

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Keamanan informasi merupakan salah satu masalah yang dihadapi saat ini. Setiap organisasi maupun perseorangan ingin meningkatkan sistem keamanan mereka. Kejahatan-kejahatan yang sering terjadi terkait masalah keamanan tersebut adalah seperti penyalahgunaan kartu kredit, peretasan oleh para peretas dan lain sebagainya. Sistem yang sering diretas atau mengalami masalah kriminal tersebut dikarenakan sistem tidak memberikan akses berdasarkan siapa *user* tersebut, tetapi apa yang *user* miliki pada sistem seperti nomor identitas, *email*, *username*, *password*, nomor PIN. Untuk mengatasi hal tersebut ada salah satu teknologi yang mampu memverifikasi dari identitas asli yang dimiliki oleh pengguna yaitu wajah. Teknologi tersebut lebih dikenal dengan pengenalan wajah atau *face recognition* (Cho, et al., 2014) (Jayakumar & Sandeep, 2015).

*Face recognition* merupakan sebuah teknik yang menggunakan komputer untuk menganalisis *image* wajah yang akan menghasilkan informasi pengenalan yang berguna dari *image* wajah tersebut. *Face recognition* sendiri merupakan salah satu topik pembahasan dari bidang kajian *computer vision*. *Computer vision* merupakan studi yang bertujuan untuk menjadikan komputer memiliki kemampuan mengenali sesuatu mendekati kemampuan manusia. Selain itu, *face recognition* mampu melakukan pengenalan dengan cara membandingkan *image* wajah yang dimasukkan ke dalam sistem dengan *image* wajah yang ada pada basis data yang ada.

Saat ini, algoritma-algoritma yang ada untuk melakukan *face recognition* dikelompokkan secara umum ke dalam tiga kategori berdasarkan pengambilan *feature*. Pertama adalah *holistic methods*: algoritma ini merepresentasikan seluruh wilayah wajah pada *image* sebagai vektor berdimensi tinggi yang digunakan sebagai acuan atau *input* untuk pengelompokkan. Berikutnya *local*

*methods*: mengambil sebagian kecil seperti hidung, mata, dsb. dari *image* wajah yang kemudian digunakan sebagai penentu pengelompokan *image*.  
*Hybrid methods*: menggabungkan kedua metode tersebut (Cho, et al., 2014).

Algoritma yang digunakan untuk melakukan *face recognition* pada penelitian ini adalah dengan menggunakan *Elastic Bunch Graph Matching* (EBGM) (Grgic & Delac, 2005). EBGM sendiri memiliki dua fase utama yaitu: pemodelan, dan pengenalan (Jayakumar & Sandeep, 2015). Fase pertama dari EBGM adalah membuat sekumpulan nilai grafis dari set model menggunakan metode *Gabor Wavelet Transform*. Setelah itu, algoritma ini akan melakukan *filter* terhadap semua *image* yang ada dengan nilai Gabor tersebut dan menghasilkan sebuah vektor untuk tiap *image* yang disebut *Jets*. Kemudian fase terakhir adalah melakukan *face recognition* dengan cara membandingkan atau mengukur kemiripan antara *feature* yang dimasukkan dengan seluruh *image* yang ada pada basis data untuk mencari *image* mana yang paling mirip (Jayakumar & Sandeep, 2015).

Pada umumnya proses *face recognition* saat ini masih menggunakan CPU yang ada pada komputer. Bagaimanapun proses *face recognition* yang dilakukan akan memakan waktu yang lama, hal tersebut disebabkan karena proses komputasi hanya ditangani oleh sebuah *thread* atau *core*. Hal tersebut menyebabkan proses komputasi yang terjadi untuk tiap pikselnya berjalan secara sekuensial, padahal piksel yang dimiliki sebuah *image* bisa berjumlah sangat banyak. Kondisi tersebut dapat diperparah dengan teknik atau algoritma yang digunakan. Jika algoritma yang digunakan sangat rumit, maka proses komputasi juga akan semakin lama (Patel & Vajani, 2015). Masalah tersebut dapat diatasi dengan menggunakan *General Purpose GPU* (GPGPU) dengan cara memparalelisasikan proses komputasi yang ada ke dalam *core* pada GPU yang jumlahnya sangat banyak, sehingga proses komputasi yang terjadi dapat dipercepat waktu komputasinya.

*Face recognition* sendiri juga memiliki masalah lain yaitu proses komputasi yang dilakukan terhadap *image* wajah terbilang kompleks dan sangat memakan waktu. Masalah ini dihadapi oleh algoritma *face recognition*

yaitu EBGM (Chen, et al., 2013). Salah satu solusinya adalah dengan menggunakan pemrograman paralel untuk mendapatkan pendeteksian wajah yang lebih cepat. Untuk mendeteksi wajah dari *image*, sub bagian dari *image* haruslah diambil terlebih dahulu dan dilakukan pengecekan apakah sub bagian tersebut wajah atau bukan dengan proses komputasi. Proses kalkulasi di tiap sub bagian *image* tidak bergantung pada sub bagian lain dan selalu mengeksekusi instruksi perhitungan yang sama. Oleh karena itu, dimanfaatkanlah GPU sebagai solusi paralelisasi dikarenakan GPU mampu melaksanakan satu instruksi secara bersamaan di tiap *core*-nya sehingga dapat diperoleh proses komputasi yang lebih cepat.

## 1.2. Rumusan Masalah

Rumusan masalah-masalah yang akan dihadapi atau hendak diselesaikan berdasarkan latar belakang adalah sebagai berikut.

- 1.2.1. Bagaimana membangun *face recognition* menggunakan metode EBGM?
- 1.2.2. Bagaimana tingkat akurasi untuk *face recognition* menggunakan algoritma EBGM menghasilkan hasil yang baik dan dapat diterima?
- 1.2.3. Bagaimana mengembangkan *face recognition* tersebut menggunakan GPU CUDA agar proses komputasi lebih optimal dan cepat?

## 1.3. Tujuan

Penelitian ini memiliki tujuan sebagai berikut.

- 1.3.1. Mampu membangun *face recognition* dengan menerapkan algoritma EBGM.
- 1.3.2. Menghitung tingkat akurasi *face recognition* menggunakan algoritma EBGM terhadap *image* baru menghasilkan hasil akurasi yang baik dan dapat diterima.
- 1.3.3. Membandingkan kinerja atau proses komputasi yang dibutuhkan untuk *face recognition* menggunakan algoritma EBGM yang sebelumnya menggunakan CPU dengan sesudah menggunakan GPU.

## 1.4. Batasan Masalah

Batasan-batasan masalah dalam penelitian ini adalah sebagai berikut.

1.4.1. *Image* yang diterima berupa sebuah image atau foto digital yang di masukkan secara manual, bukan melalui video atau *live video* dari *webcam*.

1.4.2. *Training Set* atau image yang dijadikan pelatihan adalah *image* wajah mahasiswa Atma Jaya.

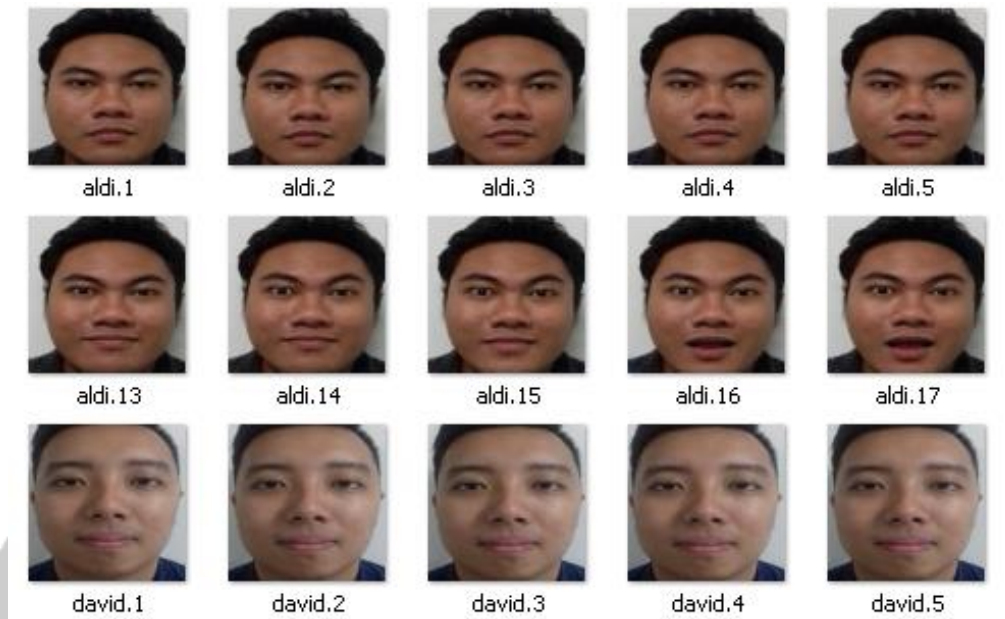
1.4.3. *Image* hanya dibatasi resolusi maksimal *image* adalah 128 x 128 dan 256 x 256.

## 1.5. Metodologi

### 1.5.1. Alat dan bahan

Pada penelitian ini akan digunakan *software development tools* Visual Studio 2012 Enterprise Edition. Penggunaan Visual Studio 2012 ini bertujuan agar pengkodean dalam bahasa C/C++ dapat berjalan dengan baik, serta untuk mendapatkan kemudahan dalam pengaturan GPU CUDA. Komputer yang digunakan penulis selama penelitian memiliki spesifikasi menggunakan sistem operasi Windows 7 Ultimate, dengan CPU Intel Core i7 dengan kecepatan tiap *core*-nya adalah 3.5 Ghz. Komputer ini juga memiliki RAM sebesar 16 GB serta memiliki GPU Nvidia Geforce GTX 660 Ti.

Selain itu, pada penelitian ini akan menggunakan kumpulan *image* dari wajah beberapa mahasiswa yang akan dijadikan ke dalam basis data sebagai *training set* untuk individu-individu yang diketahui dalam penelitian. Jumlah *dataset* yang digunakan adalah untuk satu orang berjumlah kurang lebih 400 *image* wajah dari total 13 orang dengan pembagian 75% menjadi *data training* dan 25 % menjadi *data test*. Gambar 1.1 merupakan beberapa contoh *image* yang digunakan pada penelitian ini.



Gambar 1.1. Contoh *Dataset Training* yang Digunakan

## 1.5.2. Langkah-langkah penelitian

### 1.5.2.1. Studi pustaka

Peneliti melakukan studi pustaka berdasarkan jurnal, *proceeding*, dan literatur yang berkaitan terhadap topik masalah yang hendak diselesaikan. Literatur-literatur tersebut merupakan literatur yang berkaitan dengan masalah *face recognition* dengan menggunakan algoritma yang akurat dan baik dengan masalah waktu eksekusi yang relatif memakan waktu.

### 1.5.2.2. Analisis algoritma

*Face recognition* dibutuhkan untuk mampu mengenali *image* yang dimasukkan ke dalam sistem sebagai salah satu solusi keamanan terutama dalam hal autentikasi, dan salah satu contohnya adalah dapat mencegah penyalahgunaan kartu kredit oleh orang lain untuk ke depannya jika menerapkan *face recognition*. Selain itu, algoritma EBGM sendiri digunakan agar *face recognition* terhadap suatu *image* dapat terwujud. Algoritma EBGM mampu memproses

*image* yang masuk ke dalam sistem yang kemudian dapat dikelompokkan apakah *image* tersebut mewakili salah satu individu yang *imagenya* ada pada basis data atau *training set*. Namun dari hasil penggunaan algoritma EBGM waktu eksekusinya kurang memuaskan dikarenakan algoritma ini akan mengekstrak terlebih dahulu *feature* dari satu buah *image* dengan cara mem-*filter* *image* tersebut dengan *gabor wavelet* dan dilakukan *rescaling* sebanyak 40 kali. Kemudian hal yang sama juga dilakukan untuk semua *image*, maka dari itu dilakukan paralelisasi dengan menggunakan GPU CUDA agar proses komputasi yang terjadi dapat dijalankan secara bersamaan sehingga waktu yang dibutuhkan untuk proses *face recognition* dapat berjalan jauh lebih cepat.

### 1.5.2.3. Perancangan aplikasi

Langkah-langkah untuk proses *face recognition* sendiri secara umum dibagi menjadi tiga tahap, yaitu: 1) *Generate* nilai *gabor wavelet* sesuai dengan dimensi *image* dan melakukan *filter* dengan masing-masing gambar sehingga menghasilkan *jet*, 2) pengambilan *feature point* dengan dimensi 16x16 lalu disimpan kedalam *feature vector*, 3) pengenalan wajah dari *image* berdasarkan nilai-nilai yang sudah didapatkan dari *image* yang dimasukkan dengan *image* yang ada pada *data training*. Untuk bisa melakukan itu maka digunakan algoritma atau metode EBGM.

Kemudian untuk mempercepat proses komputasi yang terjadi akan digunakan GPU CUDA, adapun langkah-langkah atau algoritma yang harus dilakukan yaitu sebagai berikut.

- a. *Image* wajah yang hendak dihitung pada algoritma EBGM maupun dalam pencarian bobot diubah ke dalam bentuk *array* satu dimensi dan bertipe *double* dan sebuah *array* yang bertipe sama sebagai penampung hasil perhitungan yang dideklarasikan dalam CPU.

- b. Memindahkan data *image* yang dibuat sebelumnya dan *array* penampung hasil perhitungan dalam CPU ke dalam GPU menggunakan fungsi bawaan dari CUDA.
- c. Membuat fungsi penghitungan atau proses komputasi untuk algoritma EBGM dan bobot ke dalam format yang diterima CUDA dan bertipe *global memory*.
- d. Memasukkan data yang sudah ada dan *array* penampung hasil perhitungan pada *memory* GPU ke dalam fungsi yang juga sudah dibuat dalam *memory* GPU agar dapat dieksekusi secara paralel.
- e. Jika proses perhitungan sudah selesai, nilai baru yang dicari akan tersimpan dalam *array* satu dimensi penampung hasil. Maka dari itu, langkah berikutnya adalah memindahkannya kembali ke dalam *memory* CPU agar dapat dilanjutkan ke proses berikutnya.

### **1.5.3. Pengimplementasian**

Pada tahap ini peneliti akan membuat aplikasi berdasarkan hasil analisis dan studi pustaka yang telah dilakukan sebelumnya. Setelah berhasil maka tahap berikutnya adalah dengan menguji aplikasi yang telah dibuat berdasarkan rumusan masalah dan tujuan penelitian yang telah ditentukan sebelumnya.

### **1.5.4. Pengujian**

Pengujian dari hasil penelitian ini adalah dengan menghitung rasio terhadap berapa *image* wajah yang dimasukkan ke dalam sistem yang berhasil dan benar untuk *image* yang memang wajah dari mahasiswa yang diketahui dan *image* yang bukan wajah. Selain itu akan dilakukan perbandingan waktu eksekusi dari program untuk proses *face recognition* dengan algoritma EBGM tanpa menggunakan paralelisasi dengan proses *face recognition* dengan

menggunakan GPU. Akan dibandingkan proses *face recognition* mana yang memiliki waktu eksekusi yang lebih baik.

## **1.6. Sistematika Penulisan Tugas Akhir**

Sistematika penulisan laporan akhir ini dapat dijabarkan sebagai berikut.

### **BAB 1: PENDAHULUAN**

Bab ini berisi tentang pembahasan mengenai *face recognition*, algoritma yang akan digunakan, serta bagaimana cara menyelesaikan masalah yang hendak diselesaikan melalui penelitian ini.

### **BAB 2: TINJAUAN PUSTAKA**

Bab ini berisi tentang penjelasan singkat dan ringkasan mengenai penelitian terdahulu yang berhubungan atau memiliki kesamaan dengan permasalahan yang akan dibahas oleh penulis di dalam tugas akhir ini.

### **BAB 3: LANDASAN TEORI**

Bab ini berisi tentang teori-teori yang digunakan penulis dalam tugas akhir seperti halnya algoritma dan lain-lain sebagainya sebagai arahan dalam pengerjaan tugas akhir ini.

### **BAB 4: ANALISIS DAN PERANCANGAN**

Bab ini berisi perancangan algoritma *Elastic Bunch Graph Matching* (EBGM) dari pembuatan *Gabor Wavelet*, pengambilan *feature*, sampai penghitungan tingkat kesamaan (*similarity*) antara *feature image test* dengan *image train* baik menggunakan CPU maupun GPU.

### **BAB 5: IMPLEMENTASI DAN PENGUJIAN**

Bab ini berisi pembahasan dan perancangan dari bagaimana memuat *image* yang sudah disiapkan, pengambilan *feature* dari *image* tersebut.



sampai bagaimana melakukan *recognition* terhadap *image* yang hendak di uji, serta bagaimana mempercepatnya menggunakan paralelisasi dengan GPU. Setelah itu akan dilakukan pengujian terkait seberapa tinggi akurasi yang dihasilkan serta melakukan pengujian / perbandingan antara waktu eksekusi program menggunakan paralelisasi GPU dengan tanpa paralelisasi GPU atau hanya menggunakan CPU saja.

## **BAB 6: KESIMPULAN DAN SARAN**

Bab ini merupakan penutup dan berisi kesimpulan dari hasil penelitian yang penulis lakukan serta saran terkait kelanjutan atau harapan akan penelitian ini untuk pengembangan lebih lanjut.