



UNIVERSITAS INDONESIA

**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**

SKRIPSI

**NUR MUHAMMAD RIDHO
0806459854**

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



UNIVERSITAS INDONESIA

**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**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana

**NUR MUHAMMAD RIDHO
0806459854**

**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 : Nur Muhammad Ridho
NPM : 0806459854
Tanda Tangan : 
Tanggal : 2 Juli 2012



HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Nur Muhammad Ridho
NPM : 0806459854
Program Studi : Teknik Komputer
Judul : 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

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 Suidana M.Eng.

()

Penguji : Yan Maraden S.T., M.Sc.

()

Ditetapkan di : Depok

Tanggal : 2 Juli 2012

KATA PENGANTAR

Puji syukur penulis ucapkan atas syukur kepada Allah SWT, Tuhan Yang Maha Esa, yang telah memberikan berkah dan karunianya sehingga penulisan skripsi ini dapat diselesaikan dengan sebaik-baiknya. Penulis juga mengucapkan terima kasih kepada semua pihak yang telah membantu dalam penulisan skripsi ini, khususnya kepada:

1. Prof. Dr. Ir. Riri Fitri Sari, M.M., M.Sc. selaku pembimbing atas segala bimbingan, ilmu, serta arahan baik dalam penulisan skripsi maupun selama studi di Teknik Komputer Universitas Indonesia.
2. Orang tua dan keluarga penulis yang selalu memberikan segala dukungannya.
3. Teman-teman di Departemen Teknik Elektro Universitas Indonesia atas segala kerja samanya selama ini, terutama rekan satu tim Prasetyawidi dan Slamet Budiayatno.
4. Teman-teman yang telah menjadi relawan untuk pengambilan data skripsi ini, yaitu Arie Valdano, Ian Herahman, Asep Sunandar, Gradina Daru Adini, dan Rani Kumalasari.
5. Rizki Sabalilah dan keluarga yang telah memberikan dukungan, semangat, dan bantuannya.

Penulisan skripsi ini adalah sebagai salah satu syarat untuk mendapatkan kelulusan pada program studi Teknik Komputer Universitas Indonesia. Penulisan skripsi ini diharapkan mampu memberikan manfaat serta inspirasi bagi yang membaca, untuk bisa mengembangkan teknologi yang dibahas pada skripsi ini.

Depok, 9 Juli 2012

Penulis,

Nur Muhammad Ridho

NPM.0806459854

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Nur Muhammad Ridho
NPM : 0806459854
Program Studi : Teknik Komputer
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis Karya : Skripsi

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

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

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

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di: Depok

Pada tanggal: 9 Juli 2012

Yang menyatakan



Nur Muhammad Ridho

ABSTRAK

Nama : Nur Muhammad Ridho
Program Studi : Teknik Komputer
Judul : 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

Jejaring Sosial merupakan salah satu gaya hidup yang diminati saat ini. Perkembangan jejaring sosial ini dimulai sejak adanya teknologi Web 2.0, yang memungkinkan para pengembang Web mengembangkan situs yang tidak hanya sekedar memiliki konten, tetapi juga interaksi langsung dengan pengguna Web. Sampai saat ini sudah semakin banyak aplikasi dan layanan yang menyuguhkan tema jejaring sosial. Setiap layanan yang ada tersebut saat ini juga dapat saling berhubungan satu sama lain. Sistem ini dirancang untuk memberikan sebuah layanan yang menampung identitas pengenalan utama seseorang yang memiliki berbagai akun jejaring sosial, berupa identitas wajah. Perancangan modul pengenalan wajah ini diimplementasikan di layanan komputasi awan *Google App Engine* berbasis Python dan memanfaatkan API Face.com sebagai pengolah data citra. Selain itu sistem ini memanfaatkan *smartphone* berbasis Android sebagai divais interaksi dengan pengguna, dan juga menerapkan *Augmented Reality* sebagai penampil informasi kepada user. Hasil pengujian sistem ini bahwa modul ini dapat mengenali wajah dengan persentasi rata-rata kesuksesan sebesar 85%. Dengan kondisi pencahayaan gelap didapatkan persentase rata-rata kesuksesan 8%, dengan kondisi sedang 86%, dan kondisi terang 100%. Posisi sudut pandang maksimal untuk bisa melakukan pengenalan wajah adalah 60° dari posisi wajah tampak depan.

Kata Kunci: *face.com, google app engine, jejaring sosial, komputasi awan, python, pengenalan wajah*

ABSTRACT

Name : Nur Muhammad Ridho
Study Program : Computer Engineering
Title : Implementation of Face Recognition System as a Connector to Social Network : Cloud Computing Services for Face Recognition Function with Python-based Google App Engine and Face.com API

Social Networking has become today's lifestyle. Development of social networking started since the deployment of web 2.0 technology. It enables web developers to develop sites that do not only have a content, but also direct interaction with the users. Until now, there are so many applications and services which serve based on social network. The social networking applications also provides features for mutual interconnection between different social network applications. This system is designed to store a person's main identification which has various social networking accounts, in the form of face ID. We design a face recognition modules which will be implemented in the Python-based Google App Engine cloud computing services. We utilized the Face.com API as an image data processor. In addition this systems utilized Android-based Smartphone as an interaction devices with user. We applied the Augmented Reality as an information viewer to the users. The result of the implementation of this system shows that the module is able to recognize face samples with the average percentage of 85%. In the dark room condition, the average percentage is 8%, 86% on average condition, and bright condition is 100%. This module is able to recognize face samples from 0° (complete face) up to the maximum angle of 60°.

Keywords: *cloud computing, face.com, face recognition, google app engine, python, social network*

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR.....	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS.....	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL	xi
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Permasalahan	2
1.3. Tujuan Penelitian	2
1.4. Batasan Penelitian.....	2
1.5. Metodologi Penelitian.....	2
1.6. Sistematika Penulisan	3
BAB 2 LANDASAN TEORI	5
2.1. Jejaring Sosial	5
2.2. <i>Face Recognition</i>	6
2.2.1. Pendeteksian Wajah (<i>Face Detection</i>).....	7
2.2.2. Pengenalan Wajah (<i>Face Recognition</i>)	7
2.3. <i>Cloud Computing</i>	9
2.4. <i>Google App Engine</i>	12
2.5. <i>Application Programming Interface</i> (API) Face.com	13
2.6. Bahasa Pemrograman Python	14
BAB 3 PERANCANGAN MODUL PENGENALAN WAJAH DENGAN PLATFORM <i>GOOGLE APP ENGINE</i> BERBASIS PYTHON	16
3.1. Deskripsi Rancangan Umum	16
3.2. Diagram Alir Cara Kerja Program.....	18
3.3. Diagram-diagram Unified Modeling Language (UML).....	19
3.3.1. <i>Use Case Diagram</i>	20

3.3.2.	<i>Activity Diagram</i>	21
3.3.3.	<i>Sequence Diagram</i>	23
3.3.4.	<i>Deployment Diagram</i>	23
BAB 4 IMPLEMENTASI MODUL PENGENALAN WAJAH DENGAN PLATFORM GOOGLE APP ENGINE BERBASIS PYTHON DAN API FACE.COM		25
4.1.	Perangkat Pendukung Pengembangan Modul	25
4.2.	Pemrograman Modul Pengenalan Wajah dan Penjelasan Program	26
4.3.	Hasil Implementasi dan Penggunaan Modul	30
BAB 5 PENGUJIAN DAN ANALISA MODUL PENGENALAN WAJAH		32
5.1.	Rancangan Pengujian Modul Pengenalan Wajah	32
5.1.1.	Pengujian Tingkat Pengenalan Wajah dengan Variasi Jumlah Data Pelatihan	32
5.1.2.	Pengujian Tingkat Pengenalan Wajah dengan Variasi Tingkat Intensitas Cahaya.....	33
5.1.3.	Pengujian Tingkat Pengenalan Wajah dengan Variasi Posisi Sudut Pandang Wajah.....	34
5.2.	Hasil Pengujian dan Analisis	35
5.2.1.	Pengujian Tingkat Pengenalan Wajah dengan Variasi Jumlah Data Pelatihan	35
5.2.2.	Pengujian Tingkat Pengenalan Wajah dengan Variasi Tingkat Intensitas Cahaya.....	38
5.2.3.	Pengujian Tingkat Pengenalan Wajah dengan Variasi Posisi Sudut Pandang Wajah.....	40
BAB 6 PENUTUP		43
6.1.	Kesimpulan	43
6.2.	Rencana Pengembangan	43
DAFTAR REFERENSI		44

DAFTAR GAMBAR

Gambar 2.1 Alur Kerja Pengenalan Wajah [2]	7
Gambar 2.2 Tahap-tahap normalisasi [2].....	8
Gambar 3.1 Diagram Blok Sistem Kerja Aplikasi.....	17
Gambar 3.2 Diagram alur kerja program	18
Gambar 3.3 Diagram Alur Manajemen Pengguna.....	19
Gambar 3.4 <i>Use Case Diagram</i> Pengguna Baru.....	20
Gambar 3.5 <i>Use Case Diagram</i> Pengguna terdaftar.....	21
Gambar 3.6 <i>Activity Diagram</i>	22
Gambar 3.7 <i>Sequence Diagram</i>	23
Gambar 3.8 <i>Deployment Diagram</i>	24
Gambar 4.1 Algoritma Proses Pengenalan Wajah	26
Gambar 4.2 Diagram Blok Fungsi Pengenalan Wajah di <i>Google App Engine</i>	27
Gambar 4.3 Kode Untuk Membuat Alamat Unggah	28
Gambar 4.4 Kode Untuk Unggah dan Rekognisi	28
Gambar 4.5 Kode Untuk Unggah dan <i>Training</i>	29
Gambar 4.7 Tampilan Awal Web untuk Pengujian	30
Gambar 4.6 Contoh Respon JSON	30
Gambar 4.8 Halaman Hasil Pengenalan Citra.....	31
Gambar 5.1 Pengaruh Jumlah Data Pelatihan Terhadap Tingkat Pengenalan.....	36
Gambar 5.2 Pengaruh Jumlah Data Pelatihan dengan Tingkat Kesalahan	37
Gambar 5.3 Pengaruh Pencahayaan terhadap Kesuksesan Pengenalan Wajah	39
Gambar 5.4 Rata-rata Pengaruh Kondisi Pencahayaan terhadap Tingkat Pengenalan Wajah.....	39
Gambar 5.5 Pengaruh Sudut Pandang Pengambilan Citra Wajah dengan Tingkat Pengenalan Wajah.....	41

DAFTAR TABEL

Tabel 5.1 Sampel Wajah Untuk Percobaan	32
Tabel 5.2 Contoh Posisi Sudut Pengambilan Gambar	35
Tabel 5.3 Perbandingan Jumlah Data Pelatihan dengan Tingkat Pengenalan dan Tingkat Kesalahan.....	36
Tabel 5.4 Perbandingan Kondisi Cahaya dengan Tingkat Pengenalan.....	38



BAB 1

PENDAHULUAN

1.1. Latar Belakang

Identitas setiap orang saat ini tidak hanya dikenal dalam kehidupan nyata saja, tetapi juga dalam dunia maya. Bahkan saat ini sudah terdapat layanan yang menyediakan tempat untuk kita memiliki kehidupan lain di dunia maya seperti layanan *Second Life*. Identitas menjadi hal utama dalam penerapan layanan jejaring sosial. Setiap orang dapat memiliki sebuah akun di setiap layanan jejaring sosial. Setiap orang menjadikan identitasnya sebagai modal utama untuk mendapatkan relasi dan jaringan yang luas.

Pada saat ini sudah terdapat berbagai macam layanan jejaring sosial populer, seperti *Facebook*, *Twitter*, dan *Google+*. Hampir setiap orang yang menggunakan internet memiliki akun di salah satu bahkan lebih dari satu jenis layanan jejaring sosial tersebut. Selain itu berbagai layanan yang ada di internet telah banyak yang mengintegrasikan layanan mereka dengan layanan jejaring sosial. Hal tersebut disebabkan karena saat ini tren yang populer adalah mengarah ke inti dari jejaring sosial, yaitu berbagi. Hampir setiap layanan yang ada di internet saat ini pasti memberikan fasilitas kepada penggunanya untuk menghubungkan layanan mereka dengan layanan jejaring sosial yang populer terutama *Facebook* dan *Twitter*. Dengan keadaan seperti itu, maka dapat kita anggap setiap orang memiliki sebuah akun di berbagai layanan populer di internet. Berawal dari hal inilah, dirancang sebuah sistem yang dapat menghubungkan seseorang dengan berbagai layanan tersebut melalui salah satu identitasnya.

Sistem ini menggunakan wajah sebagai identitas utama seseorang. Hal tersebut karena pada umumnya kita mengenal seseorang berawal dari mengenal wajahnya. Sistem ini akan menjadikan identifikasi wajah untuk mengetahui berbagai layanan jejaring sosial dan aktifitas internet lainnya yang diikuti oleh orang tersebut.

1.2. Permasalahan

Pada tulisan ini, permasalahan yang menjadi fokus adalah perancangan modul pengenalan wajah, yang dilakukan dengan bantuan layanan komputasi awan, yaitu *Google App Engine* berbasis Python. Sistem pengolahan data pengenalan wajah akan dilakukan dengan bantuan API Face.com dan bahasa pemrograman Python.

1.3. Tujuan Penelitian

Tujuan penelitian yang tertuang dalam penulisan skripsi ini adalah untuk merancang dan mengimplementasikan modul pengenalan wajah dengan layanan komputasi awan *Google App Engine* berbasis Python dan API Face.com. Sistem ini diharapkan dapat mengenali wajah seseorang melalui divais yang dimiliki untuk selanjutnya dapat mengetahui informasi data diri orang tersebut. Selain itu, akan dilakukan pengujian terhadap sistem untuk mengetahui tingkat kemampuan dalam melakukan pengenalan wajah.

1.4. Batasan Penelitian

Permasalahan dalam penulisan skripsi ini dibatasi pada implementasi modul pengenalan wajah dengan layanan komputasi awan *Google App Engine* berbasis Python dan API Face.com. Pengujian yang dilakukan terhadap modul ini menggunakan data pelatihan wajah lima orang sampel, dengan masing-masing sampel memiliki sepuluh variasi mimik, ekspresi, sudut pandang dan kondisi pencahayaan. Data pengujian yang digunakan untuk proses rekognisi berbeda dengan data pelatihan. Pengujian proses rekognisi menggunakan data dengan kondisi acak.

1.5. Metodologi Penelitian

Metode penelitian yang digunakan dalam penulisan skripsi ini adalah studi rekayasa perangkat lunak. Sebuah perangkat lunak dibangun sebagai solusi dari permasalahan yang diangkat. Pada penelitian ini dibangun sebuah perangkat lunak untuk fungsi pengenalan wajah. Setelah sistem tersebut berhasil dibangun, dilakukan pengujian untuk mengetahui tingkat keberhasilan fungsinya.

Sesuai dengan kaidah rekayasa perangkat lunak, maka terdapat tahap-tahap dalam merancang sistem ini. Permasalahan yang diangkat merupakan hasil dari penelusuran kebutuhan dari sistem yang akan dibangun. Dari kebutuhan tersebut telah dilakukan perancangan fungsi-fungsi dari sistem ini. Perancangan sistem ini juga mencakup penggunaan diagram *Unified Modeling Language* (UML) sebagai standar dalam merancang sebuah perangkat lunak. Setelah perancangan selesai, maka dilakukan implementasi berupa pengembangan kode program yang dibuat. Setelah sistem ini berhasil dibangun, maka dilakukan pengujian terhadap kinerja yang dihasilkan.

1.6. Sistematika Penulisan

Sistematika dalam penulisan skripsi ini adalah sebagai berikut.

BAB 1 Pendahuluan

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

BAB 2 Landasan Teori

Pada Bab Landasan Teori ini berisi pengantar mengenai berbagai teori yang digunakan dalam penulisan skripsi ini, antara lain Face Recognition, API Face.com, Cloud Computing, *Google App Engine*, bahasa pemrograman Python.

BAB 3 Perancangan Modul Pengenalan Wajah dengan Platform *Google App Engine* Berbasis Python dan API Face.com

Pada Bab Perancangan membahas mengenai rancang bangun salah satu bagian dari sistem pengenalan wajah yang berisi rancangan cara kerja program, diagram alir program, dan diagram-diagram UML.

BAB 4 Implementasi Modul Pengenalan Wajah dengan Platform *Google App Engine* Berbasis Python dan API Face.com

Bab 4 ini membahas mengenai implementasi modul pengenalan wajah dan modul penyimpanan berkas citra yang diunggah. Bahasan implementasi

modul hanya mencakup implementasi modul pada platform *Google App Engine*.

BAB 5 Pengujian dan Analisa Modul Pengenalan Wajah

Bab 5 ini membahas hasil pengujian modul pengenalan wajah serta analisa dari hasil tersebut. Hasil pengujian didapatkan dari hasil pengujian teknis dan pengujian oleh sampel pengguna terhadap modul pengenalan wajah.

BAB 6 Penutup

Bab 6 berisi penutup, yaitu kesimpulan dan rencana pengembangan dari penulisan skripsi ini.



BAB 2

LANDASAN TEORI

2.1. Jejaring Sosial

Jejaring sosial merupakan layanan *online*, sebuah platform, ataupun situs yang memfokuskan layanannya pada hubungan antar orang di dunia maya. Jejaring sosial terdiri dari berbagai individual yang bergabung dan mempunyai identitas, serta membangun sebuah relasi dengan individual lainnya. Tujuan membangun relasi dengan individu lainnya tersebut adalah untuk berbagi minat dan aktifitas, yang tidak terbatas pada sebuah batas politik, ekonomi, dan geografi [1]. Dengan jejaring sosial ini setiap individu dari berbagai penjuru dunia dapat berinteraksi dengan mudah.

Pada jejaring sosial terdapat representasi dari data diri setiap pengguna yang tergabung dalam jejaring sosial tersebut. Ketika mendaftarkan diri ke dalam sebuah sistem jejaring sosial, pengguna akan diminta untuk memasukkan data-data tentang dirinya, mulai dari data diri utama seperti nama, tanggal lahir, dan sebagainya, hingga foto-foto dan berbagai macam media yang menggambarkan diri pengguna tersebut. Hal tersebut merupakan transaksi utama yang berjalan pada sistem jejaring sosial.

Potensi jaringan komputer untuk memfasilitasi sebuah bentuk baru dari interaksi sosial yang dibangun dengan menggunakan media komputer sudah dimulai sejak awalnya teknologi Web 2.0 diperkenalkan. Banyak *prototype* dari jejaring sosial yang telah menjadi layanan *online* seperti *America Online*, *Prodigy*, dan *CompuServe*. Jejaring sosial terdahulu yang ada pada *world wide web* dimulai pada komunitas *online* seperti *Theglobe* (1995), *Geocities* (1994), dan *Tripod* (1995) [1]. Kebanyakan dari layanan tersebut fokus kepada bagaimana untuk menghubungkan pengguna satu sama lain melalui *chat room* dan mengajak para pengguna berbagi informasi dan ide mereka melalui halaman web pribadi.

Generasi jejaring sosial berikutnya dimulai pada akhir dekade 1990 [1]. Para developer situs jejaring sosial memberikan berbagai fitur terbaru pada jejaring sosial seperti dapat mencari dan mengatur pertemanan. Fitur tersebut menjadi

metode utama dalam berlangsungnya sebuah jejaring sosial hingga saat ini. Situs-situs seperti Friendster (2002) yang populer di kalangan remaja Indonesia saat itu menerapkan manajemen pertemanan yang menjadi topik baru pada saat itu. Hingga akhirnya Facebook yang menjadi situs jejaring sosial terbesar di dunia yang diluncurkan pada tahun 2004 [1].

Facebook menjadi jejaring sosial yang lengkap menyediakan para penggunanya berbagai media untuk berkomunikasi dan berbagi informasi. Mulai dari koleksi album foto, membuat halaman khusus, pesan instan atau *chat*, sampai *video calling*. Fitur *tagging* pada album foto yang dimiliki Facebook membuat proses berbagi informasi dan momen menjadi lebih mudah. Kemudahan berbagi informasi ini yang membuat Facebook dapat dimanfaatkan untuk berbagai kegiatan sehari-hari. Banyak yang menggunakan Facebook sebagai media toko online yang menjual berbagai barang keperluan sehari-hari. Namun dengan kemudahan tersebut membuat penyalahgunaan fungsi Facebook semakin marak. Dengan kondisi tersebut maka dalam penggunaan layanan ini kita harus menyadari ancaman-ancaman tersebut.

2.2. *Face Recognition*

Face recognition atau pengenalan wajah merupakan bidang yang sangat aktif saat ini dalam hal *Computer Vision*, yang telah dipelajari selama 25 tahun dan saat ini telah menghasilkan berbagai macam aplikasi untuk kehidupan kita dalam bidang keamanan, robotika, interaksi komputer dan manusia, kamera digital, dan lain-lain.

Pengenalan wajah pada umumnya terdiri dari dua tahap [8] yaitu:

1. Pendeteksian wajah (*face detection*), merupakan tahap sistem mengenali bagian dari citra yang dianggap sebagai wilayah wajah.
2. Pengenalan wajah (*face recognition*), merupakan tahap sistem mendeteksi dan memproses citra wajah yang didapatkan untuk kemudian dibandingkan dengan *database* wajah yang telah dimiliki, untuk menentukan siapa pemilik wajah tersebut.

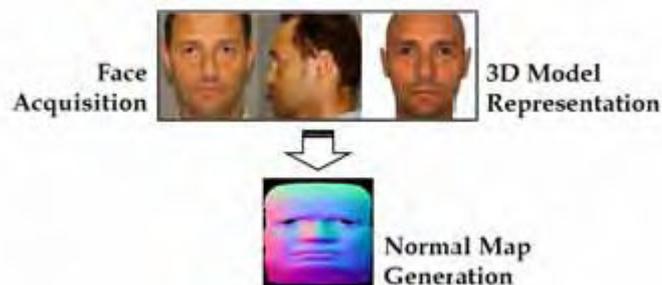
2.2.1. Pendeteksian Wajah (*Face Detection*)

Pada tahap pendeteksian wajah ini, sistem akan mendeteksi keberadaan wajah dalam sebuah citra dengan algoritma tertentu. Langkah ini bertujuan untuk menyederhanakan citra wajah yang akan dikenali, dengan menyeleksi area wajah yang ada pada citra tersebut. Ada beberapa teknik umum yang digunakan dalam mendeteksi sebuah citra wajah [15], diantaranya:

1. Menggunakan metode pengetahuan manusia untuk mengetahui keunikan dari struktur wajah manusia.
2. Menggunakan pendekatan invarian yang bertujuan untuk menemukan konfigurasi struktural dari sebuah wajah, yang akan digunakan untuk mendeteksi wajah.
3. Menggunakan sebuah *template* wajah yang menyimpan informasi sebuah wajah.
4. Menggunakan metode pembelajaran terhadap sistem kepada suatu set citra yang bervariasi, sehingga mendapatkan sebuah set data pendeteksian wajah.

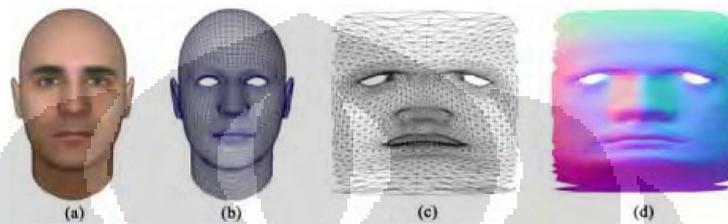
2.2.2. Pengenalan Wajah (*Face Recognition*)

Sistem pengenalan wajah ini merupakan tahap untuk mendapatkan informasi dari sebuah wajah melalui sebuah citra wajah yang ditangkap atau dengan *video streaming* [4]. Ide dasar dari pengenalan wajah ini adalah merepresentasikan sebuah permukaan wajah seseorang dengan *digital signature* yaitu pemetaan normalisasi [2]. Proses normalisasi ini adalah menjadikan sebuah gambar struktur wajah tiga dimensi menjadi citra RGB dua dimensi.



Gambar 2.1 Alur Kerja Pengenalan Wajah [2]

Ketika ingin melakukan normalisasi citra wajah, pertama adalah menjadikan citra wajah yang ditangkap ke dalam sebuah representasi model tiga dimensi. Setelah mendapatkan model 3D, maka model tersebut akan ditentukan koordinatnya dengan *wireframe model*. Setelah itu koordinat tiga dimensi tersebut akan diproyeksikan kepada bentuk dua dimensi dengan proyeksi bulat atau *spherical projection*. Dari proyeksi spasial tersebut akan dibentuk peta normalisasi yang akan digunakan untuk mengidentifikasi sebuah citra wajah.



Gambar 2.2 Tahap-tahap normalisasi [2]

Sejak 2002, pendeteksian wajah sudah dapat berjalan dengan reliabilitas yang baik, yaitu sekitar 90-95%, menggunakan modul pendeteksian wajah dari OpenCV [8]. Sebelumnya cukup sulit untuk mendeteksi wajah seseorang ketika dilihat dari berbagai sudut pandang dan sisi, bahkan terkadang membutuhkan estimasi tiga dimensi. Selain itu masih sulit untuk mendeteksi wajah ketika pencahayaan kurang, ataupun ketika wajah tersebut menggunakan aksesoris seperti kacamata. Bagaimanapun, pengenalan wajah merupakan hal yang lebih kecil reliabilitasnya yaitu hanya 30-70% keakurasiannya [8]. Namun hal tersebut menjadikan pengenalan wajah ini menjadi riset yang sangat aktif sejak era 1990 [10].

Ketika sebuah wajah telah terdeteksi, citra wajah tersebut dapat digunakan untuk pengenalan wajah. Pendeteksian wajah ini juga berguna untuk mendapatkan akurasi yang lebih tinggi, karena wajah telah dipisahkan dari wilayah lain dari citra yang diambil tersebut. Hal tersebut sangatlah penting untuk mengambil berbagai citra dengan sebuah standar yang akan diberikan kepada sistem pengenalan wajah.

Pada umumnya algoritma pengenalan wajah sangat sensitif terhadap kondisi pencahayaan, oleh karena itu diperlukan sebuah latihan untuk mengenali

seseorang dalam pencahayaan yang kurang maupun berlebih [8]. Selain itu posisi wajah juga sangat harus konsisten, seperti posisi mata yang harus sama koordinatnya, ukuran yang tetap, sudut rotasi, rambut, riasan, mimik wajah, posisi sumber cahaya, dan lain-lain. Hal ini menunjukkan bahwa sangatlah penting untuk menggunakan *filter* yang baik sebelum melakukan pengenalan wajah.

2.3. *Cloud Computing*

Komputasi awan (*Cloud Computing*) merupakan paradigma baru dalam dunia komputasi, yang menawarkan sumber daya komputasi (*compute*) dan penyimpanan (*storage*) secara massal dalam jumlah yang besar. Awan (*cloud*) di sini adalah sebuah sebutan umum dalam dunia jaringan yang mengacu pada jaringan internet. Dapat dikatakan bahwa komputasi awan ini adalah sistem komputasi yang menyerahkan berbagai aktivitas komputasi dan media penyimpanan kepada awan (jaringan internet) [5]. Layanan yang tersedia dalam komputasi awan ini dapat berupa layanan sebagai berikut.

- **Infrastruktur (*Infrastructure as a Services*)**, yaitu menyediakan layanan seperti server komputasi, penyimpanan data, *Firewall*, *Load Balancer*, dan infrastruktur lainnya.
- **Perangkat Lunak (*Software as a Services*)**, yaitu menyediakan layanan seperti *Office Suite*, *Social Network*, *Video Processing*, dan sistem perangkat lunak lainnya yang berbasis web.
- **Platform (*Platform as a Services*)**, yaitu menyediakan layanan untuk para developer sistem komputasi awan ini, seperti bahasa pemrograman, *frameworks*, *Mashup Editor*, *Structured Data*, dan platform lainnya.

Keunggulan dari komputasi awan ini adalah dalam hal penghematan energi dan biaya operasional [16]. Dengan penerapan komputasi awan dapat mengurangi penggunaan energi karena komputasi pada sisi *client* dapat diperkecil dan diserahkan kepada sisi *server*, sehingga tidak membutuhkan perangkat dengan spesifikasi tinggi yang memakan daya tinggi, bahkan dapat meningkatkan komputasi secara *mobile* [16].

Software as a Service atau SaaS merupakan layanan *Cloud Computing* yang paling dahulu populer. *Software as a Service* ini merupakan evolusi lebih lanjut dari konsep *Application Service Provider* (ASP). Sesuai namanya, SaaS memberikan kemudahan bagi pengguna untuk bisa memanfaatkan sumberdaya perangkat lunak dengan cara berlangganan, sehingga tidak perlu mengeluarkan investasi baik untuk *in-house development* ataupun pembelian lisensi. Dengan cara berlangganan via web, pengguna dapat langsung menggunakan berbagai fitur yang disediakan oleh penyedia layanan.

Hanya saja dengan konsep SaaS ini, pelanggan tidak memiliki kendali penuh atas aplikasi yang mereka sewa [5]. Hanya beberapa fitur aplikasi yang telah disediakan oleh penyedia saja yang dapat disewa oleh pelanggan. Selain itu arsitektur aplikasi SaaS yang bersifat *multitenant*, memaksa penyedia hanya menyediakan fitur yang bersifat umum, tidak spesifik terhadap kebutuhan pengguna tertentu. Meskipun demikian, kustomisasi tidak serta-merta diharamkan, meskipun hanya untuk skala dan fungsi yang terbatas. Tapi dengan berkembangnya pasar dan kemajuan teknologi pemrograman, berbagai keterbatasan itu pasti akan berkurang dalam waktu tidak lama.

Untuk contoh layanan SaaS, tentu saja kita harus menyebut layanan CRM *online* Salesforce.com. Selain itu juga ada Zoho.com, dengan harga yang sangat terjangkau, menyediakan layanan SaaS yang cukup beragam, dari mulai layanan *word processor* seperti Google Docs, *project management*, hingga *invoicing online*. Layanan akunting *online* pun tersedia, seperti yang diberikan oleh Xero.com dan masih banyak lagi. IBM dengan Lotuslive.com nya dapat dijadikan contoh untuk layanan SaaS di area kolaborasi atau *unified communication*. Untuk pasar dalam negeri sendiri, masih sangat sedikit yang ingin berinvestasi untuk menyediakan layanan SaaS ini.

Platform as a Service (PaaS) adalah layanan yang menyediakan modul-modul siap pakai yang dapat digunakan untuk mengembangkan sebuah aplikasi, yang tentu saja hanya bisa berjalan diatas platform tersebut. Seperti juga layanan SaaS, pengguna PaaS tidak memiliki kendali terhadap sumber daya komputasi dasar seperti memori, media penyimpanan, *processing power* dan lain-lain, yang

semuanya diatur oleh provider layanan ini. Pionir di area ini adalah Google AppEngine, yang menyediakan berbagai tools untuk mengembangkan aplikasi di atas platform Google, dengan menggunakan bahasa pemrograman Python dan Java. Microsoft juga mengeluarkan produk PaaS mereka yaitu Windows Azure yang mendukung bahasa C, C++, C#, ASP.NET dan lain-lain. Kemudian Salesforce juga menyediakan layanan PaaS melalui Force.com, menyediakan modul-modul untuk mengembangkan aplikasi di atas platform Salesforce yang menggunakan bahasa Apex. Selain itu mungkin yang jarang sekali kita ketahui bahwa Facebook juga bisa dianggap menyediakan layanan PaaS, yang memungkinkan kita untuk membuat aplikasi di atasnya.

Infrastructure as a Service (IaaS) yaitu IaaS terletak satu level lebih rendah dibanding PaaS. Ini adalah sebuah layanan yang menyewakan sumberdaya teknologi informasi dasar, yang meliputi media penyimpanan, *processing power*, memori, sistem operasi, kapasitas jaringan dan lain-lain, yang dapat digunakan oleh penyewa untuk menjalankan aplikasi yang dimilikinya. Model bisnisnya mirip dengan penyedia data center yang menyewakan ruangan untuk *co-location*, tapi kali ini lebih ke level mikronya. Penyewa tidak perlu tahu, dengan mesin apa dan bagaimana caranya penyedia layanan menyediakan layanan IaaS. Hal yang penting adalah permintaan atas sumber daya dasar teknologi informasi itu dapat dipenuhi. Perbedaan mendasar dengan layanan data center saat ini adalah IaaS memungkinkan pelanggan melakukan penambahan ataupun pengurangan kapasitas secara fleksibel dan otomatis.

Salah satu pionir dalam penyediaan IaaS ini adalah Amazon.com yang meluncurkan Amazon EC2 (*Elastic Computing Cloud*). Layanan Amazon EC2 ini menyediakan berbagai pilihan penyewaan mulai CPU, media penyimpanan, dilengkapi dengan sistem operasi dan juga platform pengembangan aplikasi yang bisa disewa dengan hitungan jam. Untuk jangkauan layanan, terbagi menjadi tiga yaitu *Public Cloud*, *Private Cloud* dan *Hybrid Cloud* [5].

Public Cloud, jenis *cloud* ini diperuntukkan untuk umum oleh penyedia layanannya. Beberapa layanan yang sudah disebutkan sebelumnya dapat dijadikan contoh dari *public cloud* ini.

Private Cloud, di mana sebuah infrastruktur layanan *cloud*, dioperasikan hanya untuk sebuah organisasi tertentu. Infrastruktur *cloud* itu bisa saja dikelola oleh organisasi itu atau oleh pihak ketiga. Lokasinya pun dapat dengan *on-site* ataupun *off-site*. Biasanya organisasi dengan skala besar saja yang mampu memiliki ataupun mengelola *private cloud* ini.

Hybrid Cloud, untuk jenis ini, infrastruktur *cloud* yang tersedia merupakan komposisi dari dua atau lebih infrastruktur *cloud* (*private*, *community*, atau *public*). Meskipun secara entitas mereka tetap berdiri sendiri-sendiri, tapi dihubungkan oleh suatu teknologi atau mekanisme yang memungkinkan portabilitas data dan aplikasi antar *cloud* itu. Misalnya, mekanisme *load balancing* yang antar-*cloud*, sehingga alokasi sumberdaya bisa dipertahankan pada level yang optimal.

2.4. **Google App Engine**

Google App Engine atau yang biasa disingkat GAE merupakan layanan *Platform as a Service*, *cloud computing platform* untuk membangun dan *hosting* aplikasi berbasis web kita di pusat data yang dikelola oleh Google [7]. Aplikasi yang dibangun akan divirtualisasi pada *multiple-servers*. *Google App Engine* ini memberikan sebuah layanan yang bersifat *on-demand*, yaitu sumber daya yang digunakan akan sesuai dengan kebutuhan dari pengguna. Jadi GAE ini mempunyai sistem yang otomatis untuk menyesuaikan sumber daya sistem untuk aplikasi yang dibangun terhadap pertambahan permintaan yang ada.

Pada saat ini bahasa pemrograman yang didukung adalah Python, Java, dan Go [9]. Selain itu juga mendukung bahasa pemrograman lain sebagai ekstensi seperti Groovy, Ruby, Scala, Clojure, Jython, dan PHP [9]. Pada bahasa pemrograman Python terdapat berbagai *web framework* yang dapat berjalan pada GAE ini diantaranya GAE Framework, Django, CherryPy, Pylons, Flask, web2py, dan webapp2, serta *framework* lain yang mendukung WSGI menggunakan CGI *adapter* [3].

Google App Engine ini adalah salah satu layanan platform komputasi awan yang menyediakan tipe layanan yang gratis dibanding dengan kompetitornya seperti Microsoft Azure dan Amazon WebServices. Layanan gratis ini tentu saja

memiliki batas-batas penggunaan dan spesifikasi infrastruktur. Dalam hal infrastruktur, batas-batas tersebut seperti ruang penyimpanan, memori, total waktu komputasi, total akses halaman web, dan lain-lain. Dalam hal penggunaan, batas-batas tersebut dalam hal hak akses kepada *file system*, eksekusi kode, fitur ekstra pada bahasa pemrograman, dan lain-lain [7].

Google App Engine berbasis Python memiliki dua versi, yaitu Python 2.5 dan Python 2.7. Perbedaan yang paling menonjol di antara keduanya adalah mengenai dukungan *library*. Pada Python versi terbaru, dukungan terhadap *library* baru lebih banyak. Fitur-fitur pada Python 2.7 ini juga lebih banyak dibandingkan dengan Python 2.5. Misalkan pada Python 2.7 sudah mendukung *framework* Django dengan versi yang lebih baru, penggunaan JSON yang tidak lagi menggunakan *simplejson*, dan lain sebagainya.

Untuk hal penyimpanan berkas dan basis data, *Google App Engine* memiliki layanan yang disebut *Datastore*. *Datastore* ini menyimpan objek data yang disebut *entities*. Sebuah entitas ini memiliki satu atau lebih *properties* yang memiliki satu atau lebih nilai. Nilai properti ini dapat berupa *integer*, *float*, *string*, *binary*, *blob*, dan lain-lain. Entitas *Datastore* ini dibuat dengan objek Python. Atribut dari objek tersebut menjadi *properties* dari entitas. Entitas dari *Datastore* ini tidak memiliki skema seperti *relational database* lainnya.

Datastore ini memiliki batas ukuran maksimum dari entitas yang akan dibuat, yaitu 1 MB. Oleh karena itu, *Google App Engine* juga menyediakan layanan untuk menyimpan sebuah data yang memiliki ukuran lebih dari 1 MB, yaitu *Blobstore*. *Blobstore* mengizinkan aplikasi yang dibuat untuk menyediakan data objek yang disebut *blobs*. *Blob* ini dibuat dengan mengunggah sebuah berkas melalui *HTTP request*. *Blob* pada *Blobstore* ini tidak berhubungan dengan *blob* pada properti *Datastore*.

2.5. Application Programming Interface (API) Face.com

Face.com merupakan layanan *Application Programming Interface* (API) untuk proses pengenalan wajah. Face.com dapat digolongkan ke dalam layanan komputasi awan *Platform as a Service* [11]. Face.com menyediakan API bagi para

developer untuk mengembangkan sebuah perangkat lunak yang menerapkan fungsi pengenalan wajah.

Face.com menggunakan teknologi algoritma *Labeled Faces in the Wild* (LFW). LFW merupakan database dari foto wajah yang didisain untuk mempelajari masalah dalam pengenalan wajah tak terbatas (*unconstrained*) [11]. Data yang digunakan mengandung lebih dari 13000 citra wajah yang dikumpulkan dari web. Face.com mengembangkan algoritma pengenalan wajah dari pembelajaran tentang LFW ini. Dengan mengembangkan LFW ini, maka Face.com dapat membuat sebuah API pengenalan wajah untuk jumlah yang besar.

Untuk menggunakan layanan Face.com, maka pengguna hanya perlu mendaftar. Setelah mendaftar, pengguna akan diberikan sebuah API_KEY dan API_SECRET. Fungsi dari API_KEY dan API_SECRET adalah sebagai kunci untuk mengakses API. Selain itu terdapat layanan untuk terhubung dengan Facebook dan Twitter. Dengan layanan tersebut, pengguna dapat membuat sebuah aplikasi di Facebook dan Twitter yang menggunakan API Face.com.

Proses pengenalan wajah dengan API Face.com juga seperti proses pengenalan wajah pada umumnya. Pertama, pengguna harus melakukan proses *pelatihan* terhadap wajah yang baru. Dalam proses pelatihan ini, ada tahap pendeteksian wajah dan menyimpan hasil deteksi tersebut. Setelah wajah tersebut telah berhasil melewati tahap pelatihan, maka wajah tersebut bisa dikenali oleh sistem.

2.6. Bahasa Pemrograman Python

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat kegunaan kode [6]. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Python mendukung multiparadigma pemrograman, namun tidak dibatasi. Python masuk ke dalam kategori pemrograman berorientasi objek, pemrograman interpretatif, dan pemrograman fungsional [6]. Salah satu fitur yang tersedia pada Python adalah sebagai bahasa pemrograman dinamis yang

dilengkapi dengan manajemen memori otomatis. Seperti halnya bahasa pemrograman dinamis lainnya, Python umumnya digunakan sebagai bahasa *script*, meski pada prakteknya penggunaan bahasa ini lebih luas. Python juga bisa digunakan sebagai *library* dalam sebuah aplikasi. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi, beberapa di antaranya adalah [6]:

- Linux/Unix
- Windows
- MacOSX
- Java Virtual Machine
- OS/2
- Symbian
- Palm

Python dikenal sebagai bahasa pemrograman interpreter, karena Python dieksekusi dengan sebuah interpreter. Terdapat dua cara untuk menggunakan interpreter, yaitu dengan mode baris perintah dan modus script. Pada mode baris perintah, Anda memanggil program Python dan sebuah interpreter langsung menampilkan hasilnya. Cara lain dari modus baris perintah adalah dengan menyimpan perintah-perintah python dalam satu file, yang disebut sebagai *script*. Pada umumnya file yang berisi *script* tersebut disimpan dengan ekstensi *.py.

BAB 3

PERANCANGAN MODUL PENGENALAN WAJAH DENGAN PLATFORM *GOOGLE APP ENGINE* BERBASIS PYTHON

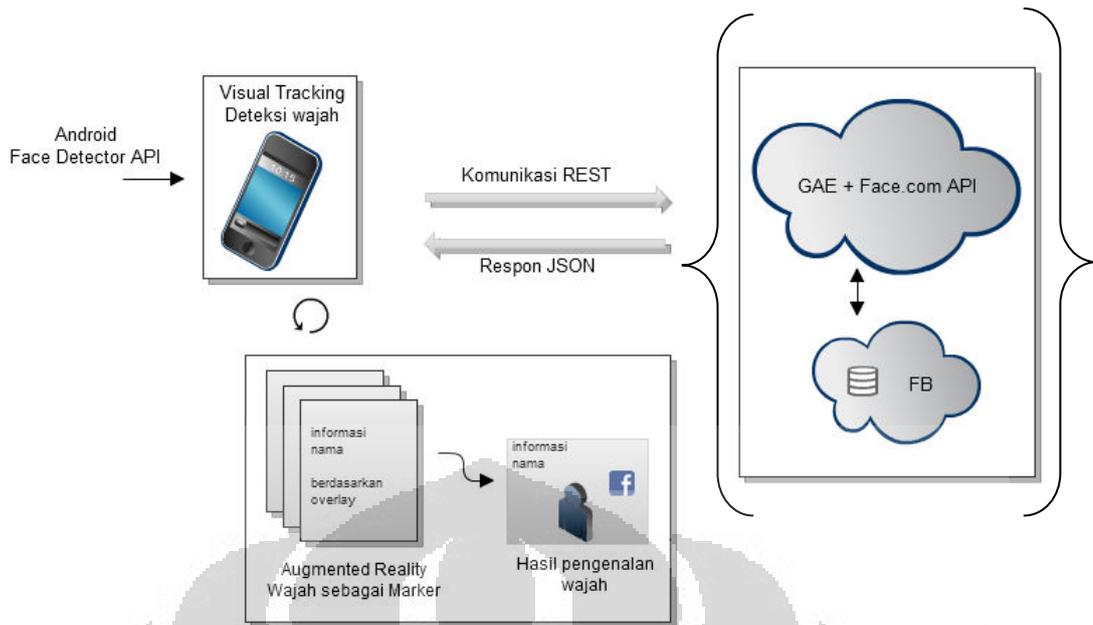
3.1. Deskripsi Rancangan Umum

Seperti yang telah dituliskan pada bagian awal tulisan ini, sistem ini akan dibangun pada dua *platform* yaitu pada layanan komputasi awan *Google App Engine* dan *smartphone* berbasis Android yang bekerja dengan saling terkoneksi. Dengan sistem ini pengguna akan dapat melakukan pengenalan wajah melalui divais Android yang dimilikinya.

Sistem akan melakukan pendeteksian wajah melalui divais Android. Aplikasi yang dibuat akan melakukan proses terhadap *video stream* yang diambil terhadap objek berupa wajah yang diarahkan pengguna sesuai operasi yang dipilih pada aplikasi. Direncanakan operasi yang dapat dilakukan oleh sistem ini adalah sebagai berikut.

- a. Melakukan fungsi pendeteksian wajah untuk mengetahui keberadaan posisi wajah.
- b. Melakukan pengambilan citra wajah untuk diproses sebagai pelatihan data wajah pada sistem.
- c. Melakukan pengenalan wajah sesuai dengan data yang telah ada di sistem dan memberikan informasi data kepada pengguna dalam bentuk objek yang menerapkan prinsip *Augmented Reality*.

Pengerjaan sistem ini dilakukan oleh sebuah tim yang terdiri atas Nur Muhammad Ridho, Slamet Budiayatno [14], dan Prasetyawidi Indrawan [13]. Sistem ini secara umum digambarkan pada gambar berikut:

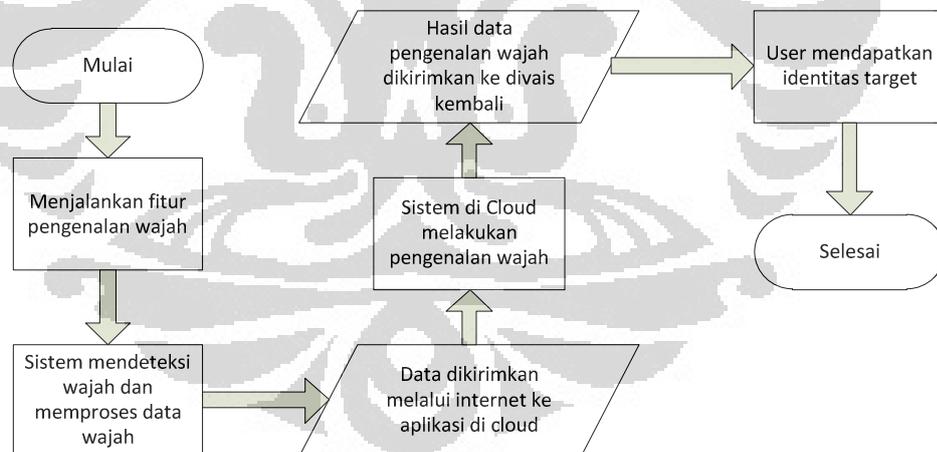


Gambar 3.1 Diagram Blok Sistem Kerja Aplikasi

Yang akan dibahas pada modul ini adalah perancangan fungsi pengenalan wajah yang dibangun pada platform komputasi awan *Google App Engine*, dengan bahasa pemrograman Python, dan API Face.com. Modul ini merupakan bagian dari keseluruhan sistem secara umum yang dirancang untuk melakukan pendeteksian wajah, pengambilan data wajah, dan pengenalan wajah. Secara singkat kerja dari modul ini adalah sebagai berikut. Dalam proses pendaftaran, pengguna akan menggunakan aplikasi Android dan mendaftar untuk pertama kali. Aplikasi Android akan langsung mengakses sistem yang dibangun di *Google App Engine*. Pengguna akan diminta untuk mengisi data personal dan data jejaring sosial yang digunakan. Setelah itu modul akan menghubungkan ke divais kamera yang ada sebagai input data wajah. Proses pertama yang dilakukan oleh modul adalah melakukan pendeteksian wajah untuk mencari posisi wajah yang akan diambil. Setelah itu data citra tersebut akan tersimpan di penyimpanan GAE. Dengan data citra tersebut, sistem akan melakukan pelatihan terhadap data-data citra yang ada dan menghubungkannya dengan data personal dari pengguna. Dalam proses pengenalan wajah, modul akan menerima gambar wajah yang diambil melalui divais kamera, untuk dicocokkan dengan data citra yang telah ada dalam sistem. Bila sesuai maka modul akan memberikan keluaran berupa informasi yang dimiliki oleh citra tersebut.

3.2. Diagram Alir Cara Kerja Program

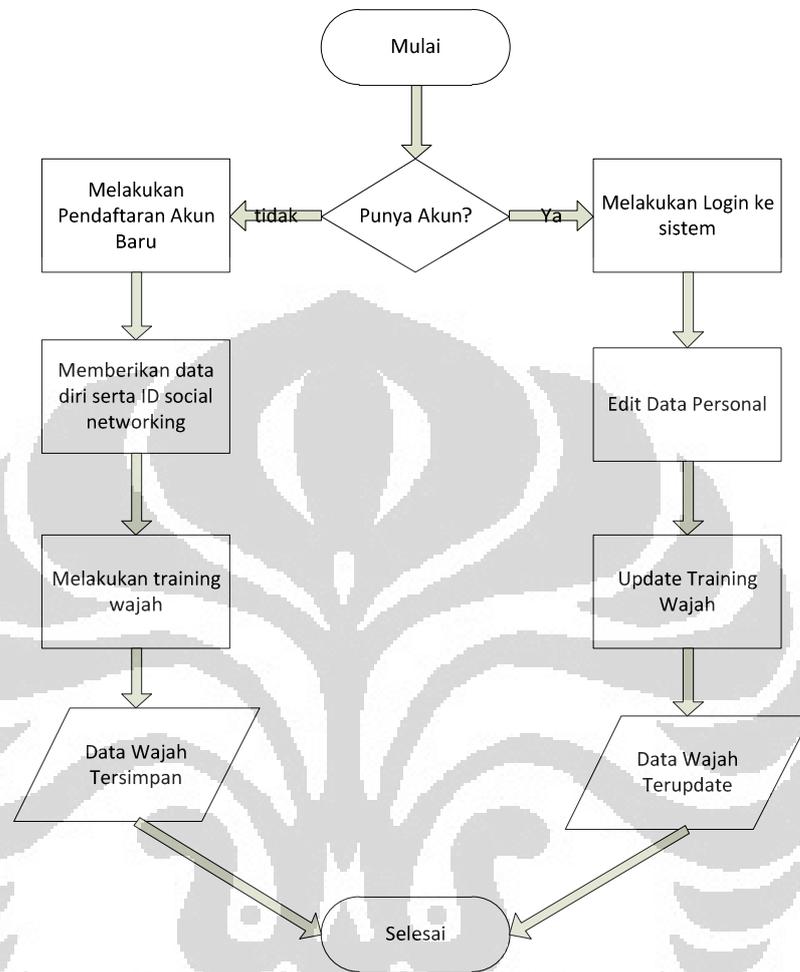
Program yang dibahas pada bagian ini adalah bagian dari keseluruhan sistem yang telah dijelaskan sebelumnya. Program ini adalah inti dari proses pengenalan wajah yang menjadi fungsi utama dari sistem ini. Program ini akan melakukan pengenalan citra wajah dari objek wajah yang dituju. Pertama program ini akan menerima masukan berupa *video stream* dari kamera. Kemudian program akan melakukan pendeteksian wajah untuk mengetahui lokasi wajah pada *viewfinder* kamera. Setelah itu program akan mengambil citra wajah sesuai dengan lokasi wajah yang telah terdeteksi. Citra wajah yang telah diambil tersebut akan diekstrak data-datanya dan disimpan. Ketika ingin melakukan pengenalan wajah, langkah yang ditempuh juga dimulai dengan mendeteksi lokasi wajah pada kamera. Setelah itu citra wajah diambil dan diproses. Dalam proses tersebut program akan mencocokkan data wajah yang tersimpan pada *database* dengan citra yang akan dikenali. Bila terdapat kecocokan dengan data yang ada, maka program akan memberikan keluaran berupa data personal orang yang dikenali wajahnya tersebut. Berikut adalah diagram alir jalannya program secara umum.



Gambar 3.2 Diagram alur kerja program

Dalam sistem keseluruhan, terdapat dua jenis tipe akses dalam sistem ini. Pertama adalah pengguna yang sudah teregistrasi dan pengguna umum. Pengguna umum hanya bisa untuk mengakses program pengenalan wajah, tetapi pengguna tersebut tidak bisa dikenali wajahnya oleh pengguna lain, karena belum terdaftar di dalam sistem. Pengguna yang sudah teregistrasi dapat melakukan pengenalan

wajah terhadap orang lain dan dirinya sendiri. Berikut adalah diagram alir manajemen pengguna pada sistem ini.



Gambar 3.3 Diagram Alur Manajemen Pengguna

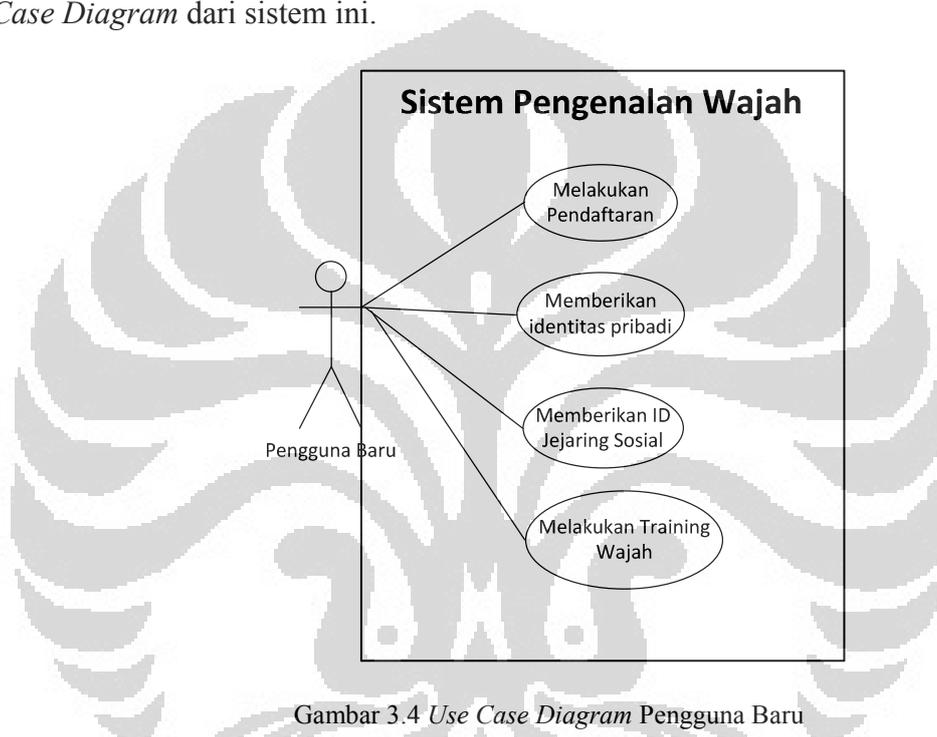
3.3. Diagram-diagram Unified Modeling Language (UML)

Unified Modelling Language (UML) merupakan sebuah metode untuk merepresentasikan atau mendeskripsikan desain perangkat lunak ke dalam notasi-notasi grafis yang terstandarisasi [12]. UML dengan notasinya yang universal memudahkan pengembang melakukan kolaborasi maupun mendokumentasikan rancangan perangkat lunaknya. Dengan UML, rancangan perangkat lunak dapat direpresentasikan ke dalam diagram-diagram yang memiliki fungsi masing-masing. Berikut adalah beberapa diagram yang merepresentasikan rancangan dari

modul yang akan dibuat, meliputi *use case diagram*, *activity diagram*, dan *sequence diagram*.

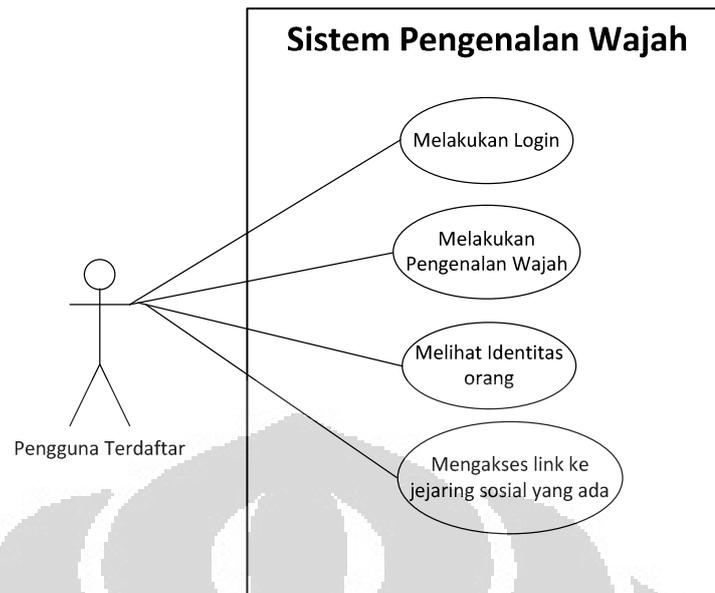
3.3.1. Use Case Diagram

Use Case Diagram digunakan untuk mewakili fungsionalitas dari sistem. Diagram ini menggambarkan interaksi antara pengguna, yang diwakili dengan notasi *actor*, dengan sistem untuk mencapai tujuannya [12]. Berikut adalah *Use Case Diagram* dari sistem ini.



Gambar 3.4 *Use Case Diagram* Pengguna Baru

Pada diagram *use case* di atas menjelaskan interaksi antara pengguna baru dengan sistem. Pengguna baru akan mendaftar ke dalam sistem untuk menggunakan aplikasi ini. Dalam pendaftaran tersebut pengguna akan mengisi pendaftaran dengan identitas pribadi dan identitas jejaring sosial yang diikutinya. Selain itu pengguna juga akan diminta untuk memberikan citra wajahnya untuk dikenali oleh sistem. Hal tersebut disebut sebagai pelatihan wajah untuk sistem.

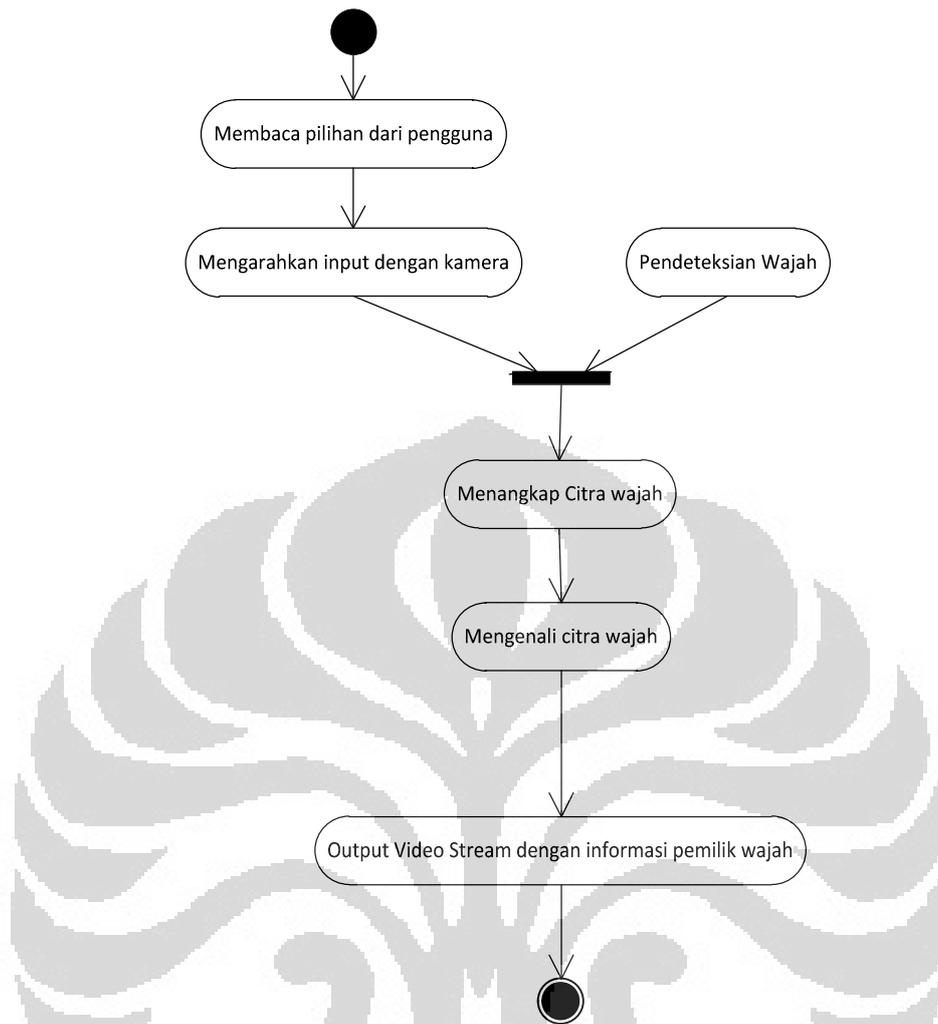


Gambar 3.5 Use Case Diagram Pengguna terdaftar

Pada *use case* di atas merupakan interaksi antara pengguna yang sudah terdaftar dengan sistem. Pengguna akan *login* ke dalam aplikasi untuk mengakses fitur di dalamnya. Pengguna dapat melakukan pengenalan wajah seseorang dengan mengarahkan kamera ke wajah orang tersebut. Setelah sistem melakukan pengenalan wajah, maka pengguna dapat melihat identitas orang yang dikenali tersebut. Selain itu pengguna juga dapat mengakses jejaring sosial yang orang tersebut gunakan.

3.3.2. Activity Diagram

Activity diagram merupakan diagram UML yang digunakan untuk merepresentasikan logika, *business process*, atau alur kerja program [12]. Berikut adalah gambar diagram aktifitas dari sistem ini.

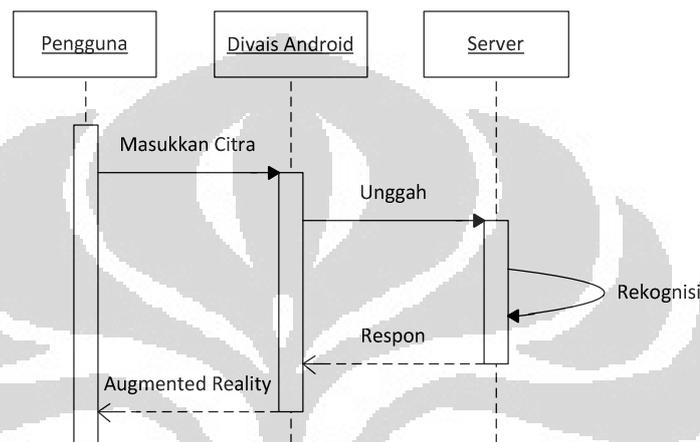


Gambar 3.6 Activity Diagram

Pertama, sistem akan membaca masukkan pilihan dari pengguna untuk melakukan pengenalan wajah. Selanjutnya aplikasi akan membawa menuju *interface* dengan *streaming* kamera. Ketika *streaming* kamera tersebut berjalan, sistem akan melakukan pendeteksian lokasi dari wajah. Setelah mendapatkan lokasi wajah, maka pengguna akan mengambil gambar wajah tersebut. Citra wajah yang ditangkap tersebut akan dikirim menuju *server* aplikasi untuk dilakukan pengenalan citra wajah. Setelah citra wajah tersebut dikenali, maka data akan dikirim kembali menuju sisi klien. Antar muka di sisi klien akan kembali dalam bentuk *streaming* kamera dan kembali seperti pada saat pendeteksian wajah, namun kali ini akan ditampilkan informasi orang yang dikenali dengan identitas jejaring sosialnya, menggunakan konsep *augmented reality*.

3.3.3. Sequence Diagram

Sequence Diagram termasuk salah satu *interaction diagram* yang digunakan untuk menggambarkan perilaku dari objek-objek yang terlibat dalam sistem dan pesan-pesan antar objek yang berada dalam *use case* [12]. Berikut adalah gambar dari *sequence diagram* dari sistem ini.



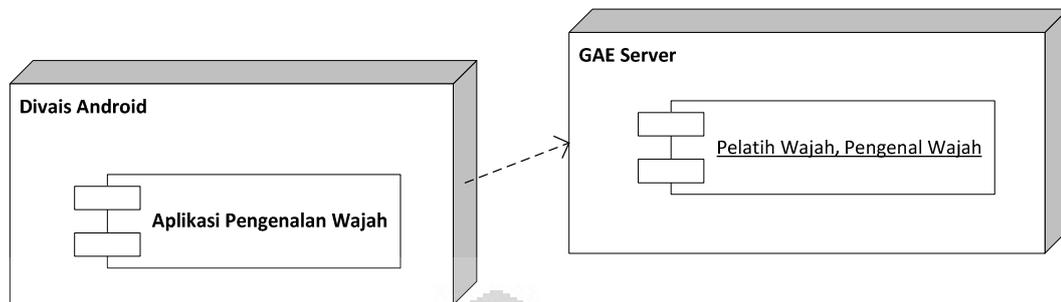
Gambar 3.7 *Sequence Diagram*

Gambar di atas merupakan *sequence diagram* dari sistem pengenalan wajah. Diagram tersebut menunjukkan terdapat tiga entitas yang berinteraksi, yaitu pengguna, divais, dan server modul pengenalan wajah. Divais Android atau web browser yang digunakan oleh pengguna akan mengambil gambar seseorang yang ingin dikenali sebagai masukkan yang akan diproses oleh modul pengenalan wajah. Modul akan mencocokkan dengan *database* yang ada, dan akan mengembalikan data wajah beserta informasi yang terkait dengannya ke divais pengguna.

3.3.4. Deployment Diagram

Deployment Diagram adalah diagram yang digunakan untuk merepresentasikan rancangan tata letak fisik dari sistem yang menunjukkan perangkat keras dan perangkat lunak dalam sistem tersebut [12]. Dalam sistem yang dibangun ini terdapat dua buah perangkat keras, yaitu divais Android pada

sisi klien dan server komputasi awan. Deployment Diagram ini dapat dilihat pada Gambar 3.8.



Gambar 3.8 *Deployment Diagram*



BAB 4

IMPLEMENTASI MODUL PENGENALAN WAJAH DENGAN PLATFORM *GOOGLE APP ENGINE* BERBASIS PYTHON DAN API FACE.COM

4.1. Perangkat Pendukung Pengembangan Modul

Implementasi modul pengenalan wajah ini dilakukan dengan bantuan perangkat lunak *Integrated Development Environment* (IDE), yang dapat membuat pengembangan menjadi lebih efektif dan efisien. IDE yang digunakan dalam pengembangan ini adalah Eclipse Indigo 3.7 dengan *plugin* PyDev, karena bahasa pemrograman yang digunakan adalah Python, dengan versi interpreter 2.7. Selain itu dibutuhkan sebuah SDK *Google App Engine* Python dengan versi 1.6.5 sebagai alat untuk mengunggah sistem yang dibangun ke server *Google App Engine*. Untuk pengolahan pengenalan citra wajah dengan API Face.com diperlukan *library* Python sebagai penghubung antara klien dengan server API.

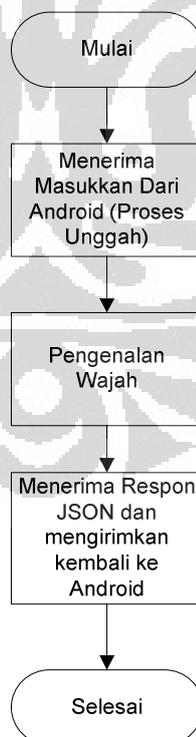
SDK *Google App Engine* yang digunakan dalam membangun sistem ini dapat digunakan sebagai server lokal untuk mencoba menjalankan aplikasi yang dibuat di komputer lokal. SDK *Google App Engine* yang digunakan dalam pengembangan sistem ini adalah versi Windows. Perbedaan versi Windows dengan versi Linux atau Mac OS adalah dalam hal *user interface*, karena dalam versi Windows memiliki *Graphic User Interface* (GUI) sehingga lebih mudah digunakan.

Library Python untuk API Face.com yang digunakan dalam pengembangan ini memiliki lisensi *Open Source*, sehingga dapat dikembangkan sendiri untuk menyesuaikan dengan sistem yang akan dibuat. Selain itu Face.com ini memiliki *library* dalam berbagai bahasa lain yang dikembangkan secara terbuka oleh berbagai pihak.

4.2. Pemrograman Modul Pengenalan Wajah dan Penjelasan Program

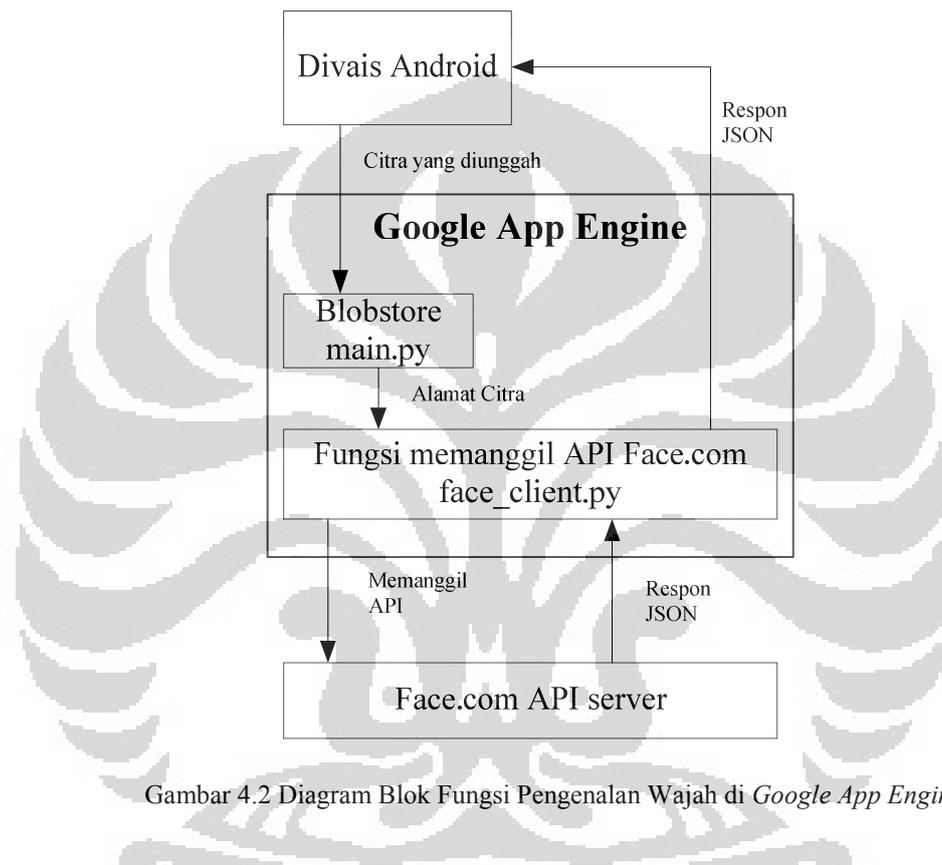
Pengembangan modul pengenalan wajah yang akan dijalankan pada server *Google App Engine*. Terdapat dua bagian, pertama fungsi penyimpan hasil tangkap citra wajah yang dilakukan oleh aplikasi pada divais Android, dan fungsi untuk pengenalan citra wajah yang diunggah tersebut dengan API Face.com. Untuk mempermudah pengujian secara per unit, maka untuk modul pengenalan wajah di *Google App Engine* ini dibuat antar-muka dalam bentuk website yang dapat diakses pada alamat <http://gaefacerecognition.appspot.com>.

Algoritma yang diterapkan pada modul pengenalan wajah ini dapat dilihat pada Gambar 4.1. Pertama, modul ini akan menerima berkas citra wajah yang diunggah oleh aplikasi Android. Dalam proses unggah ini, akan langsung dihubungkan dengan proses pengenalan wajah dengan API Face.com. Jadi setelah proses unggah selesai, maka berkas citra tersebut langsung dikenali dalam bentuk respon JSON. Respon JSON ini akan dikirimkan kembali ke aplikasi Android untuk ditampilkan dalam bentuk *augmented reality*.



Gambar 4.1 Algoritma Proses Pengenalan Wajah

Struktur program yang ada pada server *Google App Engine*, terdiri dari tiga bagian utama, yaitu *main.py*, *face_client.py*, dan *template*. Dalam *main.py* terdapat kelas-kelas yang menangani proses utama dari sistem ini. Skrip *face_client.py* berisi metode-metode yang digunakan untuk mengakses API dari *Face.com*. *Template* merupakan skrip html yang berfungsi sebagai penampil halaman web untuk pengujian modul ini.



Gambar 4.2 Diagram Blok Fungsi Pengenalan Wajah di *Google App Engine*

Proses pertama dari pengenalan maupun pelatihan wajah adalah mengunggah foto atau citra ke dalam sistem di *Google App Engine* ini. Untuk menangani proses ini digunakan layanan *Blobstore* yang ada pada *Google App Engine*. *Blobstore* merupakan layanan untuk menangani penyimpanan berkas yang ukurannya lebih dari 1MB. Hal tersebut disebabkan layanan *Datastore* yang ada tidak bisa menangani berkas dengan ukuran di atas 1MB. Proses ini terdapat pada skrip *main.py*. Untuk menggunakan fungsi *blobstore*, maka terlebih dahulu membuat alamat unggah yang menangani fungsi unggah tersebut. Kode untuk membuat alamat unggah dapat dilihat pada Gambar 4.3. Setelah alamat unggah

telah dibuat, maka alamat tersebut akan dijadikan *handler* untuk proses unggah yang ditangani oleh kelas untuk mengunggah.

```
class AksesAndroidRekognisi(webapp.RequestHandler):
    def get(self):
        android_rekognisi = blobstore.create_upload_url('/recognize')
        rekognisi_host = 'http://%s%s' %(os.environ['HTTP_HOST'],
            android_rekognisi)
```

Gambar 4.3 Kode Untuk Membuat Alamat Unggah

Ada dua kelas untuk mengunggah yaitu UploadRekognisi dan UploadTraining. UploadRekognisi untuk menangani proses unggah dalam pengenalan wajah, sedangkan UploadTraining untuk menangani proses unggah dalam pelatihan wajah. Kode untuk proses unggah beserta rekognisi dapat dilihat pada Gambar 4.4, dan kode untuk proses unggah beserta trainig dapat dilihat pada Gambar 4.5.

```
class UploadRekognisi(blobstore_handlers.BlobstoreUploadHandler):
    def post(self):
        file_rekognisi = self.get_uploads('citra')
        blob_info = file_rekognisi[0]
        id_gambar_rekognisi = blob_info.key()
        alamat_gambar_rekognisi =
        'http://%s/lihat/%s' %(os.environ['HTTP_HOST'],id_gambar_rekognisi
        )
        api = FaceClient('2cb5cd4b0a3b047aecbf71cd874ce843',
            'e8f2f6597b0882981e9db331b87f776b')
        json_rekognisi = api.faces_recognize(uids = 'all', urls =
            alamat_gambar_rekognisi, namespace = 'ujitraining')
        uid =
        json_rekognisi['photos'][0]['tags'][0]['uids'][0]['uid']
        conf =
        json_rekognisi['photos'][0]['tags'][0]['uids'][0]['confidence']
        ambil_tag = api.tags_get(uids=uid, limit=1)
        label = ambil_tag['photos'][0]['tags'][0]['label']
```

Gambar 4.4 Kode Untuk Unggah dan Rekognisi

```
class UploadTraining(blobstore_handlers.BlobstoreUploadHandler):
    def post(self):
        file_training = self.get_uploads('citra')
        blob_info = file_training[0]
        id_gambar_training = blob_info.key()
        alamat_gambar_training =
        'http://%s/lihat/%s' %(os.environ['HTTP_HOST'],id_gambar_training)
        uids=self.request.get("uid")
        uname=self.request.get("username")
```

```

    api = FaceClient('2cb5cd4b0a3b047aecbf71cd874ce843',
'e8f2f6597b0882981e9db331b87f776b')
    json_train = api.faces_detect(alamat_gambar_training)
    tids = [photo['tags'][0]['tid'] for photo in
json_train['photos']]
    api.tags_save(tids = tids, uid = '%s@ujitraining' %uids,
label = uname)
    status_train = api.faces_train('%s@ujitraining' %uids)
    if status_train['status'] == 'success':
        status = 'Foto Berhasil Dilatih!'
    else:
        status = 'Foto Gagal Dilatih!'

```

Gambar 4.5 Kode Untuk Unggah dan *Training*

Hasil dari proses rekognisi atau pelatihan tersebut merupakan sebuah respon dari API Face.com. Respon yang berupa file JSON tersebut merupakan hasil pengolahan yang dilakukan API Face.com terhadap citra yang dikirimkan. API Face.com tersebut membantu dalam proses pelatihan terhadap citra-citra wajah yang dimiliki pengguna pada profil Facebook yang dimiliki. Selain itu informasi Facebook yang dimiliki oleh pemilik citra wajah yang dikenali juga dapat diketahui.

Struktur dari respon JSON terdiri dari beberapa bagian, yaitu keterangan hasil pengenalan, user id dari akun Facebook atau *namespace* lain yang dikenali, struktur wajah berupa koordinat, serta atribut lain seperti perkiraan umur, jenis kelamin, aksesoris yang digunakan, mimik wajah, hingga kondisi mood. User id yang didapatkan dari hasil pengenalan terdiri dari beberapa akun yang memiliki kemungkinan pemilik wajah yang dikenali. Tingkat kemungkinan tersebut terdapat dalam nilai *confidence* pada setiap *uid*. Contoh isi respon JSON hasil pengenalan ditampilkan dalam Gambar 4.6.

```

{
  -photos: [
  -{
    url: http://gaefacehosting.appspot.com/image/6001/
pid: "F@a82cffbec3a279d4276ff652004949e0_4b4b4c6d54c37"
width: 389
height: 259
  -tags: [
  -{
    tid:
    "TEMP_F@a82cffbec3a279d4276ff652004949e0_4b4b4c6d54c37_31.11_41.31
_1_1"

```

```

recognizable: true
threshold: 58
-uids: [
- {
uid: "100000972723531@facebook.com"
confidence: 54
}]]

```

Gambar 4.6 Contoh Respon JSON

4.3. Hasil Implementasi dan Penggunaan Modul

Modul pengenalan wajah yang dibangun di *Google App Engine* ini bila dilihat dari sistem secara keseluruhan hanya dapat diakses melalui aplikasi Android yang dibuat. Namun untuk pengujian secara unit, dibuatlah sebuah antar-muka dalam bentuk web untuk memudahkan pengujian, yang dapat diakses di alamat <http://gaefacerecognition.appspot.com>.



Gambar 4.7 Tampilan Awal Web untuk Pengujian

Pada Gambar 4.7, ditampilkan halaman utama dari web untuk pengujian modul pengenalan wajah. Pada halaman tersebut terdapat tautan Rekognisi untuk mengunggah file citra yang ingin dikenali. Selain itu terdapat tautan *Training* untuk mengunggah file citra yang ingin dilatih. Setelah tautan tersebut dipilih maka web akan menuju ke halaman untuk memilih berkas citra yang ingin diunggah. Hasil dari proses rekognisi dapat dilihat pada Gambar 4.8.



Gambar 4.8 Halaman Hasil Pengenalan Citra

BAB 5

PENGUJIAN DAN ANALISA MODUL PENGENALAN WAJAH

5.1. Rancangan Pengujian Modul Pengenalan Wajah

Hasil pengenalan wajah yang telah dibangun, akan didiskusikan pada bab ini. Pengujian ini bertujuan untuk mendapatkan analisa terhadap tingkat kesuksesan sistem yang dibangun dalam melakukan fungsinya. Dalam pengujian modul pengenalan wajah ini, terdapat tiga variasi pengujian untuk tingkat keakurasian pengenalan wajah. Pengujian pertama yaitu dengan variasi jumlah data pelatihan, kedua dengan variasi intensitas cahaya, ketiga yaitu dengan variasi sudut pandang wajah.

Data pelatihan wajah yang digunakan dalam pengujian ini berjumlah 50 wajah, dengan rincian tiga pria dan dua wanita yang masing-masing memiliki 10 data wajah. Untuk proses rekognisi, digunakan data wajah yang berbeda dengan data pelatihan. Setiap objek sampel wajah, memiliki dua sampel wajah tambahan yang berbeda kondisi dengan data pelatihan untuk digunakan sebagai data rekognisi dengan kondisi beragam. Contoh wajah yang digunakan dalam pengujian ini dapat dilihat pada Tabel 5.1.

Tabel 5.1 Sampel Wajah Untuk Percobaan

Objek Wajah	Objek 1	Objek 2	Objek 3	Objek 4	Objek 5
Gambar Wajah					

5.1.1. Pengujian Tingkat Pengenalan Wajah dengan Variasi Jumlah Data Pelatihan

Dalam pengujian ini, akan dibandingkan tingkat keakurasian pengenalan wajah dengan jumlah data pelatihan. Dalam melakukan proses pengenalan wajah ini, terdapat fase pelatihan terhadap citra atau gambar wajah yang diunggah

pengguna. Jumlah data pelatihan ini ditentukan dari banyaknya foto yang dimiliki oleh objek wajah tersebut. Pengujian ini dilakukan sebanyak sepuluh kali untuk masing-masing jumlah data pelatihan.

Data pelatihan yang digunakan terdiri dari berbagai macam kondisi pengambilan gambar wajah. Dari sepuluh data pelatihan, tiga data dengan kondisi cahaya gelap, tiga data dengan kondisi cahaya sedang, dan empat data dengan kondisi cahaya terang. Penjelasan mengenai kondisi pencahayaan dapat dilihat pada subbab 5.1.2. Pada masing-masing kondisi pencahayaan tersebut, terdapat tiga posisi sudut pandang wajah, yaitu 0 derajat, 10 derajat, dan 20 derajat. Untuk kondisi terang ditambahkan posisi 30 derajat, karena terdapat empat data. Penjelasan lebih spesifik mengenai posisi wajah dapat dilihat pada subbab 5.1.3.

Jumlah data pelatihan diurutkan dari 1 hingga 10. Tahap-tahap proses pengujian dengan variasi jumlah data pelatihan ini dijelaskan dalam urutan berikut ini.

1. Setiap objek wajah dilakukan pelatihan dengan mengunggah satu data pelatihan.
2. Setelah kelima objek wajah telah dilatih, maka selanjutnya dilakukan proses pengujian rekognisi. Setiap objek wajah terdapat dua data wajah untuk pengujian rekognisi, sehingga total data pengujian adalah 10 data untuk setiap variasi jumlah data pelatihan.
3. Mencatat keberhasilan setiap proses rekognisi dan persentase kesalahan yang ditampilkan pada setiap hasil rekognisi.
4. Ulangi langkah 1 sampai 3, hingga data pelatihan sudah berjumlah 10 data.

5.1.2. Pengujian Tingkat Pengenalan Wajah dengan Variasi Tingkat Intensitas Cahaya

Dalam pengujian ini, akan dibandingkan tingkat keakurasian pengenalan wajah dengan tingkat intensitas cahaya sekitar pada saat melakukan proses pengenalan. Variasi tingkat intensitas cahaya ini didapatkan dengan melakukan proses pengenalan wajah ditempat yang berbeda-beda kondisi pencahayaannya. Kondisi pencahayaan yang digunakan adalah gelap, sedang, dan terang.

Untuk kondisi gelap, pengujian dilakukan di sebuah ruangan dengan ukuran lebar 5 meter, panjang 7 meter, dan tinggi 3 meter. Pengujian dilakukan pada pagi hari pukul 09.35, dengan kondisi cuaca di luar cerah, namun di dalam ruangan tersebut tidak menggunakan cahaya lampu tambahan. Hasil pengukuran dengan lux meter, yaitu alat pengukur intensitas cahaya, didapatkan nilai 254,34 lux.

Untuk kondisi sedang, pengujian dilakukan di ruangan yang sama dengan ruangan yang digunakan pada kondisi gelap. Pengujian dilakukan pada pagi hari pukul 10.14, dengan kondisi cuaca di luar cerah, dan lampu di ruangan tersebut dalam kondisi menyala. Lampu yang digunakan adalah lampu TL dengan daya 36 watt, dan berjumlah empat buah. Hasil pengukuran dengan lux meter didapatkan nilai 476,21 lux.

Untuk kondisi terang, pengujian dilakukan di luar ruangan dengan sinar matahari langsung. Pengujian dilakukan pada pagi hari pukul 10.36, dengan kondisi cuaca di luar cerah. Hasil pengukuran dengan lux meter didapatkan nilai 892,89 lux.

Pengujian dengan variasi kondisi pencahayaan ini, dilakukan setelah pengujian dengan variasi data pelatihan. Oleh karena itu, data pelatihan yang digunakan untuk pengujian kali ini berjumlah 10, dengan spesifikasi data pelatihan yang telah dijelaskan pada subbab 5.1.1. Setiap objek sampel wajah dilakukan dua kali pengambilan untuk setiap kondisi pencahayaan, sehingga setiap kondisi pencahayaan, memiliki total pengujian sebanyak 10 kali.

5.1.3. Pengujian Tingkat Pengenalan Wajah dengan Variasi Posisi Sudut Pandang Wajah

Dalam pengujian ini, akan dibandingkan tingkat keakurasian pengenalan wajah dengan posisi sudut pandang wajah pada saat melakukan proses pengenalan. Variasi posisi sudut pandang wajah ini didapatkan dengan melakukan proses pengenalan wajah dengan mengarahkan kamera ke posisi wajah yang berbeda-beda. Posisi sudut pandang wajah ditetapkan dalam ukuran derajat sudut. Posisi 0° merupakan posisi tepat di depan wajah (*frontal face*). Variasi posisi sudut

ditingkatkan dengan penambahan 10° pada setiap variasi, hingga 90° . Tabel 5.2 menampilkan posisi wajah, dari 0° hingga 90° .

Tabel 5.2 Contoh Posisi Sudut Pengambilan Gambar

Posisi Wajah	0	10	20	30	40
Gambar Wajah					
Posisi Wajah	50	60	70	80	90
Gambar Wajah					

5.2. Hasil Pengujian dan Analisis

Dari tiga rancangan yang telah disebutkan pada subbab 5.1, dilakukan pengujian serta analisis. Pengujian yang dilakukan menghasilkan tiga tabel data percobaan. Dari masing-masing tabel tersebut dibuat grafik yang menggambarkan hubungan antara variabel yang diuji. Tabel dan grafik tersebut akan dianalisa untuk mendapatkan penjelasan tentang hubungan variabel yang diuji.

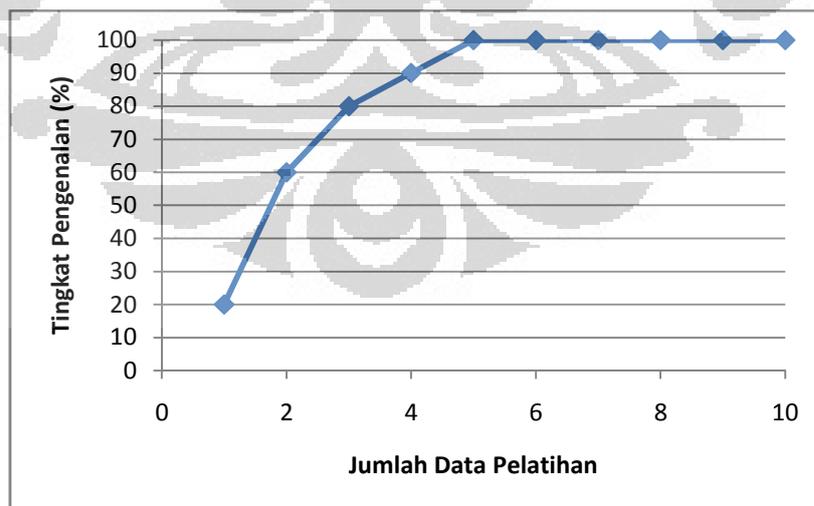
5.2.1. Pengujian Tingkat Pengenalan Wajah dengan Variasi Jumlah Data Pelatihan

Pada pengujian yang dilakukan dengan variasi jumlah data pelatihan, didapatkan hasil yang ditunjukkan pada Tabel 5.2. Pada Tabel 5.2 ditampilkan data perbandingan antara jumlah data pelatihan dengan tingkat pengenalan wajah. Kolom pertama merupakan variasi jumlah data pelatihan dalam pengujian kali ini. Kolom kedua merupakan persentase keberhasilan pengenalan wajah dalam sepuluh kali percobaan. Kolom ketiga merupakan tingkat kesalahan rata-rata dari lima sampel objek.

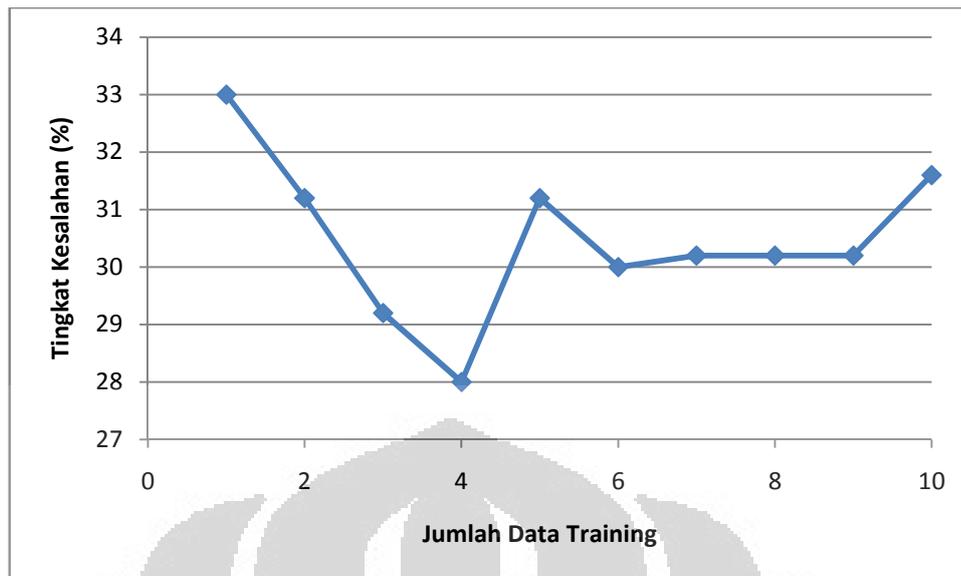
Tabel 5.3 Perbandingan Jumlah Data Pelatihan dengan Tingkat Pengenalan dan Tingkat Kesalahan

Jumlah Data Pelatihan	Rata-rata Tingkat Pengenalan (%)	Rata-rata Tingkat Kesalahan (%)
1	20	33
2	60	31,2
3	80	29,2
4	90	28
5	100	31,2
6	100	30
7	100	30,2
8	100	30,2
9	100	30,2
10	100	31,6
Rata-Rata	85	30,48

Dari Tabel 5.2 di atas dapat dihasilkan dua buah grafik. Grafik pertama dapat dilihat pada Gambar 5.1. Gambar 5.1 menunjukkan pengaruh jumlah data pelatihan terhadap tingkat pengenalan wajah oleh sistem ini. Sumbu X menyatakan variasi jumlah data pelatihan, sedangkan sumbu Y menyatakan tingkat pengenalan wajah. Grafik kedua dapat dilihat pada Gambar 5.2. Gambar 5.2 menunjukkan pengaruh jumlah data pelatihan terhadap tingkat kesalahan yang dihasilkan.



Gambar 5.1 Pengaruh Jumlah Data Pelatihan Terhadap Tingkat Pengenalan



Gambar 5.2 Pengaruh Jumlah Data Pelatihan dengan Tingkat Kesalahan

Berdasarkan tabel dan grafik di atas dapat dilakukan beberapa analisa. Dengan jumlah data pelatihan hanya satu buah, tingkat pengenalan masih rendah, yaitu hanya 20%. Seiring dengan bertambahnya jumlah data pelatihan, maka terjadi peningkatan tingkat pengenalan. Ketika data pelatihan berjumlah lima buah, maka tingkat pengenalan sudah mencapai rata-rata 100% dari 10 percobaan.

Berbeda dengan tingkat pengenalan, tingkat kesalahan yang didapatkan dengan variasi jumlah data pelatihan ini mempunyai nilai yang cukup beragam. Dengan data pelatihan hanya satu buah, tingkat kesalahan yang didapatkan berada pada nilai 33%. Nilai kesalahan tersebut terus menurun hingga data pelatihan berjumlah 4, dengan nilai kesalahan 28%. Namun ketika jumlah data pelatihan terus ditambah, tingkat kesalahan kembali meningkat hingga pada nilai 31,2%. Setelah itu nilai kesalahan ini akan kembali stabil pada nilai 30,2%.

Perbedaan tingkat pengenalan serta tingkat kesalahan yang terjadi, menunjukkan jumlah data pelatihan mempengaruhi kedua nilai tersebut. Ketika data pelatihan sangat sedikit, maka pembelajaran sistem terhadap wajah tersebut akan kurang. Semakin banyak data pelatihan, maka pembelajaran sistem akan semakin banyak, sehingga dapat mengenali berbagai macam kondisi pada saat proses pengenalan.

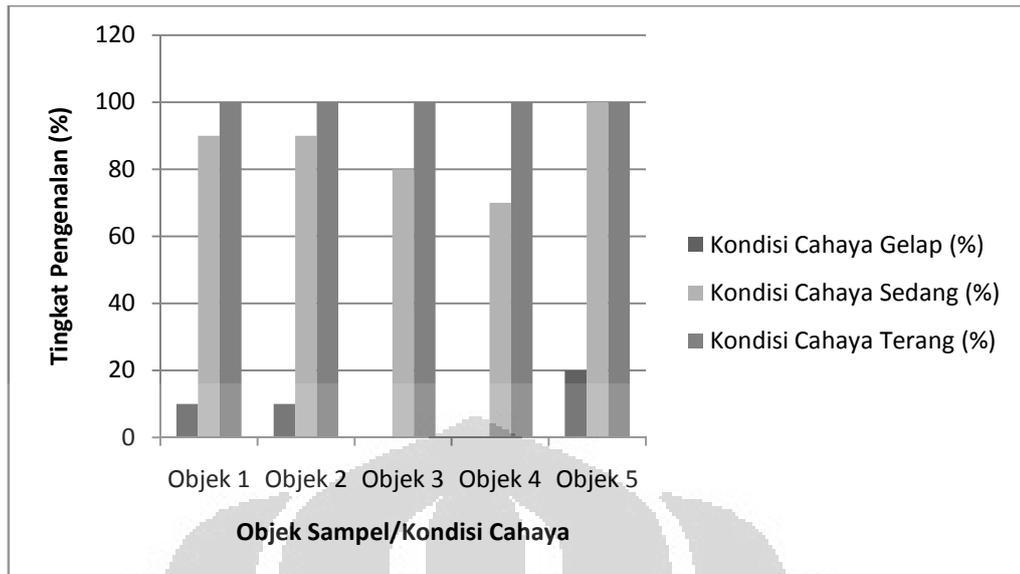
5.2.2. Pengujian Tingkat Pengenalan Wajah dengan Variasi Tingkat Intensitas Cahaya

Pada pengujian yang dilakukan terhadap modul pengenalan wajah, didapatkan hasil yang ditunjukkan pada Tabel 5.3. Pada Tabel 5.3 ditampilkan data perbandingan antara tingkat intensitas cahaya dengan tingkat pengenalan wajah. Kolom pertama merupakan sampel objek wajah untuk uji pengenalan wajah ini. Kolom kedua merupakan tingkat rekognisi (dalam persen) dengan kondisi gelap atau dalam ruangan tanpa lampu pada siang hari. Kolom ketiga merupakan tingkat rekognisi (dalam persen) dengan kondisi sedang atau dalam ruangan dengan lampu. Kolom ketiga merupakan tingkat rekognisi (dalam persen) dengan kondisi terang atau berada diluar ruangan dengan cuaca cerah.

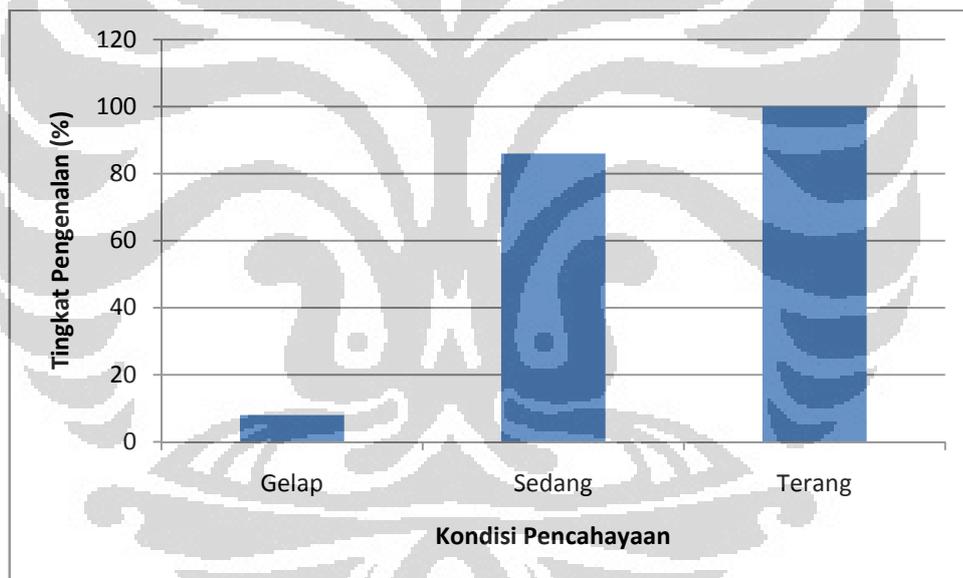
Tabel 5.4 Perbandingan Kondisi Cahaya dengan Tingkat Pengenalan

Sample Objek Wajah	Kondisi Cahaya		
	Gelap (%)	Sedang (%)	Terang (%)
Objek 1	10	90	100
Objek 2	10	90	100
Objek 3	0	80	100
Objek 4	0	70	100
Objek 5	20	100	100
Rata-rata	8	86	100

Dari Tabel 5.3 tersebut dihasilkan grafik tingkat kesuksesan pengenalan wajah pada sampel objek wajah dengan kondisi pencahayaan yang berbeda-beda. Gambar 5.3 menunjukkan grafik hasil pengolahan data pada Tabel 5.3. Sumbu X dan sumbu Y masing-masing menyatakan persentase kesuksesan dan nama objek wajah yang diujikan. Gambar 5.4 menunjukkan grafik rata-rata persentase kesuksesan pengenalan wajah dengan variasi cahaya.



Gambar 5.3 Pengaruh Pencahayaan terhadap Kesuksesan Pengenalan Wajah



Gambar 5.4 Rata-rata Pengaruh Kondisi Pencahayaan terhadap Tingkat Pengenalan Wajah

Berdasarkan tabel dan grafik dari pengujian dengan variasi pencahayaan di atas dapat dilakukan beberapa analisis. Pada kondisi gelap, tingkat kesuksesan pengenalan wajah berada pada rata-rata 8%. Pada kondisi sedang, tingkat kesuksesan pengenalan wajah berada pada rata-rata 86%. Untuk kondisi terang, tingkat kesuksesan wajah berada pada rata-rata 100%.

Perbedaan tingkat kesuksesan pengenalan wajah tersebut dipengaruhi oleh kamera yang memiliki kualitas hasil tangkap yang berbeda pada masing-masing kondisi cahaya. Ketika kondisi cahaya gelap, maka *noise* yang ditimbulkan semakin banyak. Bila kondisi cahaya semakin terang, maka *noise* yang dihasilkan akan semakin sedikit, sehingga kualitas gambar akan semakin jelas. Dengan kualitas gambar yang semakin jelas, maka proses rekognisi akan semakin baik.

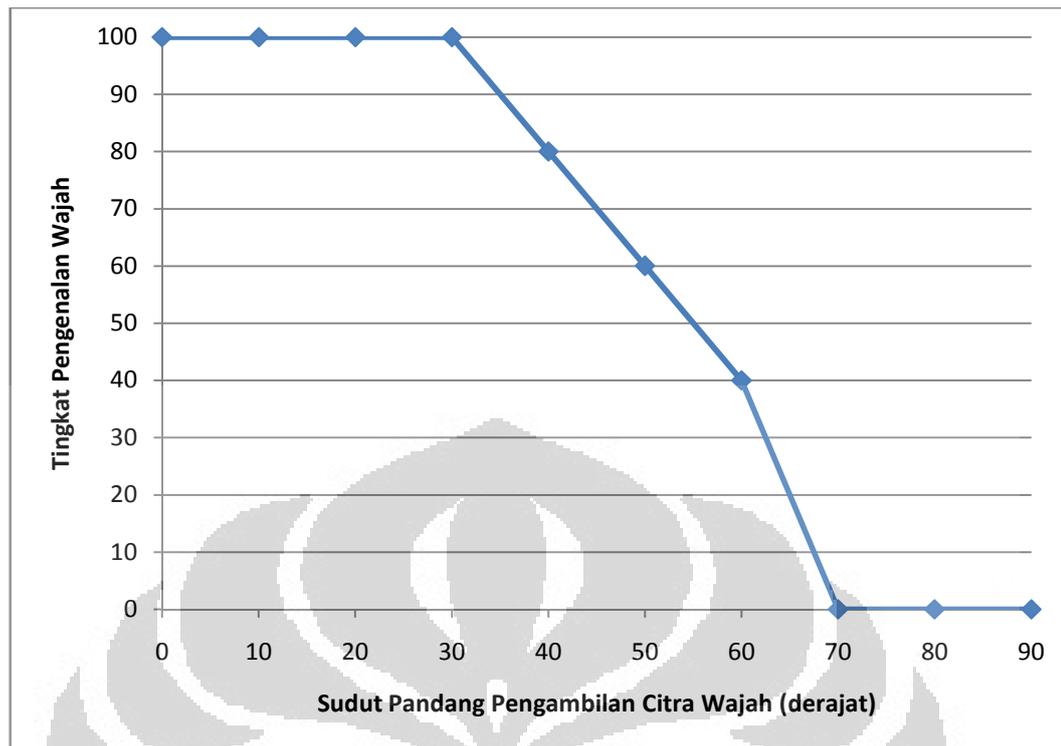
5.2.3. Pengujian Tingkat Pengenalan Wajah dengan Variasi Posisi Sudut Pandang Wajah

Pada pengujian kali ini, terdapat empat variasi posisi sudut pandang wajah. Sudut acuan pertama yaitu dengan posisi *frontal face*, yaitu tepat dihadapan kamera. Tiga variasi berikutnya dilakukan dengan posisi yang semakin menyamping, hingga tepat berada di samping wajah. Tabel 5.4 menunjukkan contoh posisi pengambilan wajah.

Dari hasil pengujian dengan variasi sudut pengambilan wajah ini didapatkan data pengujian yang dapat dilihat pada Tabel 5.5. Kolom pertama merupakan variasi posisi wajah seperti pada Tabel 5.4. Kolom kedua merupakan tingkat pengenalan yang merupakan rata-rata dari 10 kali percobaan. Dari Tabel 5.5 maka dapat dibuat sebuah grafik hubungan antara variasi sudut pandang wajah dengan tingkat pengenalan wajah yang dapat dilihat pada Gambar 5.5.

Tabel 5.4 Data Percobaan Variasi Sudut Pandang Wajah

Sudut Pengambilan	Tingkat Pengenalan (%)
0	100
10	100
20	100
30	100
40	80
50	60
60	40
70	0
80	0
90	0



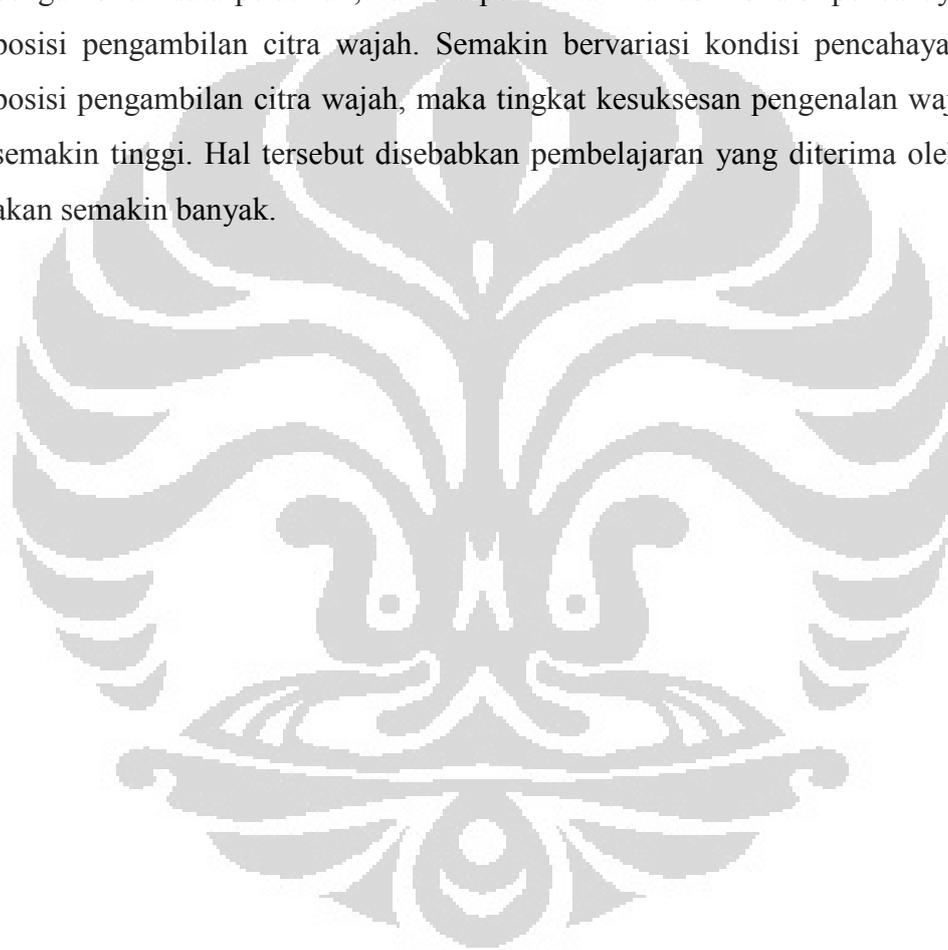
Gambar 5.5 Pengaruh Sudut Pandang Pengambilan Citra Wajah dengan Tingkat Pengenalan Wajah

Dari data dan grafik di atas, dapat dilakukan beberapa analisis. Dengan posisi pengambilan tepat di depan wajah, didapatkan rata-rata pengenalan wajah 100%. Kemudian posisi pengambilan dilakukan semakin menyamping hingga pada posisi kurang lebih 30 derajat yang mendapatkan rata-rata pengenalan wajah masih 100%. Selanjutnya pada posisi kurang lebih 60 derajat, rata-rata pengenalan wajah menurun hingga 40%. Pada posisi 70 hingga 90 derajat atau dapat dikatakan dari posisi samping wajah, maka proses pengenalan yang dilakukan tidak berhasil atau 0%. Maka semakin jauh posisi pengambilan wajah dari acuan posisi tepat di depan wajah, maka tingkat pengenalan semakin menurun. Batas posisi pengambilan wajah dapat disimpulkan pada posisi 60 derajat.

Dari ketiga percobaan di atas, dapat dilakukan beberapa analisis. Ketiga variabel yang digunakan dalam percobaan memiliki pengaruh terhadap kesuksesan sistem melakukan pengenalan wajah. Semakin banyak data pelatihan, maka tingkat kesuksesan pengenalan akan semakin tinggi. Kondisi cahaya pengambilan citra juga mempengaruhi tingkat kesuksesan. Semakin baik kondisi pencahayaan, maka tingkat pengenalan akan semakin tinggi. Sudut pandang

pengambilan citra wajah juga mempengaruhi kemampuan sistem mengenali wajah. Posisi tepat di hadapan wajah merupakan posisi yang paling baik untuk mengenali wajah. Bila posisi semakin menyamping, maka kemampuan pengenalan wajah akan semakin menurun.

Dari kondisi tersebut, maka untuk dapat mendapatkan sebuah pengenalan wajah yang baik, harus memperhatikan tiga faktor. Faktor tersebut adalah jumlah data pelatihan, kondisi pencahayaan, dan posisi pengambilan citra wajah. Dalam pengambilan data pelatihan, harus diperhatikan variasi kondisi pencahayaan dan posisi pengambilan citra wajah. Semakin bervariasi kondisi pencahayaan serta posisi pengambilan citra wajah, maka tingkat kesuksesan pengenalan wajah akan semakin tinggi. Hal tersebut disebabkan pembelajaran yang diterima oleh sistem akan semakin banyak.



BAB 6

PENUTUP

6.1. Kesimpulan

Berikut adalah kesimpulan dari penulisan skripsi ini.

1. Modul pengenalan wajah yang dibangun pada *Google App Engine* berbasis Python dan API Face.com ini dapat melakukan pengenalan wajah dengan tingkat kemampuan pengenalan rata-rata 85% dan total kesalahan 30,48%.
2. Proses pengenalan wajah dengan kondisi pencahayaan gelap hanya menghasilkan 8% kesuksesan pengenalan wajah, dengan kondisi pencahayaan sedang menghasilkan 86% kesuksesan, dan dengan kondisi terang dapat menghasilkan 100% kesuksesan.
3. Proses pengenalan wajah ini dapat dilakukan dengan sudut pandang maksimal pada kisaran 60 derajat dari posisi wajah tampak depan.
4. Penggunaan data pelatihan yang semakin banyak dan beragam akan meningkatkan kemampuan pengenalan wajah dengan beragam kondisi.

6.2. Rencana Pengembangan

Dalam pengembangan sistem ini, diharapkan dapat menggabungkan metode pelatihan dengan koleksi foto dari berbagai jejaring sosial seperti Facebook, Twitter, Google+, dan lain-lain, sehingga kemampuan mengenali wajah akan semakin meningkat. Selain itu diperlukan adanya manajemen privasi dari citra yang diunggah untuk proses rekognisi.

DAFTAR REFERENSI

- [1] Hiltz, R., & Turoff, M. (1998). *The Network Nation 2*. USA: Addison-Wesley.
- [2] Delac, K. (2007). *Face Recognition*. USA: I-TECH Education and Publishing.
- [3] <http://code.google.com/appengine/docs/python/runtime.html> Diakses pada 2 November 2011.
- [4] Delac, K. (2008). *Recent Advances In Face Recognition*. USA: I-TECH Education and Publishing.
- [5] R. Buyya, J. B. (2011). *Cloud Computing, Principles and Paradigms*. New Jersey: John Wiley & Sons Inc.
- [6] <http://python.org/about/> Diakses pada 2 November 2011.
- [7] <http://code.google.com/appengine/docs/whatisgoogleappengine.html> Diakses pada 2 November 2011.
- [8] <http://www.shervinemami.co.cc/faceRecognition.html> Diakses pada 2 November 2011.
- [9] De Jonge, A. (2011). *Essential App Engine: Build High-Performance Java Apps with Google App Engine*. USA: Addison-Wesley.
- [10] <http://www.face-rec.org/> Diakses pada 2 November 2011.
- [11] <http://developers.face.com/docs/> Diakses pada 4 Mei 2012.
- [12] Fowler, M. (2003). *UML Distilled: A Brief Guide to the Standard Object Modelling Language*. USA: Addison Wesley.
- [13] Indrawan, P. (2012). *Implementasi Sistem Pengenalan Wajah Sebagai Penghubung Jejaring Sosial : Aplikasi Mobile Untuk Deteksi Wajah Dengan Android Face Detector API dan Komunikasi REST ke Komputasi Awan*. Depok: Universitas Indonesia.
- [14] Budiayatno, S. (2012). *Implementasi Sistem Pengenalan Wajah Sebagai Penghubung Jejaring Sosial: Penerapan Augmented Reality Sebagai*

Penampil Informasi Hasil Pengenalan Wajah Pada Perangkat Android.
Depok: Universitas Indonesia.

- [15] Zuong, M., Zeng, G., & Tu, X. (2010). *Research and Improvement of Face Detection Algorithm Based on the OpenCV*. International Conference on Information Science and Engineering, Hangzhou, China, 2010.
- [16] Ferzli, R., & Khalife, I. (2011). *Mobile Cloud Computing Educational Tool For Image/Video Processing*. Digital Signal Processing Workshop, Arizona, USA, 2011.

