [18], dan Prasetyawidi Indrawan [19].

Perancangan sistem pengenalan wajah meliputi dua aspek yang berkaitan satu dengan lainnya, yaitu perancangan sistem berbasis *mobile phone* dan pemanfaatan layanan *cloud*. Dua aspek ini terintegrasi satu sama lain melalui proses komunikasi dengan bantuan *Representational State Transfer* (REST) yang menghubungkan perangkat *mobile* dengan layanan *cloud*. Untuk melihat gambaran umum sistem secara keseluruhan dapat dilihat arsitektur sistem pengenalan wajah berikut ini.

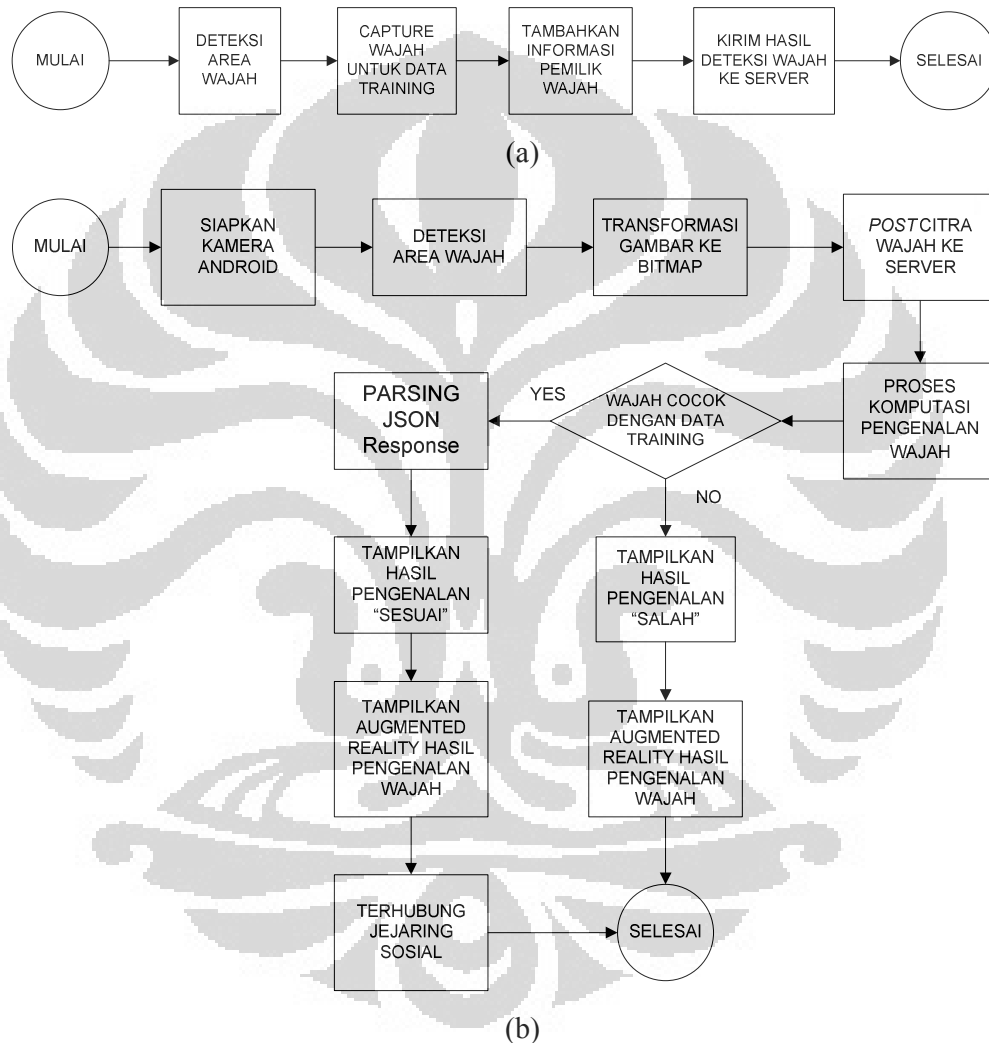


Gambar 3.1 Arsitektur Sistem Pengenalan Wajah

Arsitektur sistem yang ditampilkan pada Gambar 3.1, menjelaskan bahwa pada perancangannya sistem pengenalan wajah dibangun pada *cloud* dengan memanfaatkan layanan dari Google App Engine. Pada *cloud* tersebut akan ditanamkan sebuah modul yang bertugas melakukan proses komputasi pengenalan wajah melibatkan layanan dari *Face.com* API. Hasil proses komputasi tersebut dapat berupa teks dan gambar yang menyajikan informasi pemilik wajah. Selain melakukan proses komputasi, *cloud* juga dimanfaatkan sebagai tempat penyimpan citra wajah bagi para pengguna. Pada sisi yang berbeda sistem pengenalan wajah ini juga diterapkan pada perangkat Android. Proses komputasi yang dilakukan pada perangkat *mobile* tersebut adalah proses deteksi wajah. Hal ini dilakukan guna mengefisiensi sumber daya yang terpakai oleh perangkat *mobile*. Untuk proses komputasi pengenalan wajah dilakukan pada *cloud*, dengan sebelumnya *client* pada perangkat android melakukan permintaan pada *cloud* sehingga hasil pengenalan wajah dapat dikirimkan kembali pada perangkat android. Hasil yang dikirimkan dari *cloud* tersebut ditampilkan dalam bentuk *Augmented Reality* sehingga tampilan lebih terlihat interaktif.

3.2 Alur Kerja Sistem

Sistem pengenalan wajah merupakan suatu sistem yang melibatkan beberapa komponen pendukung meliputi modul pengenalan wajah, layanan komputasi awan, dan *augmented reality*. Untuk mempermudah visualisasi penggunaan, berikut adalah alur kerja sistem pengenalan wajah.



Gambar 3.2 Diagram Alir Umum Sistem Pengenalan Wajah berbasis Cloud

(a). Training Wajah dan (b). Rekognisi Wajah

Sesuai dengan Gambar 3.2, perancangan sistem pengenalan wajah ini terdiri atas dua bagian yang saling terhubung yakni *training* wajah dan proses

rekognisi wajah. Pada Gambar 3.2 (a), sistem berjalan diawali dengan proses pengambilan sebuah citra wajah melalui kamera perangkat android untuk melakukan *training* (pelatihan) terhadap wajah. Pelatihan tersebut dilakukan dengan mengambil area depan bagian wajah seseorang dengan berbagai ekspresi. Setelah citra wajah diambil, sistem membutuhkan sisipan informasi mengenai pemilik wajah sebagai label *database* masing-masing wajah. Selanjutnya citra wajah yang telah memiliki label dikirimkan ke layanan *cloud server* untuk pengolahan citra wajah.

Apabila pelatihan data wajah telah selesai, maka proses rekognisi wajah dapat dimulai. Pengenalan terhadap citra wajah berlangsung melalui tahapan yang terlihat pada Gambar 3.2 (b). Berdasarkan skema yang ditunjukkan, pertama dilakukan pengarahannya kamera android ke wajah seseorang yang ingin dikenali. Permulaan dari rekognisi wajah yakni pemilihan area wajah dan mengabaikan area yang bukan wajah. Proses ini dikenal dengan nama deteksi wajah. Deteksi wajah akan mentransformasikan format wajah ke dalam bitmap. Selanjutnya wajah hasil deteksi tersebut dikirimkan ke layanan *cloud* untuk diproses. Citra wajah hasil deteksi yang telah sampai pada *cloud* akan dikembalikan kepada android *client* dalam bentuk respon JSON. Proses pengenalan dimulai dengan mencocokkan citra wajah masukan dengan citra wajah yang sebelumnya telah dilatih. Jika citra wajah sesuai dengan data *training* maka dilakukan proses komputasi. Komputasi yang terjadi yakni pengambilan informasi terhadap wajah serta penyisipan *augmented reality* sebelum sampai kepada pengguna. Hasil dari komputasi ini dikirimkan kembali pada perangkat android sebagai hasil identitas diri wajah yang ingin diketahui. Melalui *augmented reality* yang ditampilkan pengguna dapat melakukan interaksi sehingga mampu terhubung dengan jejaring sosial.

3.3 Diagram Unified Modelling Language (UML)

Perancangan sistem ini dibuat berdasarkan kaidah *Unified Modelling Language* (UML). UML merupakan sebuah metode untuk merepresentasikan atau mendeskripsikan desain perangkat lunak ke dalam notasi-notasi grafis yang terstandarisasi. Dengan UML, rancangan perangkat lunak dapat direpresentasikan

ke dalam diagram-diagram yang memiliki fungsi masing-masing. Berikut adalah diagram-diagram yang menggambarkan rancangan dari sistem yang akan dibuat, meliputi *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*, dan *deployment diagram*.

3.3.1 Use Case Diagram

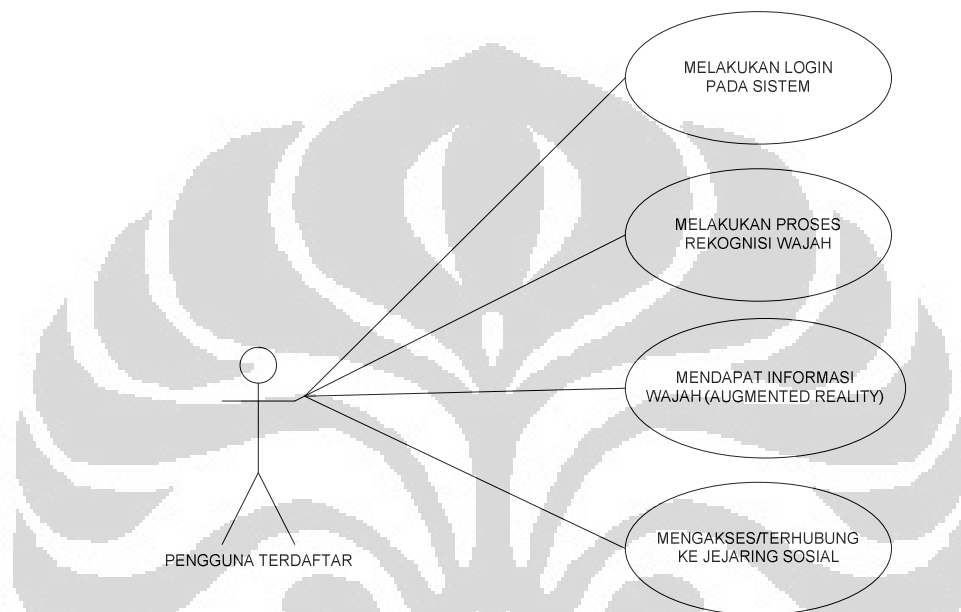
Use Case Diagram menjelaskan fungsi yang dimiliki oleh sistem dari segi fungsionalitas yang dilakukan oleh pengguna sistem [17]. *Use Case Diagram* dari sistem ini terbagi menjadi dua jenis yaitu *Use Case Diagram* untuk registrasi awal pengguna sebagai tahap awal untuk proses *training* data wajah pengguna aplikasi (dapat dilihat pada Gambar 3.2) dan *Use Case Diagram* untuk pengguna terdaftar sebagai tahap akhir proses pengenalan wajah (dapat dilihat pada Gambar 3.3).



Gambar 3.3 *Use Case Diagram* Registrasi Pengguna

Gambar 3.3 merupakan diagram *use case* dari sistem pengenalan wajah, yaitu diagram *use case* yang menunjukkan interaksi antara pengguna baru dengan sistem. Pertama, pengguna harus memilih fitur registrasi pada sistem. Selanjutnya

akan masuk pada tampilan fitur *training*, pengguna diminta untuk mengambil citra wajah depan dirinya dari berbagai ekspresi. Kemudian citra wajah yang sudah diambil harus diberikan label berupa informasi data diri. Sebagai tahapan akhir, citra wajah yang telah memiliki label informasi diunggah ke *cloud server* sebagai data *training*.

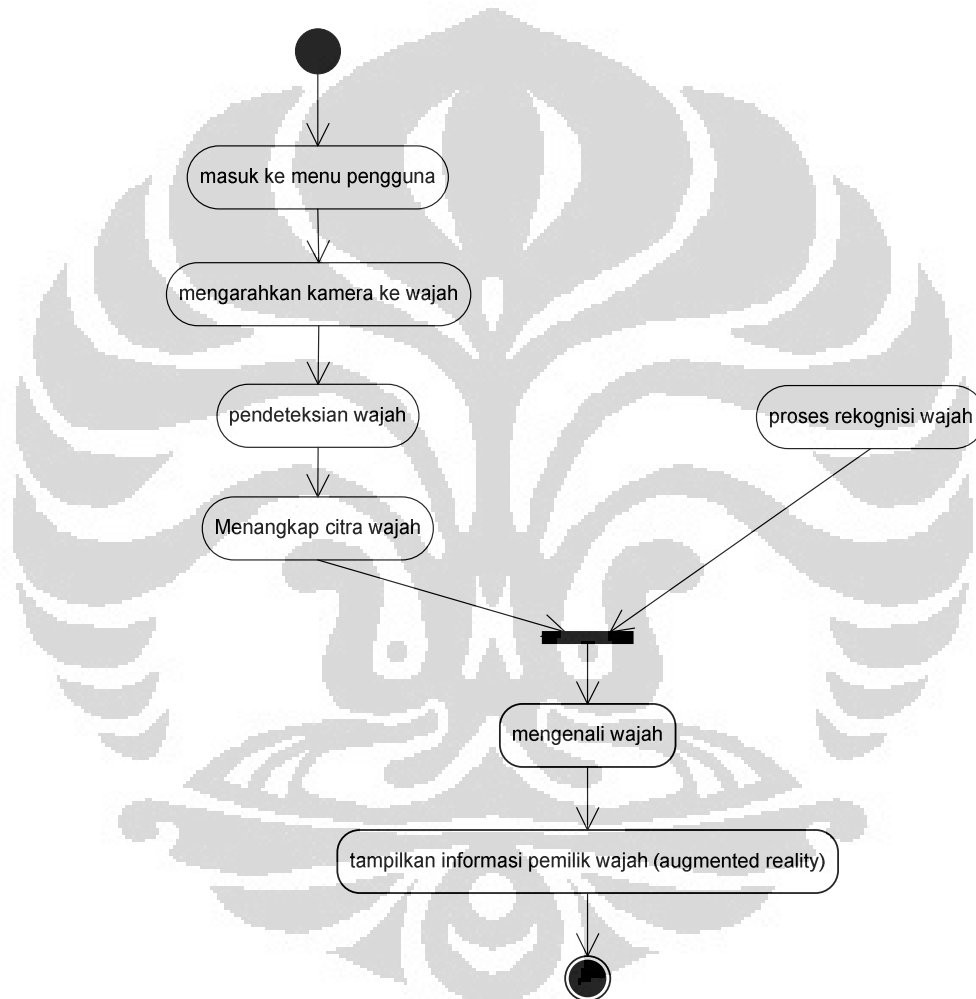


Gambar 3.4 Use Case Diagram Pengguna terdaftar

Gambar 3.4 di atas merupakan diagram *use case* yang menunjukkan interaksi antara pengguna terdaftar dengan sistem. Untuk menggunakan sistem pengguna harus melakukan login terlebih dahulu. Selanjutnya pengguna harus memilih fitur rekognisi untuk mengetahui wajah yang ingin dikenal. Saat proses pendeteksian wajah dimulai, berlanjut menuju proses pengenalan wajah yang dilayani oleh *cloud*. Hasil informasi yang diperoleh akan ditampilkan ke pengguna dalam bentuk *Augmented Reality* sebagai wajah yang dikenal. Selanjutnya pengguna dapat berinteraksi dengan informasi tersebut untuk menghubungkannya ke jejaring sosial.

3.3.2 Activity Diagram

Activity Diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana aliran berawal, keputusan yang mungkin terjadi, dan bagaimana aliran berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi juga menjelaskan proses dan jalur aktivitas dari level atas secara umum.



Gambar 3.5 Activity Diagram

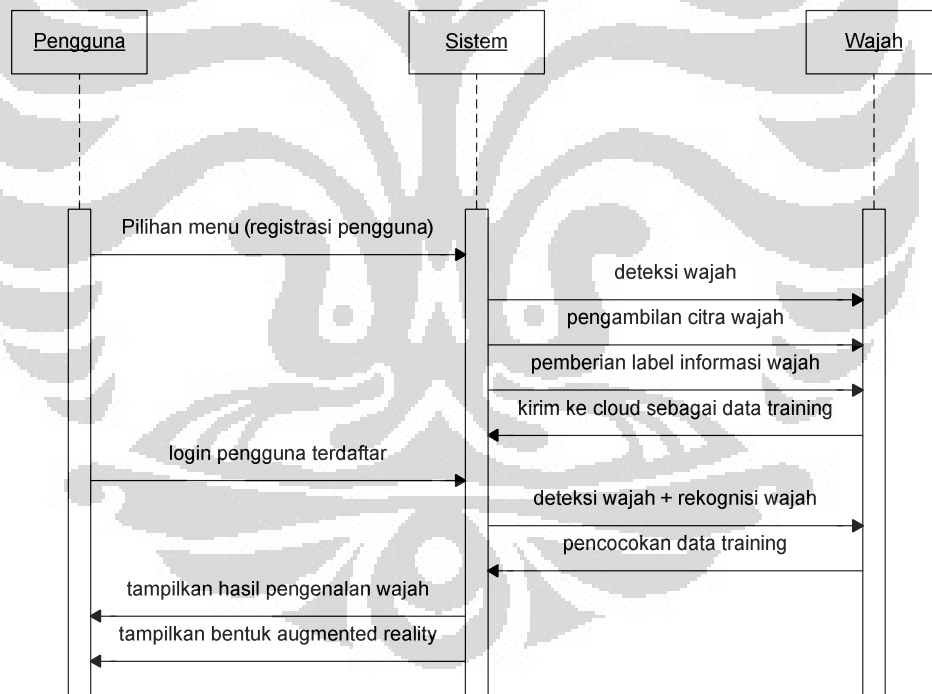
Activity diagram pada Gambar 3.5 di atas menjelaskan alur kerja sebagai berikut :

1. Sistem akan memberikan pilihan kepada pengguna untuk registrasi terlebih dahulu atau menuju halaman login bila sudah terdaftar.

2. Pengguna harus memilih untuk melakukan proses *training* atau proses rekognisi. Proses *training* memungkinkan pengguna untuk melakukan pelatihan terhadap wajah dan proses rekognisi untuk melakukan pengenalan terhadap wajah. Kedua proses tersebut diawali dengan deteksi terhadap wajah.
3. Ketika deteksi wajah berakhir, pengguna dapat meng-*capture* citra wajah untuk melanjutkan proses rekognisi.
4. Sistem mengolah dan menampilkan informasi kembali pada pengguna dalam bentuk *augmented reality*.

3.3.3 Sequence Diagram

Sequence Diagram merupakan salah satu diagram untuk menggambarkan perilaku objek yang berhubungan langsung dengan sistem [15].



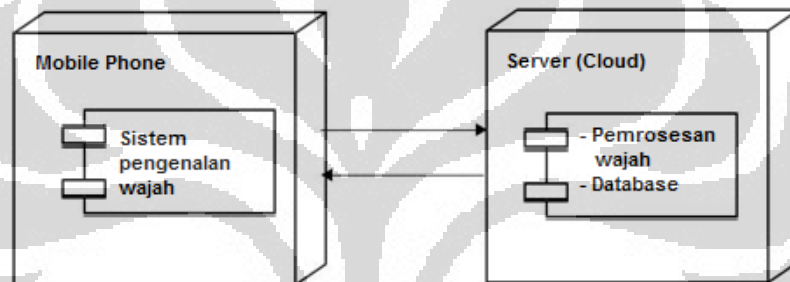
Gambar 3.6 Sequence Diagram

Berdasarkan Gambar 3.6, interaksi pengguna dengan sistem ditunjukkan saat proses pengambilan wajah orang yang ingin dikenal dengan mengirimkannya ke *cloud*. *Cloud* melakukan proses komputasi dengan mencocokkan wajah,

kemudian diklasifikasi sebagai wajah yang dikenal atau tidak. Data hasil komputasi akan dikirimkan balik ke perangkat mobile sebagai bentuk informasi orang yang ingin dikenal. Informasi hasil pengenalan wajah yang diterima pengguna dilihat sebagai visualisasi informasi dalam bentuk *augmented reality*.

3.3.4 Deployment Diagram

Deployment Diagram merepresentasikan rancangan fisik dari sistem berupa perangkat keras maupun perangkat lunak untuk menunjukkan proses yang ada padanya [17].



Gambar 3.7 Deployment Diagram

Pada sistem pengenalan wajah berbasis *cloud* ini melibatkan dua komponen penting yaitu perangkat *mobile phone* dengan *platform* Android dan *server* berupa *Cloud*. Pada sisi perangkat mobile dibangun aplikasi pengenalan wajah sebagai media komunikasi dengan bantuan REST sedangkan pada sisi *cloud* dibangun modul pengenalan wajah sebagai proses komputasi pencocokan wajah dengan pemiliknya yang dilayani atas bantuan *Face.com API*. Hasil komputasi akan dikembalikan pada perangkat mobile dalam bentuk informasi pemilik wajah yang ditampilkan menggunakan konsep *Augmented Reality*.

BAB 4

IMPLEMENTASI MODUL *AUGMENTED REALITY* DAN INTEGRASI PADA SISTEM PENGENALAN WAJAH

4.1 Perancangan Modul *Augmented Reality*

Pada bab ini akan dibahas mengenai implementasi modul *Augmented Reality* pada sistem pengenalan wajah. Awalan dari proses implementasi modul tersebut ialah perancangan dalam bentuk *standalone* (terpisah). Selanjutnya dilakukan integrasi terhadap sistem keseluruhan. Pembahasan tersebut dapat dilihat pada Subbab dibawah ini.

4.1.1 Perangkat Pendukung Pengembangan

Seperti yang telah dijelaskan pada bab sebelumnya, sistem pengenalan wajah dengan memanfaatkan *Augmented Reality* dan *Cloud Computing* ini akan dikembangkan pada perangkat android. Untuk itu dibutuhkan lingkungan pengembangan terintegrasi (*Integrated Development Environment*; IDE) sebagai wadah melakukan pengembangan program yang lebih efektif dan efisien. Untuk melakukan pengembangan tersebut, digunakan lingkungan pengembangan terintegrasi *Eclipse Indigo*. *Eclipse Indigo* merupakan IDE untuk java yang sampai saat ini merupakan IDE terpopuler untuk para *developer*. Bahkan sekarang Eclipse IDE tidak hanya mendukung bahasa pemrograman Java, melainkan juga mendukung PHP, C/C++, Python, Ruby dan lain-lain.

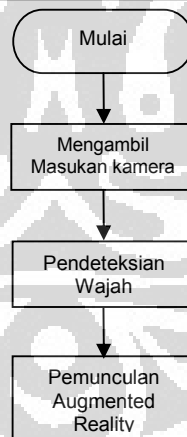
Pengembangan sistem ini juga melibatkan kode *native* seperti C/C++ untuk proses pengolahan citra yang melibatkan pustaka lain seperti OpenCV. OpenCV sangat cocok dalam ranah *Computer Vision* dan ditulis dalam bahasa C. Jadi untuk proses eksekusi dibutuhkan *Java Native Interface* (JNI) yang menyediakan sekumpulan API yang digunakan untuk mengakses fungsi-fungsi pada library (*.dll atau *.so) yang dibuat dengan bahasa pemrograman C yang terdapat pada OpenCV.

4.1.2 Perancangan Modul *Augmented Reality* Sistem Terpisah

Perancangan dapat dimulai ketika Eclipse Indigo terkonfigurasi plugin serta *Software Development Kit* (SDK) dan *Native Development Kit* (NDK) telah terpasang dengan baik. Selanjutnya pengembangan sistem dimulai dengan proses deteksi wajah. Pada fase ini pengembangan difokuskan untuk hasil modul yang berjalan sendiri untuk keperluan pengujian sementara sebelum diintegrasikan kedalam sistem secara keseluruhan.

Secara sederhana *augmented reality* yang terjadi pada sistem terpisah ini ditunjukkan oleh algoritma pada modul *pre-processing* Gambar 4.1. Diawali dari perangkat *mobile* android yang melakukan proses pengambilan citra wajah melalui kamera. Untuk mengakses perangkat kamera, sistem harus dideklarasikan *permission* untuk *camera* dalam *android manifest* dan juga menyertakan elemen `<uses-feature>` untuk mendeklarasikan *feature* yang digunakan. Berikut ini *permission* untuk aplikasi yang menggunakan kamera pada perangkat android.

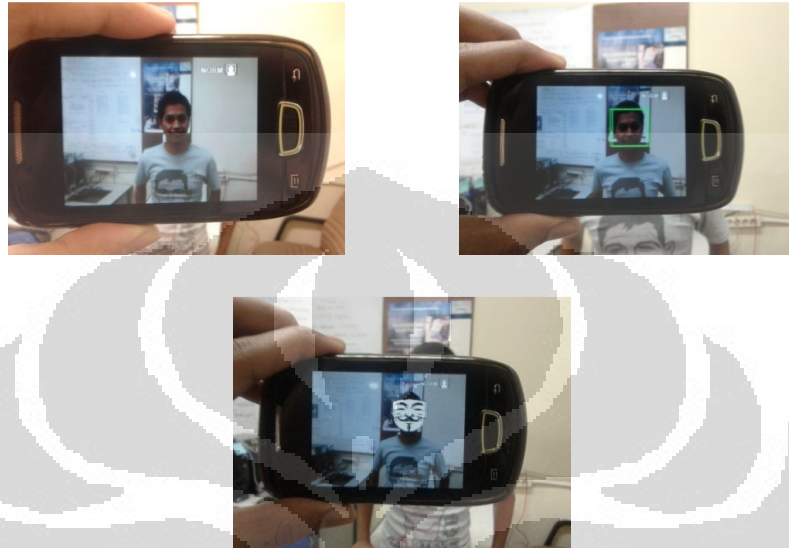
```
<uses-permission android:name="android.permission.CAMERA" />
<uses-feature android:name="android.hardware.camera" />
```



Gambar 4.1. Algoritma Modul *Pre-processing*

Tahap selanjutnya modul akan membaca masukan dari citra tersebut untuk dilakukan proses deteksi wajah. Deteksi wajah yang dimaksud adalah mengambil area wajah berdasar citra masukan dan mengabaikan area selain wajah. Proses tersebut bekerja melalui fungsi `android.media.FaceDetector` sebagai

deteksi wajah yang disediakan oleh Android. Hasil akhir setelah proses deteksi wajah adalah munculnya tampilan *augmented reality* tepat pada daerah wajah seperti yang terlihat pada Gambar 4.2.



Gambar 4.2 Tampilan modul *Augmented Reality* terpisah.

4.1.3 Pemrograman Modul *Augmented Reality*

Pada Subbab 4.1.3 akan dibahas mengenai teknis pemrograman yang menghasilkan modul terpisah sebelum terintegrasi pada sistem keseluruhan. Penggunaan modul *augmented reality* pada program *stand-alone* ini berdasarkan proses deteksi wajah yang ditangkap oleh kamera android. Gambar 4.3 menunjukkan fungsi utama deteksi wajah menggunakan kelas `FaceDetectThread`, fungsi ini menjadi awal proses masuknya objek wajah pada kamera, kemudian terjadi aktivitas penentuan area wajah sebagai mode pendeteksian ditandai dengan label kotak hijau pada area sekitar wajah.

```

/* Thread Class for Face Detection */
private class FaceDetectThread extends Thread {
    /* variables */
    private Handler handler_;
    private byte[] graybuff_ = null;

    /* Constructor */
    public FaceDetectThread(Handler handler){
        handler_ = handler;}

```



```

        /* set buffer */
        public void setBuffer(byte[] graybuff){
            graybuff_ = graybuff;
        }

        /* run the thread */
        @Override
        public void run() {
            /* face detector only needs grayscale image */
            // grayToRgb(graybuff_,rgbs_);
            // jni method

            gray8toRGB32(graybuff_,previewWidth_,previewHeight_,rgbs_);
            // java method
            float aspect =
            (float)previewHeight_/((float)previewWidth_;
            int w = prevSettingWidth_;
            int h = (int)(prevSettingWidth_*aspect);
            float xScale =
            (float)previewWidth_/((float)prevSettingWidth_;
            float yScale =
            (float)previewHeight_/((float)prevSettingHeight_;
            Bitmap bmp = Bitmap.createScaledBitmap(

            Bitmap.createBitmap(rgbs_,previewWidth_,previewHeight_,Bitmap.Con
            fig.RGB_565),
                w,h,false);
            // Log.i(TAG,"downscale
            w="+bmp.getWidth()+" ,h="+bmp.getHeight());
            int prevfound=0,trackfound=0;
            for(int i=0; i<MAX_FACE; i++){
                FaceResult face = faces_[i];
                float eyedist = face.eyesDistance();
                if(eyedist==0.0f)
                    continue;
                PointF midEyes = new PointF();
                face.getMidPoint(midEyes);
                prevfound++;
                PointF lt = new PointF(midEyes.x-
                eyedist*2.5f,midEyes.y-eyedist*2.5f);
                Rect rect = new
                Rect((int)(lt.x),(int)(lt.y),(int)(lt.x+eyedist*5.0f),(int)(lt.y+eyedist
                *5.0f));
                /* fix to fit */
                rect.left = rect.left < 0 ? 0 : rect.left;
                rect.right = rect.right > w ? w : rect.right;
                rect.top = rect.top < 0 ? 0 : rect.top;
                rect.bottom = rect.bottom > h ? h :
                rect.bottom;
                if(rect.left >= rect.right || rect.top >=
                rect.bottom )
                    continue;
                /* crop */
                Bitmap facebmp =
                Bitmap.createBitmap(bmp,rect.left,rect.top,rect.width(),rect.height());
                FaceDetector.Face[] trackface = new

```

```

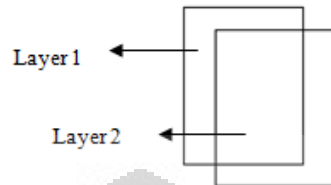
FaceDetector.Face[1];
FaceDetector tracker = new FaceDetector(
facebmp.getWidth(),facebmp.getHeight(),1);
int found = tracker.findFaces(facebmp,
trackface);
if(found!=0){
PointF ptTrack = new PointF();
trackface[0].getMidPoint(ptTrack);
ptTrack.x += (float)rect.left;
ptTrack.y += (float)rect.top;
ptTrack.x *= xScale;
ptTrack.y *= yScale;
float trkEyedist =
trackface[0].eyesDistance()*xScale;
faces_[i].setFace(ptTrack,trkEyedist);
trackfound++;
}
}
if(prevfound==0||prevfound!=trackfound){
FaceDetector.Face[] fullResults = new
FaceDetector.Face[MAX_FACE];
fdet_.findFaces(bmp, fullResults);
/* copy result */
for(int i=0; i<MAX_FACE; i++){
if(fullResults[i]==null)
faces_[i].clear();
else{
PointF mid = new PointF();
fullResults[i].getMidPoint(mid);
mid.x *= xScale;
mid.y *= yScale;
float eyedist =
fullResults[i].eyesDistance()*xScale;
faces_[i].setFace(mid,eyedist);
}
}
}
/* post message to UI */
handler_.post(new Runnable() {
public void run() {
overlayLayer_.postInvalidate();
// turn off thread_lock
isThreadWorking_ = false;
}
}

```

Gambar 4.3. Fungsi Deteksi Wajah

Pada prinsipnya alur kerja dari modul *augmented reality* yang dibangun pada perangkat android terletak pada penentuan aktivitas per *layer* (lapisan). Setiap lapisan memiliki aktivitas atau tugas masing-masing. Penentuan tugas

tersebut menentukan aktivitas dari lapisan, apakah tampil ke permukaan sebagai lapisan teratas atau terbawah. Sehingga saat kedua aktivitas itu berjalan bersamaan, pengguna melihatnya menjadi satu kesatuan.



Gambar 4.4 Layer pada *Augmented Reality*

Lapisan pertama berisi aktivitas untuk menjalankan fungsi kamera yang terdapat pada potongan kode dibawah ini (Gambar 4.5). Lapisan tersebut menjalankan fungsi *camera preview*. Hal inilah yang membuat pandangan kamera dapat mencatat atau menangkap gambar untuk menunjukkan bahwa kamera dapat melihat objek didepannya.

```
private PreviewView camPreview_;

.....
/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    /* set Full Screen */
    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
        WindowManager.LayoutParams.FLAG_FULLSCREEN);
    /* set window with no title bar */
    requestWindowFeature(Window.FEATURE_NO_TITLE);

    .....
    /* create camera view */
    camPreview_ = new PreviewView(this);
    camPreview_.setAppMode(appMode_);
    camPreview_.setfdetLevel(fdetLevel_, true);
    setContentView(camPreview_);

    .....
}
```

Gambar 4.5 Kode untuk pengaturan tampilan Kamera.

Sedangkan lapisan kedua menjalankan aktivitas untuk pemanggilan fungsi *Augmented Reality* yaitu dengan melibatkan kelas `SurfaceView`. Perangkat keras kamera yang terdapat pada android secara langsung menampilkan hasil objek yang tertangkap kamera ke permukaan layar dengan tambahan informasi atau gambar. Keluaran tersebut dibuat agar informasi atau gambar tambahan secara otomatis menumpuk pada permukaan layer sebelumnya. Hal tersebut dapat dilakukan dengan melibatkan `SurfaceView`. Ketersediaan `SurfaceView` membuat aktivitas utama dari aplikasi dapat ditempatkan ke area permukaan terdepan melapisi aktivitas sebelumnya. Untuk dapat melakukan kondisi tersebut aplikasi harus bertindak sebagai `SurfaceHolder` agar dapat melakukan pengendalian aktivitas yang tampil ke permukaan layar dan menerapkan metode `SurfaceHolder.Callback` untuk memungkinkan perangkat android menjalankan aktivitas sebagai tampilan akhir yang dimunculkan ke permukaan layar.

```

/* Class PreviewView for camera surface */
ClassPreviewView extends SurfaceView implements SurfaceHolder.Callback,
PreviewCallback, AutoFocusCallback

```

Gambar 4.5 Kode kelas `PreviewView`

Seperti yang telah dijelaskan diatas, aktivitas dibagi berdasarkan lapisannya. Lapisan teratas adalah lapisan *overlay*. Setiap gambar, teks atau benda lainnya yang tertangkap kamera dapat digunakan sebagai *overlay*. Gambar 4.6 adalah kode untuk pemakaian kelas *overlay*. Melalui kelas ini dapat ditambahkan suatu tanda atau marker sesuai yang diinginkan sebagai bagian dari realita tertambah.

```

/* Overlay Layer class */
public class OverlayLayer extends View {
    private Paint paint_ = new
Paint(Paint.ANTI_ALIAS_FLAG);
    /* Constructor */
    public OverlayLayer(Context context) {
        super(context);
        paint_.setStyle(Paint.Style.STROKE);
        paint_.setColor(0xFF33FF33);
        paint_.setStrokeWidth(3); }

```

Gambar 4.6 Kode pembuatan Overlay pada Android

Selanjutnya saat masing-masing lapisan telah berada pada posisinya, proses akan menuju pada mode pemilihan yang digunakan. Pada perancangan skripsi ini, modul *augmented reality* dibuat dengan dua mode seperti Gambar 4.7 yakni mode pertama sebagai mode deteksi dan mode kedua sebagai mode *augmented reality*.

```

public void setAppMode(int mode) {
    if(appMode_ == mode)
        return;
    appMode_ = mode;
    switch(appMode_)
    {
        case 0: {
            overlayBitmap_ = null;
            break;
        }
        case 1: {
            overlayBitmap_ =
            BitmapFactory.decodeResource(context_.getResources(),
            R.drawable.mask_laugh);
            break; }
    }
}

```

Gambar 4.7 Kode untuk pilihan *mode*

Aktivitas deteksi wajah berlangsung pada kelas *FaceDetectThread* yang telah disebutkan dibagian awal. Pada kode dibawah ini Gambar 4.8, hasil deteksi digambarkan pada sebuah kanvas yang berada dilapisan *overlay*. Kondisi pendeteksian dapat dimulai saat `if(appMode_==0)`. Hasil tersebut divisualisasikan berupa kotakan hijau yang telah dideklarasikan pada kelas *OverlayLayer* dengan menginisialisasi `paint_.setColor(0xFF33FF33)` sebagai warna dasar hasil proses deteksi wajah yakni warna hijau. Apabila kondisi `if(overlayBitmap_!=null)` itu terpenuhi, akan menjalankan mode lainnya yang terdapat pada potongan kode Gambar 4.7. Sehingga saat wajah terdeteksi akan muncul tampilan *augmented reality* berupa masker yang menutupi area wajah.

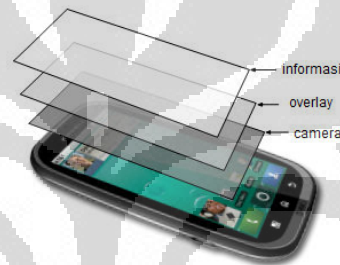
```

/* onDraw - Draw Face rect */
@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    int w = canvas.getWidth();
    int h = canvas.getHeight();
    if(fdtmodeBitmap_!=null){
        int x = w-100;
        int y = 10;
    }
}

```


rekognisi wajah berupa JSON. JSON inilah yang dimanfaatkan sebagai informasi utuh dari proses rekognisi wajah.

Pengembangan augmented reality pada skripsi ini diawali dari proses deteksi wajah. Hampir serupa dengan modul terpisah yang sebelumnya telah dijelaskan, bahwa aktivitas yang terjadi akan berlangsung dengan begitu cepat sehingga pengaturan kerja antar lapisan harus diperjelas agar tipis perbedaan antara lingkungan nyata dan objek *virtual* yang ditanam. Aktivitas utama yang mengatur tampilan yang nampak ke pengguna terletak pada *PreviewView*. Gambar di bawah ini akan memberikan visualisasi kerja dari *Augmented Reality* pada Android



Gambar 4.9 Antarmuka dari tiap lapisan *Augmented Reality*

Lapisan paling bawah mengarah pada objek yang tertangkap langsung oleh kamera. Apabila program dijalankan maka aktivitas akan dilanjutkan dengan deteksi wajah yang dilakukan pada kelas *OverlayLayer*. Pada kelas tersebut terdapat fungsi penggambaran pada *Canvas* sebagai lapisan di atasnya.

```

/* Digunakan untuk menampilkan wajah yang terdeteksi */
@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    Paint paint = new Paint();
    paint.setColor(Color.BLUE);
    paint.setTextSize(15);
    paint.setStrokeWidth(4);

    Paint paint2 = new Paint();
    paint2.setColor(Color.RED);
    paint2.setTextSize(15);

    String s = "Visual Tracking Wajah";
    float textWidth = paint.measureText(s);
    canvas.drawText(s, (getWidth()-textWidth)/2, 20,paint);
    int w = canvas.getWidth();
    int h = canvas.getHeight();

```


Android menyediakan fungsi multimedia yang komprehensif meliputi pengolahan *image*, *record*, *play*, dan *music*. Pada potongan kode di atas dilakukan penggambaran objek pada kanvas yang menggunakan method `onDraw(Canvas canvas)`. Kanvas yang sudah disediakan akan diisi dengan teks sebagai informasi awal hasil penggambaran melibatkan `canvas.drawText`. Dengan demikian tampilan awal dimunculkan dengan objek kamera yang berjalan disisipi dengan teks statik pada permukaannya. Melalui pendeklarasian variabel dengan kode `paint.setColor(Color.BLUE)` maka teks tersebut diatur dengan warna biru. Saat fungsi `if(fdtmodeBitmap_!=null)` yang menyatakan bahwa terdapat wajah yang terdeteksi. Maka kondisi tersebut menunggu respon dari pengguna. Respon tersebut diletakkan pada bagian pojok kanan bawah dengan nama *tombol*, melalui penggambaran objek bitmap pada canvas melalui `canvas.drawBitmap`. Secara lokal objek gambar bernama *tombol* diletakkan pada folder *res/drawable* dan untuk mengaksesnya digunakan *resources identifier* yaitu `R.drawable.tombol` pada Gambar 4.10.

```

//Coordinate for recognize button
Rect tombol = new Rect(w-64,h-64,w,h);
if( rect.contains(touchPt.x,touchPt.y)){
    if(!isSDCardPresent_)
        Toast.makeText(context_,
R.string.SDCardNotPresentAlert, Toast.LENGTH_LONG).show();
    else{
        takingPicture_ = true;
        selFacePt_ = new
PointF((float)touchPt.x/w,(float)touchPt.y/h);
    }
    break;
}
else if(tombol.contains(touchPt.x,
touchPt.y)){
    stopwatch.start();
    takePicture();
}
}
/* call autofocus. */
camera_.autoFocus(this);
return true;
}
return false; }

```

Gambar 4.11 Kode penentuan posisi objek pada *canvas*

Dari kode yang terdapat pada Gambar 4.11. Objek yang telah digambar pada kanvas, diletakan pada koordinat $w-64, h-64, w, h$. Koordinat tersebut memberikan posisi objek yang berada di atas kanvas terletak dibagian pojok kanan. Apabila objek tersebut disentuh, maka kamera melakukan pemfokusan pada wajah. Selanjutnya bila objek *tombol* diberikan aksi maka terjadi aktivitas rekognisi.

```

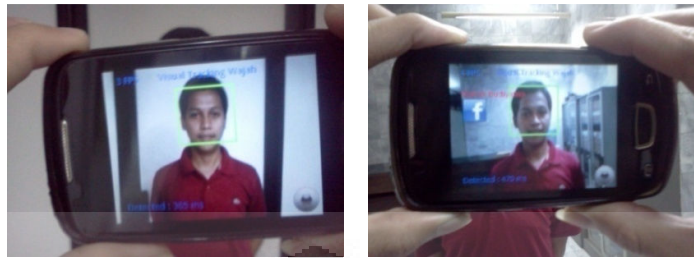
public boolean onTouchEvent(MotionEvent event) {
    /* do not handle if already going to take picture */
    if(takingPicture_)
        return false;
    /* detect touch down */
    if(event.getAction()==MotionEvent.ACTION_DOWN){
        int w = this.getWidth();
        int h = this.getHeight();
        Point touchPt = new
Point((int)event.getRawX(),(int)event.getRawY());
        float xRatio = (float)w / previewWidth_;
        float yRatio = (float)h / previewHeight_;
        selFacePt_ = null;
        for(int i=0; i<MAX_FACE; i++){
            FaceResult face = faces_[i];
            float eyedist = face.eyesDistance()*xRatio;
            if(eyedist==0.0f)
                continue;
            PointF midEyes = new PointF();
            face.getMidPoint(midEyes);
            /* assume face rect is x3 size of eye distance each side */
            PointF lt = new PointF(midEyes.x*xRatio-
eyedist*1.5f,midEyes.y*yRatio-eyedist*1.5f);
            Rect rect = new
Rect((int)(lt.x),(int)(lt.y),(int)(lt.x+eyedist*3.0f),(int)(lt.y+eyedi
st*3.0f));

```

Gambar 4.12 Kelas *onTouchEvent* untuk aksi aktivitas rekognisi wajah.

Setelah *canvas* berada pada lapisan *overlay*, dengan demikian objek gambar sudah menyatu dan tampil ke permukaan layar sebagai lapisan menumpuk. Aksi tersebut dapat berlangsung dengan metode *onTouchEvent()* yang ada pada Gambar 4.12. Metode tersebut dijalankan setiap kali disentuh dengan parameter *MotionEvent*. Jika sentuhan telah diangkat dari layar dapat digunakan metode *getAction()*. Sentuhan ke layar tersebut akan membawa pengguna pada

proses rekognisi wajah. Berikut ini adalah tampilan *Augmented Reality* pada sistem pengenalan wajah yang tampak pada Gambar 4.13.



Gambar 4.13 Antarmuka Sistem Pengenalan Wajah berbasis *Augmented Reality*

Pada gambar 4.13 di atas menunjukkan proses awalan untuk rekognisi (bagian kiri) dan hasil pengenalan wajah berupa augmented reality (bagian kanan). Informasi yang tampil dalam bentuk augmented reality tersebut diperoleh melalui proses *parsing*. Proses *parsing* dilakukan pada respon JSON yang kembali pada perangkat android saat permintaan rekognisi berjalan. Respon tersebut diberikan oleh layanan komputasi awan seperti yang tampak pada gambar dibawah ini.

```

{
  photos: [
    {
      url: http://face.com/images/ph/71934bf0ccdb3cf9055df506352fbd4d.jpg
      pid: "F8cd5c0242981cc3904ed2eb862ad086d4_4b4b4c6d54c37"
      width: 1024
      height: 768
      tags: [
        {
          tid: "TEMP_F8cd5c0242981cc3904ed2eb862ad086d4_4b4b4c6d54c37_52.64_40.76_1_1"
          recognizable: true
          threshold: 73
          uids: [
            {
              uid: "1584769417@facebook.com"
              confidence: 78
            }
            {
              uid: "1369347595@facebook.com"
              confidence: 57
            }
          ]
        }
      ]
    }
  ]
  status: "success"
}

```

Gambar 4.14 Contoh respon JSON hasil rekognisi dengan *Facebook*

Hasil rekognisi yang diterima masih berupa informasi yang belum dapat diidentifikasi oleh pengguna. Hasil rekognisi ditunjukkan dengan label berupa `uid:"1584xxxx@facebook.com"`. Setiap objek atau individu yang

terdaftar pada *facebook* memiliki identitas berupa ID. Atribut ID tersebut dapat dikenali dengan mengakses <https://graph.facebook.com/ID>. ID tersebut menjadi identitas yang unik untuk setiap informasi. Oleh karena itu dari hasil rekognisi yang diperoleh perlu dilakukan translasi uid kedalam notasi informasi yang dikenal dengan bantuan Graph API pada Facebook.

```
{
  "id": "1584xxxxx",
  "name": "Slamet Budiayatno",
  "first_name": "Slamet",
  "last_name": "Budiayatno",
  "link": "http://www.facebook.com/slamet.budiayatno",
  "username": "slamet.budiayatno",
  "birthday": "10/27/1990",
  "gender": "male",
  "location": {
    "id": "102173726491792",
    "name": "Jakarta, Indonesia"
  },
  "verified": true,
  "updated_time": "2011-12-24T15:01:59+0000",
  "type": "user",
}
```

Gambar 4.15 Respon JSON hasil translasi Graph API.

Respon JSON pada Gambar 4.14 diubah kedalam respon JSON lain yang memiliki informasi yang dapat diidentifikasi oleh pengguna, ditunjukkan oleh Gambar 4.15. Hanya saja tidak keseluruhan informasi tersebut ditampilkan secara utuh, melainkan informasi penting saja yang dimunculkan sebagai informasi hasil pengenalan wajah.

```
//Get Person Information based on facebook data
public String getPersonInformation(String uidFb){
    HttpURLConnection con;
    URL url;
    String payload = null;
    try {
        url = new URL(GRAPH_API_FACEBOOK + uidFb +
"&access_token=" + fbToken);
        con = (HttpURLConnection) url.openConnection();
        con.setReadTimeout(10000); // milliseconds
        con.setConnectTimeout(15000); // milliseconds

        con.setRequestMethod("GET");
        con.setDoInput(true);
        // Start the query
        con.connect();
    }
}
```

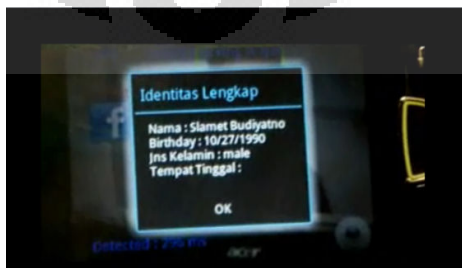
Gambar 4.16 Proses koneksi dengan *Graph API*

Kode di atas memberikan penjelasan mengenai pendahuluan proses untuk mendapat informasi facebook hasil translasi respon JSON Gambar 4.13. Pada kode di atas dilakukan konektivitas dengan <https://graph.facebook.com/ID> sebagai penyedia informasi.

```
// Read results from the query
    BufferedReader reader = new BufferedReader(new
InputStreamReader(con.getInputStream(), "UTF-8"));
    payload = reader.readLine();
    Log.i("DetectSave", payload);
    reader.close();
    //payload = new
JSONObject(payload).getString("status");
    resultName = new
JSONObject(payload).getString("name");
    resultBirthday = new
JSONObject(payload).getString("birthday");
    resultGender = new
JSONObject(payload).getString("gender");
    //String h = new
JSONObject(payload).getString("hometown").toString();
    resultHometown = new
JSONObject(payload).getString("hometown");
    String resultWork = new
JSONObject(payload).getJSONArray("work").getJSONObject(0).toString();
```

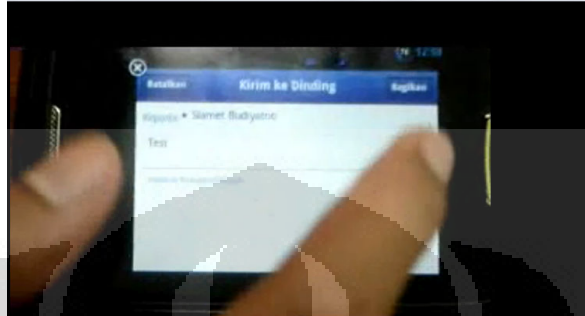
Gambar 4.17 Pengambilan beberapa informasi

Melalui potongan kode pada Gambar 4.17, dapat dipilih informasi apa saja yang akan diambil. Pada skripsi ini, diambil informasi JSON seseorang pada Gambar 4.15 yang terdiri atas nama, tanggal lahir, jenis kelamin, tempat tinggal dan pekerjaan. Namun informasi yang ditampilkan dalam bentuk *augmented reality* berupa nama saja. Untuk melihat informasi tambahan digunakan metode `OnTouchEvent` sebagai proses lanjutan interaksi. Apabila nama tersebut diberikan aksi maka dapat ditampilkan informasi tambahan lain seperti yang tampak pada gambar di bawah ini.



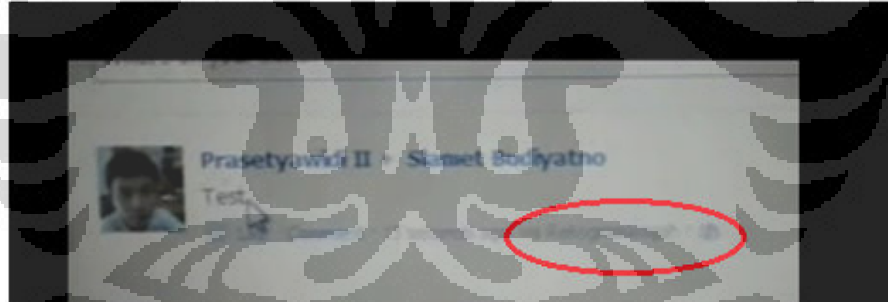
Gambar 4.18 Informasi lengkap hasil pengenalan wajah

Selain memperoleh informasi, pengguna juga dapat berinteraksi langsung dengan jejaring sosial seperti menulis pada dinding orang yang telah direkognisi. Hasil interaksi tersebut dapat dilihat pada gambar di bawah ini.



Gambar 4.19 Menulis pada *wall* orang yang dikenal.

Gambar 4.19 dan 4.20 merupakan ilustrasi hasil rekognisi oleh Prasetyawidi sebagai pengguna aplikasi, saat melakukan rekognisi wajah Slamet Budiyatno. Kemudian pengguna mampu memperoleh akses jejaring sosial untuk menulis sebuah tanggapan pada dinding orang yang dikenali tersebut.



Gambar 4.20 Hasil tulisan yang tampil pada Facebook.

BAB 5

PENGUJIAN DAN ANALISA MODUL *AUGMENTED REALITY* PADA PROSES PENGENALAN WAJAH

Pada bagian ini dibahas hasil pengujian dan analisa dari sistem yang telah diimplementasikan. Sistem telah berhasil ditanam pada perangkat android dengan versi 2.3 (*Android Gingerbread*), fungsionalitas dari sistem utuh pun sudah berjalan. Namun perlu dilakukan pengujian untuk mengetahui apa saja kekurangan dari sistem serta menemukan kesalahan (*error*) yang terjadi. Pengujian dan analisis ini dibagi menjadi beberapa bagian berdasarkan *translation time* (waktu translasi), tingkat pencahayaan untuk menampilkan informasi *Augmented Reality* dan pengujian kualitatif oleh responden.

5.1 Pengujian dan Analisis *Translation Time* pada *Augmented Reality*

Pada bagian pertama ini, akan dijelaskan hasil pengujian dan analisa tampilan *Augmented reality* berdasarkan waktu translasi. *Augmented reality* yang tertanam pada sistem utuh menampilkan hasil informasi pengenalan wajah. Hasil informasi tersebut diperoleh dari respon JSON yang dilayani *cloud server* ketika permintaan rekognisi wajah terjadi. Saat respon JSON tersebut dikembalikan, terdapat selang waktu yang membuatnya tampil dihadapan pengguna sebagai informasi utuh pengenalan wajah. Namun hasil keluaran tersebut belum masih dalam format JSON. Oleh karena itu perlu dilakukan translasi dari JSON kedalam bahasa manusia sebagai informasi dengan melakukan *parsing JSON*.

Waktu translasi pada skripsi ini dapat dinyatakan sebagai lamanya penerjemahan respon JSON yang dilayani *cloud server* untuk diolah sebagai informasi utuh hasil pengenalan wajah dan akumulasi *parsing JSON* untuk ditampilkan pada layar. Berdasarkan proses rekognisi yang terintegrasi dengan *facebook* hasil pengenalan wajah masih berupa `uid@facebook.com`, sehingga dengan bantuan Graph API Facebook informasi utuh atas ID tersebut dapat diketahui. Penerjemahan `uid` mampu menghasilkan informasi diri seseorang

yang dinotasikan dalam format JSON dengan beberapa izin akses informasi. Tidak semua informasi yang diperoleh akan ditampilkan, oleh karena itu perlu dilakukan pembatasan akses informasi. Pada pengujian ini dilakukan pengukuran lamanya waktu translasi berdasarkan izin akses informasi. Terdapat dua jenis izin akses informasi yang akan dibandingkan yakni izin akses informasi terdiri atas biodata diri, *e-mail*, lokasi, *website*, tempat tinggal, pendidikan, pekerjaan, foto, dan tanggal kelahiran orang yang akan dikenal dan izin akses informasi berupa biodata diri seseorang yang akan dikenal. Pengujian ini dilakukan untuk mengetahui pengaruh atas banyak informasi yang diambil dan lamanya waktu translasi JSON yang akan terjadi.

Pengujian dilakukan pada kondisi pencahayaan normal menggunakan sepuluh sampel wajah bagian depan dan dilakukan proses rekognisi sebanyak 10 kali percobaan untuk masing-masing wajah dengan ekspresi yang berbeda. Hal ini dimaksudkan untuk memaksimalkan hasil rekognisi wajah yang didapat serta memperoleh informasi yang lebih detail atas wajah yang ingin dikenali. Tabel 5.1 dan Tabel 5.2 menunjukkan hasil pengujian waktu translasi yang dilakukan untuk variasi 9 akses informasi dan 1 akses informasi.

Tabel 5.1 Waktu Translasi 9 akses informasi

Nama Wajah	Percobaan ke - (ms)										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
Arie	1206	1168	1202	1490	1708	1087	940	1306	952	991	1205.0
Asep	1775	1130	867	973	979	1717	991	1166	1128	1390	1211.6
Dahlan	1160	1341	1405	1872	1253	1126	952	1298	1020	1335	1276.2
Helmi	838	743	793	853	700	983	808	1247	1531	1383	987.9
Ian	832	1974	764	897	811	1029	929	978	977	1341	1053.2
Irvanda	813	703	858	739	1826	1091	1109	1362	1375	1136	1101.2
Pras	1384	816	1301	702	614	739	981	775	956	748	901.6
Ridho	715	656	1025	644	622	1566	736	630	795	669	805.8
Maun	701	873	671	727	617	887	752	1138	794	748	790.8
Yoga	990	781	1512	1189	815	754	862	682	710	914	920.9

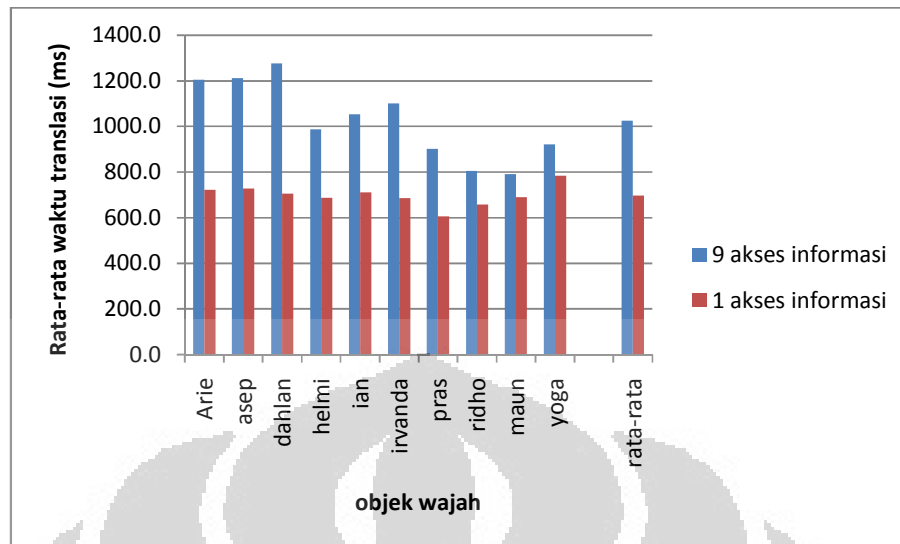
Tabel 5.2 Waktu Translasi 1 akses informasi.

Nama Wajah	Percobaan ke - (ms)										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
Arie	546	638	718	707	751	678	801	860	970	551	722
Asep	650	781	638	659	680	781	804	789	921	576	727.9
Dahlan	581	578	932	695	624	609	719	970	678	663	704.9
Helmi	609	607	732	731	849	746	592	668	699	640	687.3
Ian	694	687	665	651	686	695	641	780	772	833	710.4
irvanda	732	683	617	748	739	609	706	564	732	720	685
pras	602	615	617	716	582	640	549	562	555	625	606.3
ridho	613	615	634	626	682	795	677	616	620	705	658.3
maun	842	747	650	576	744	593	718	794	622	624	691
yoga	804	676	749	719	802	808	812	866	788	817	784.1

Berdasarkan Tabel 5.1 dan 5.2 yang diperoleh sebelumnya, dihasilkan sebuah rata-rata waktu translasi yang ditampilkan pada Tabel 5.3 dibawah ini.

Tabel 5.3 Rataan Waktu Translasi JSON

Objek Wajah	Rata-Rata Waktu Translasi 9 akses informasi (ms)	Rata-Rata Waktu Translasi 1 akses informasi (ms)
Arie	1205.00	722
asep	1211.60	727.9
dahlan	1276.20	704.9
helmi	987.90	687.3
ian	1053.20	710.4
irvanda	1101.20	685
pras	901.60	606.3
ridho	805.80	658.3
maun	790.80	691
yoga	920.90	784.1
Rata-rata	1025.42	697.7



Gambar 5.1 Rata-Rata Waktu Translasi.

Pada Tabel 5.3 dan Gambar 5.1 di atas terlihat bahwa terdapat perbedaan waktu translasi hasil pengenalan wajah dalam bentuk *augmented reality* dengan variasi informasi yang akan diakses. Dari gambar tersebut diperoleh waktu translasi atas 9 izin akses informasi yang berisi data yang lengkap mengenai pemilik wajah yang dikenal jauh lebih lambat bila dibandingkan dengan 1 izin akses informasi untuk menampilkan biodata diri. Dalam menampilkan informasi lengkap waktu translasi berlangsung selama 1025.42 ms, sedangkan untuk menampilkan biodata diri saja waktu translasi menghabiskan waktu 697.7 ms.

Hal ini terjadi karena sistem bekerja dengan melakukan pendekatan awal berupa deteksi wajah dan selanjutnya menjalankan proses pengenalan wajah. Kondisi yang terjadi saat pengujian pun mengabaikan nilai pengenalan wajah yang dikeluarkan baik itu sesuai atau tidak dengan citra wajah masukan. Penentuan area wajah mempengaruhi waktu proses pengenalan wajah karena wajah hasil deteksi akan dijadikan sebagai marker utama dalam pengenalan wajah. Jika terdapat objek-objek yang menghalangi area wajah seperti kerudung atau rambut yang menutup dahi, maka algoritma rekognisi wajah akan memberikan performa yang kurang sesuai sehingga waktu rekognisi pun tidak sama untuk setiap waktunya. Hal ini berimbas pada proses translasi yang akan berjalan.

Selain disebabkan oleh pengaruh proses rekognisi, perlu dicermati bahwa informasi yang akan ditampilkan atas permintaan pun berbeda. Banyaknya informasi yang di-*request* pengguna mengakibatkan parsing JSON akan lebih lambat. Pertukaran data JSON yang terjadi saat informasi lengkap, memiliki ukuran JSON yang lebih besar meskipun hanya berbeda ratusan *bytes* bila dibandingkan dengan JSON untuk biodata diri saja saat akan dikirimkan ke pengguna. Hal tersebut membuat timbulnya perbedaan waktu translasi pada kedua variasi akses informasi.

Proses translasi akan berjalan saat citra wajah masukan melewati proses rekognisi. Hasil rekognisi akan mengeluarkan nilai pengenalan yang benar, salah dan tidak dikenal atas citra wajah masukan tersebut. Nilai pengenalan benar dinyatakan saat citra wajah masukan mendapat hasil keluaran rekognisi yang sesuai. Sedangkan nilai pengenalan salah dinyatakan ketika hasil rekognisi yang muncul tidak sesuai dengan citra masukan meskipun telah terdapat citra wajahnya pada data training. Kedua nilai tersebut memberikan sebuah keluaran berupa respon JSON. Berbeda dengan hasil tidak dikenal, karena citra wajah masukan tersebut memang tidak terdapat dalam data training wajah sehingga tidak mengeluarkan respon JSON. Respon JSON yang ada mengalami proses translasi sebagai bentuk penerjemahan informasi yang ditampilkan ke layar dalam bentuk Augmented Reality. Informasi yang diambil tidak semuanya terpakai untuk ditampilkan. Oleh karena itu, banyak tidaknya respon JSON yang diminta dari sisi *client* akan berpengaruh saat dikirimkan kembali *client* bila kondisi jaringan tidak stabil. Meskipun terdapat pengaruh konektivitas namun faktor ini tidak dapat dijadikan fokus utama.

Berdasarkan pengujian pada Subbab ini diperoleh hasil yang menarik. Augmented Reality pada sistem ini menampilkan informasi hasil pengenalan wajah. Untuk mencapai kondisi *real-time* waktu translasi 697.7 ms sangat cocok, hanya saja terdapat keterbatasan informasi yang diperoleh tentang pemilik wajah. Berbeda halnya apabila tujuan utamanya memperoleh informasi lengkap. Waktu akan jauh lebih lama, namun kondisi tersebut membuat tampilan Augmented Reality dalam mencapai karakteristik *real-time* menjadi sedikit terganggu.

5.2 Pengujian dan Analisa Performa *Augmented Reality* berdasarkan Tingkat Pencahayaan

Pada bagian ini dibahas hasil pengujian dan analisis menggunakan variabel yang ada disekitar kita yaitu pencahayaan. Pengujian ini ditujukan untuk mengetahui pengaruh intensitas cahaya terhadap tingkat kesuksesan pengenalan wajah dan waktu translasi hasil pengenalan wajah dalam bentuk *augmented reality*. Pengujian ini dilakukan sebanyak satu kali untuk 10 wajah yang berbeda dan dengan jarak terbaik, serta pada kondisi didalam ruangan dengan pemilihan latar belakang area pengujian yang sederhana. Hal ini dimaksudkan guna memperoleh hasil deteksi wajah yang sempurna. Kondisi cahaya yang diuji akan diklasifikasikan berdasarkan tingkat pencahayaan redup, sedang dan terang.

Pengujian kondisi terang adalah kondisi pengujian yang dilakukan diluar ruangan dan terkena matahari. Kondisi sedang adalah kondisi pencahayaan di dalam ruangan tertutup dari cahaya matahari memanfaatkan lampu ruangan. Kondisi redup adalah kondisi pencahayaan di dalam ruangan tertutup dari cahaya matahari tanpa bantuan lampu penerang. Gambaran kondisi pengujian dapat dilihat pada Gambar 5.3, selanjutnya pada Tabel 5.3 dan Gambar 5.4 berikut ini menunjukkan hasil pengujian yang dilakukan.



(a) Kondisi redup, (b) Kondisi sedang
(c) Kondisi terang.

Gambar 5.2 Gambaran kondisi pencahayaan saat pengujian.

Tabel 5.4 Hasil Pengujian Variasi Pencahayaan pada Sistem Pengenalan Wajah berbasis *Augmented Reality*

Objek Wajah	Waktu translasi tingkat pencahayaan (ms)		
	Terang	Sedang	Redup
Arie	1077	929	1220
Asep	1211	843	1347
Dahlan	1058	1066	1117
Helmi	937	854	1321
Ian	989	707	1002
Irvanda	925	804	1126
Pras	1026	833	904
Ridho	949	870	1067
Maun	865	772	775
Yoga	1139	978	1012
Rata-rata	1017.600	865.600	1089.100

Berdasarkan nilai rata-rata yang diperoleh pada Tabel 5.3, dihasilkan Tabel 5.4 yang menyatakan keberhasilan *augmented reality* dalam mengeluarkan hasil informasi pengenalan wajah dengan variasi kondisi cahaya.

Tabel 5.5 Tabel Performa AR berdasar Pencahayaan

Tipe Pencahayaan	Persentase Pengenalan Wajah	Waktu translasi (ms)
Terang	80%	1017.6
Sedang	80%	865.6
Redup	30%	1089.1

Pada Tabel 5.4 di atas terlihat bahwa variasi tingkat pencahayaan sangat mempengaruhi tingkat deteksi wajah, akibatnya proses pengenalan wajah dan waktu translasi menjadi beragam saat tingkat pencahayaan divariasikan. Bahkan pada kondisi pencahayaan yang kurang wajah terdeteksi dengan samar-samar. Hal ini terjadi karena metode deteksi wajah tidak mampu membedakan antara wajah dan bukan wajah pada kondisi cahaya tersebut. Dengan demikian wajah tidak dapat digunakan sebagai marker utama untuk mengeluarkan *augmented reality* hasil pengenalan wajah yang sesuai.

Selain itu, secara umum dapat dilihat bahwa kondisi terang dan sedang keduanya memberikan hasil rekognisi yang baik dengan menghasilkan tingkat rekognisi yang sama sebesar 80 %. Artinya deteksi wajah diawal pemrosesan berjalan dengan lancar. Wajah pun dapat digunakan sebagai marker. Hal yang turut diperhatikan pula yakni waktu translasi dengan dua kondisi cahaya tersebut. Kondisi cahaya yang sedang menghabiskan waktu translasi yang jauh lebih cepat dibandingkan dengan saat cahaya terang.

Salah satu penyebabnya yakni objek wajah yang tertangkap oleh kamera saat wajah terdeteksi dan dilakukan proses rekognisi untuk dikirimkan ke *cloud* terkompresi menjadi citra JPEG. Dalam pengkodean yang ada pada sistem, citra potongan hasil deteksi wajah yang dikirim ke *cloud* dimampatkan dengan kualitas 90% dari citra asli. Citra tersebut tersimpan secara temporer pada lokasi */sdcard/DCIM/RekognisiWajah/*. Pemampatan citra dilakukan sebagai upaya mengoptimalkan kinerja sistem saat citra dikirimkan dan terkompresi di *cloud*. Berdasarkan citra temporer yang tersimpan, citra wajah pada kondisi terang memiliki ukuran sedikit lebih besar dari pada kondisi sedang. Beberapa area wajah seperti lekukan pada dahi dan cekungan pipi nampak agak kabur, kemungkinan sumber datangnya cahaya turut menjadi faktor. Secara kasat mata terdapat sedikit perbedaan citra wajah hasil kompresi dengan citra asli, namun pemampatan citra mengurangi kualitas citra serta menimbulkan degradasi warna pada citra meskipun tidak terlihat jelas oleh manusia. Akibatnya proses rekognisi yang terjadi berjalan dengan tempo waktu yang cukup lama.

Dari kondisi yang dinyatakan sebelumnya, pencahayaan yang terang justru mengakibatkan wajah hasil deteksi menjadi sedikit silau sehingga menjadi kabur ketika tertangkap oleh kamera, namun proses rekognisi masih dapat berjalan. Oleh karena itu, dalam proses deteksi wajah pencahayaan yang diberikan sebaiknya secukupnya saja namun tidak terlalu gelap. Pencahayaan yang baik akan mengakibatkan wajah lebih tampak jelas saat proses deteksi. Waktu translasi pun akan mendapatkan pengaruh saat proses rekognisi berjalan. Kemudian arah sumber datangnya cahaya turut menjadi hal yang patut dipertimbangkan dalam melakukan proses rekognisi wajah.

5.3 Pengujian Kualitatif

Selain menguji ketepatan atau kesuksesan tampilnya *augmented reality* dengan nilai rekognisi yang baik menggunakan metode kuantitatif yang telah dibahas pada Subbab sebelumnya. Maka pada Subbab ini akan dilakukan pengujian secara kualitatif dengan meminta 10 orang responden untuk melakukan uji coba sistem secara keseluruhan. Setelah responden mencoba, responden diminta mengisi borang yang didalamnya terdapat beberapa pertanyaan. Salah satu pertanyaan pada borang merupakan pertanyaan mengenai pendapat responden saat menggunakan sistem pengenalan wajah ini, dengan melakukan pengambilan wajah seseorang untuk dilakukan rekognisi wajah. Opsi yang dapat dipilih oleh responden adalah “Sangat setuju” dengan nilai 5, “Setuju” yang bernilai 4, “Agak setuju” yang bernilai 3, “Tidak setuju” yang bernilai 2, dan “Sangat tidak setuju” yang bernilai 1 pada interval kepercayaan 95 %. Persamaan untuk interval kepercayaan dapat dilihat dibawah ini.

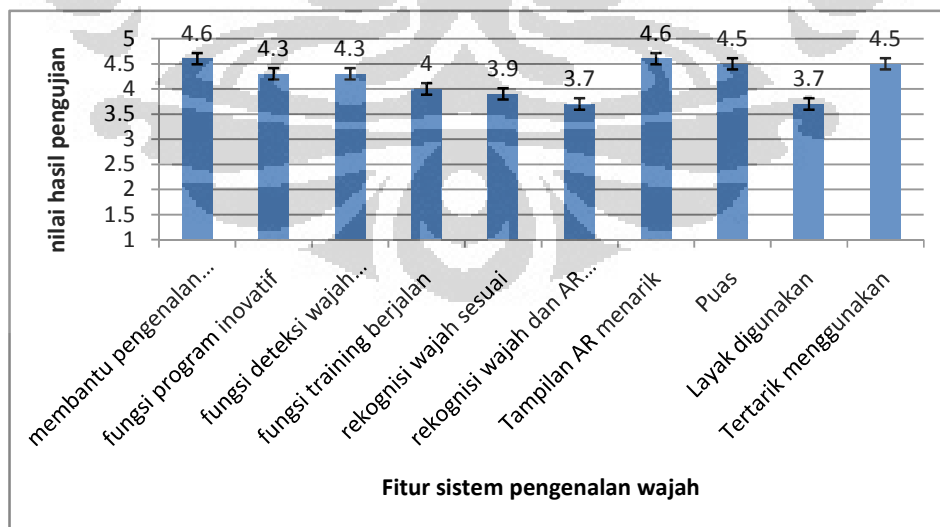
$$\text{Interval keyakinan 95\%} = \text{Rata - rata} \pm \left(\frac{1,96 \times \text{Standar Deviasi}}{\sqrt{\text{Populasi}}} \right) \quad (5.1)$$

Tabel 5.5 di bawah ini menunjukkan hasil pengujian responden saat menggunakan sistem pengenalan wajah dan *augmented reality*. Berikut adalah hasil pengujian rata-rata setelah dilakukan pengolahan.

Tabel 5.5 Hasil Pengujian Pengalaman Penggunaan Sistem oleh Responden

No.	Pernyataan	Perhitungan Interval Keyakinan 95 % Rata-Rata
1	Program membantu proses pengenalan wajah.	4.6
2	Fungsi-fungsi program inovatif	4.3
3	Pada fungsi deteksi wajah, hasil program berjalan sesuai fungsinya	4.3
4	Pada fungsi training wajah, program berjalan sesuai fungsinya.	4
5	Pada fungsi pengenalan wajah, program mampu mengenali objek wajah masukan	3.9
6	Pada fungsi pengenalan wajah dengan <i>augmented reality</i> , program mampu memberikan hasil yang sesuai.	3.7
7	Tampilan hasil pengenalan wajah dengan <i>Augmented Reality</i> lebih menarik	4.6
8	Anda puas menggunakan program ini	4.5
9	Aplikasi sudah layak digunakan banyak pengguna	3.7
10	Jika dipublikasikan, Anda tertarik menggunakan aplikasi ini	4.5

Berdasarkan tabel hasil pengujian yang telah dilakukan pada Tabel 5.5 dihasilkan grafik pada Gambar 5.5. Gambar 5.5 merupakan grafik pengujian yang dilakukan oleh sepuluh responden terpilih. Sumbu X menyatakan fitur yang ada pada borang pengujian dan sumbu Y menyatakan nilai rata-rata hasil pengujian.



Gambar 5.5. Grafik Hasil Pengalaman Penggunaan Sistem oleh Responden

Dari Gambar 5.5, pengujian terhadap persepsi yang kaitannya dengan fungsi rekognisi wajah dan *augmented reality* pada pernyataan 6 disetujui oleh pengguna dengan nilai rata-rata sebesar 3,7. Nilai tersebut menunjukkan bahwa kesesuaian fungsi pengenalan wajah dan *augmented reality* yang bekerja didalamnya sudah cukup baik untuk memenuhi ekspektasi pengguna. Setelah dilakukan pengolahan terhadap tanggapan responden dari persamaan 5.1 bila dikhususkan pada pernyataan 6, interval kepercayaan 95% yang didapat adalah $3,7 \pm 0.658$. Dari interval kepercayaan 95% dan lima opsi yang dapat dipilih responden, dapat dinyatakan responden cenderung setuju bahwa fungsi pengenalan wajah dan *augmented reality* sesuai. Ketidaksesuaian hasil pengenalan wajah yang terjadi mungkin diakibatkan oleh ketidaktepatan proses deteksi diawal pengujian, efeknya terjadi pergeseran posisi wajah yang membuat hasil rekognisi menjadi salah.

Pada pernyataan 7 terkait fungsi *augmented reality* dalam memberikan tampilan yang menarik disetujui pengguna dengan nilai rata-rata 4.6. Nilai tersebut merepresentasikan kepuasan pengguna saat konsep augmented reality diterapkan pada hasil pengenalan wajah. Pengolahan data memberikan hasil bahwa interval kepercayaan 95% yang didapat adalah 4.6 ± 0.447 . Hasil tersebut menunjukkan ekspektasi yang sesuai dengan harapan pengguna bahwa tampilan augmented reality mampu memberikan antarmuka yang lebih ramah bagi para pengguna.

BAB 6 PENUTUP

6.1 Kesimpulan

Berikut adalah kesimpulan dari penulisan skripsi ini :

1. Sistem pengenalan wajah (*Face Recognition*) merupakan suatu sistem yang dirancang untuk mengidentifikasi wajah seseorang melalui proses pencocokan dengan *database* wajah hasil *training*. Pemanfaatan layanan *Cloud Computing* pada skripsi ini, merupakan sebuah mekanisme yang memungkinkan penghematan sumber daya yang digunakan dalam hal pemrosesan wajah terutama disisi *client*.
2. Penggunaan sistem pengenalan wajah berbasis *Augmented Reality*, memberikan tampilan yang lebih menarik kepada pengguna. Pada pengujian sepuluh wajah dengan masing-masing percobaan sebanyak 10 kali, diperoleh waktu translasi *Augmented Reality* sebesar 1025.42 ms pada model informasi lengkap dan 697.7 ms untuk model informasi biodata saja. Pada kasus lain, pencahayaan menjadi faktor utama penentu hasil rekognisi wajah. Dengan kondisi cahaya yang sedang hasil rekognisi mendapatkan akurasi 80% dengan rata-rata waktu translasi 865.6 ms. Sedangkan kondisi cahaya yang terlalu terang atau terlalu redup mengakibatkan penggunaan wajah sebagai marker menjadi buruk, imbasnya akan terjadi dalam hasil rekognisi wajah dan waktu pemrosesan.
3. Pengujian berdasarkan hasil responden diperoleh tanggapan yang baik. Pengujian terhadap fungsi rekognisi wajah dan *augmented reality* mendapat interval kepercayaan 95% sebesar $3,7 \pm 0.658$. Hasil tersebut menyatakan responden cenderung setuju bahwa fungsi pengenalan wajah dan *augmented reality* sesuai. Pernyataan tampilan AR mampu memberikan tampilan menarik, ditanggapi responden dengan interval kepercayaan 95 sebesar 4.6 ± 0.447 . Hasil tersebut menunjukkan kesesuaian ekspektasi pengguna terhadap *augmented reality*.

6.2 Pengembangan kedepan

Perancangan sistem pengenalan wajah berbasis *cloud computing* memanfaatkan konsep *Augmented Reality* sebagai penampil informasi, diharapkan dapat mempermudah pengguna melakukan proses identifikasi terhadap wajah yang ingin dikenal. Implementasi yang tepat guna akan menjadi pedoman bagi setiap perancang aplikasi yang memanfaatkan *cloud computing* sebagai media efisiensi penggunaan sumber daya. Harapan selanjutnya, *Augmented Reality* akan menjadi teknologi pendukung utama dalam memberikan informasi dengan model yang interaktif.



DAFTAR REFERENSI

- [1] Kouma Jean-Paul, "Intelligent home security system", Thesis of Umea University, Swedia, 2006.
- [2] A.P. Miettinen and J.K. Nurminen, "Energy efficiency of mobile clients in cloud computing," HotCloud 2nd USENIX Workshop on Hot Topics in Cloud Computing, 2010.
- [3] Pramudya, Pudja, "Beginning Windows Azzure with PHP SDK", Bandung, Indonesia, 2010.
- [4] B.Kulawade, Vaibhav dkk 2011. *Project Report-Face Recogintion System*. Thesis Maharashtra State Board of Technical Education (MSBTE), Mumbai, 2011.
- [5] Face Detection and Face Recognition with Real-time Training from a Camera <http://www.shervinemami.info/faceRecognition.html>. Diakses pada 6 November 2011.
- [6] OpenCV Wiki. <http://opencv.willowgarage.com/wiki/>. Diakses pada 21 Oktober 2011.
- [7] Yang, M.H., Kriegman, D., Ahuja, N., 2002, "Detecting Faces in Images: A Survey" , IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 1.
- [8] Frederic P Miller, Agnes F Vandome, John McBrewster.(2009). *Augmented Reality*. VDM Publishing House.
- [9] Kim, Jaeyoung & Jun, Heesung, "Implementation of Image Processing and Augmented Reality Programs for Smart Mobile Device", *IEEE The 6th International Forum on Strategic Technology, Korea, 2011*.
- [10] Bimber, Oliver, & Ramesh, Raskar. (2005). *Spatial Augmented Reality*. Massachusetts : A K Peters.
- [11] Syaikhu, Akhmad, "Komputasi Awan (Cloud Computing) Perpustakaan Pertanian", *Jurnal Pustakawan Indonesia Volume 10 No.1*. Diakses pada 18 November 2011.
- [12] Joseph Katzman and Fred Donovan, "Head in the Clouds: DoD Turns to Cloud Computing," *Defense Industry Daily*. May 25, 2010, at:

- <http://www.defenseindustrydaily.com/defense-cloud-computing-06387/> .
Diakses pada 20 November 2011.
- [13] Lijun Mei, W.K. Chan, T.H. Tse, "A Tale of Clouds: Paradigm Comparisons and Some Thoughts on Research Issues," *apscc*, pp.464-469, 2008 *IEEE Asia-Pacific Services Computing Conference, 2008*.
- [14] Vincenzo D. Cunsolo, Salvatore Distefano, Antonio Puliafito, Marco Scarpa, "Volunteer Computing and Desktop Cloud: The Cloud@Home Paradigm," *nca*, pp.134-139, 2009 *Eighth IEEE International Symposium on Network Computing and Applications, 2009*.
- [15] Allan Carey, "Cloud Assurance Still Missing," *Information Assurance Newsletter*, Vol. 13, No. 1 (Winter 2010), 34.
- [16] Bradski, Gary, & Adrian Kaehler. (2008). *Learning OpenCV*. USA : O'Reilly Media.
- [17] Fowler, Martin. (2003). *UML Distilled: A Brief Guide to the Standard Object Modelling Language, Third Edition*. USA: Addison Wesley.
- [18] Muhammad Ridho, Nur. *Implementasi Sistem Pengenalan Wajah Sebagai Penghubung Jejaring Sosial : Layanan Komputasi Awan Untuk Fungsi Pengenalan Wajah Dengan Google App Engine Berbasis Python dan API Face.com*. Depok : Universitas Indonesia, 2012.
- [19] Indrawan, Prasetyawidi. *Implementasi Sistem Pengenalan Wajah Sebagai Penghubung Jejaring Sosial : Layanan Komputasi Awan pada Perangkat Mobile dengan Android Face Detector API dan Bantuan Komunikasi REST*. Depok : Universitas Indonesia, 2012.
- [20] Ableson, Frank. *Reuse existing C code with the Android NDK*. 12 April 2011.
- [21] Statistik Distribusi Penggunaan Platform Android Version per Maret 2012.
<http://developer.android.com/resources/dashboard/platform-versions.html>
Diakses pada 23 Mei 2012.
- [22] Daniel Wagner and Dieter Schmalstieg. Artoolkitplus for pose tracking on mobile devices. In CVWW'07: Proceedings of 12th Computer Vision Winter Workshop, hal 139-146, Graz University of Technology, Institute for Computer Graphics and Vision, February 2007. <http://www.icg.tu-graz.ac.at/Members/daniel/Publications/ARToolKitPlus>.

- [23] Patrick Brady. Anatomy & physiology of an android. Presentation, Google Inc. May 2008. <https://sites.google.com/site/io/anatomy--physiology-of-an-android>. Diakses pada 5 Juni 2012.
- [24] Domhan, Tobias. *Augmented Reality on Android Smartphones*. Stuttgart : Jerman 2010.

