





#### Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

yang penting digunakan untuk memanipulasi data gambar sesuai yang diinginkan dalam suatu kebutuhan informasi (Kurniawan, 2010).

Salah satu teknik pengolahan citra adalah deteksi tepi. Deteksi tepi merupakan salah satu dari banyak operasi yang ada pada pengolahan citra dan dalam pengolahan dalam visi komputer (*computer vision*) karena memiliki peran yang cukup signifikan dalam berbagai aplikasi pengolahan citra pada visi mesin (*machine vision*). Deteksi tepi sangat penting dalam pengolahan citra karena pendeteksian tepi merupakan langkah awal untuk melingkupi informasi di dalam citra dimana tepi merincikan batas-batas objek.

Seringkali gambar yang digunakan untuk proses pengolahan berukuran sangat besar akan tetapi pengolahan membutuhkan waktu yang cepat. Solusi yang paling tepat adalah dengan mengadaptasi hardware yang mampu mengakomodasi proses. Solusi yang lebih baik adalah memanfaatkan arsitektur *multicore* dengan pengoperasian paralel (N.E.A.Khalid, 2011).

Pada penelitian (Vijayarani & Vinupriya, 2013) yang berjudul “*An Efficient Edge Detection Algorithm for Facial Image in Image Mining*” yakni membandingkan kecepatan waktu eksekusi operator Canny dengan operator Sobel dengan menggunakan gambar wajah. Dari hasil penelitiannya operator Canny memiliki waktu eksekusi sebesar 34,7 detik sedangkan operator Sobel memiliki waktu eksekusi yang lebih lambat yaitu sebesar 34,9 detik.

Namun, dari penelitian tersebut memiliki sebuah kekurangan yaitu proses pendeteksian dapat menghabiskan waktu yang lama jika citra yang digunakan akan dideteksi memiliki ukuran pixel dengan dimensi yang besar. Ditambah lagi, algoritma pada deteksi tepi Sobel masih menggunakan pemrograman secara serial. Untuk dapat memproses deteksi pada gambar yang ukuran pixel dengan dimensi yang besar membutuhkan perangkat keras yang menunjang (Munir, 2004).

Masing-masing citra digital memiliki ukuran file yang berbeda-beda tergantung dari panjang dan lebar citra. Semakin besar dimensi dan kedalaman warna dari suatu citra digital maka akan semakin mahal proses komputasi yang dilakukan.



#### Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Salah satu indikator kerja proses komputasi citra digital dapat dilihat berdasarkan lamanya proses pengolahan citra (Mulya & Abdiansah, 2013).

Karena pemrograman serial memiliki beberapa kelemahan diantaranya yaitu hanya dapat mengeksekusi satu proses atau instruksi dalam satu waktu. Hal ini menyebabkan waktu yang lama dalam mengeksekusi beberapa instruksi. Pemrograman paralel merupakan salah satu cara yang bisa digunakan untuk dapat mengatasi kelemahan pada pemrograman serial.

Pada penelitian sebelumnya berjudul “*Parallel approach of Sobel Edge Detector on Multicore Platform*” yang dilakukan oleh N.E.A.Khalid, S.A.Ahmad, N.M.Noor, A.F.A.Fadzil dan M.N.Taib melakukan penelitian deteksi tepi pada gambar secara paralel menggunakan ukuran gambar 1000 x 1000, 2000 x 2000, 3000 x 3000 dan menggunakan komputer Quad core dan Intel Core 2 Duo. Hasil dari penelitian tersebut adalah menggunakan algoritma Sobel secara paralel meningkatkan performa algoritma sequensial. Pemrosesan paralel lebih baik dari pemrosesan sequensial dalam hal kecepatan, performa dan efisiensi memanfaatkan prosesor individual. Meningkatkan jumlah ukuran data yang akan diproses, multicore memberikan alternatif untuk proses yang cepat.

Dalam melakukan proses paralel, diperlukan mekanisme dalam melakukan komputasi paralel untuk dapat menjalankan aplikasi-aplikasi secara paralel. Mekanisme proses paralel yang digunakan adalah MPI (*Message Passing Interface*). Ada banyak perangkat lunak yang dikenal untuk implementasi dari MPI, antara lain : MPICH, FT-MPI, LA-MPI, LAM/MPI dan OpenMPI (Moleong, C, & dkk, 2013). Dalam penelitian ini, penulis menggunakan pustaka OpenMPI sebagai perangkat lunak untuk dapat mendistribusikan proses ke simpul-simpul. Berdasarkan penelitian M.Dedy Syahputra dan Irfan Akbar tahun 2013 dalam judul “*Uji kerja OCTAVE dengan MPITB menggunakan pustaka LAM/MPI dan OpenMPI*”, didapatkan hasil perbandingan antara pustaka LAM/MPI dan OpenMPI bahwa tingkat efisiensi dari kecepatan pustaka OpenMPI lebih baik dibandingkan dengan kecepatan LAM/MPI. Hal tersebut didapat dari beberapa hasil uji coba, diantaranya NAS Parallel Benchmark (NPB) EP, Pi, serta Mandelbrot.



### Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Pada penelitian ini, penulis akan membangun *cluster* untuk melakukan proses pendeteksian tepi pada gambar menggunakan operator Sobel dengan bahasa pemrograman c++ menggunakan pustaka OpenMPI.

## 1.2 Rumusan Masalah

Berdasarkan permasalahan yang telah dipaparkan pada latar belakang diatas, maka didapatkan rumusan masalah pada penelitian ini adalah sebagai berikut :

- a. Bagaimana mengimplementasikan deteksi tepi dalam pemrograman paralel menggunakan pustaka OpenMPI
- b. Bagaimana mengetahui *speedup* komputasi pada pemrograman paralel dalam melakukan pendeteksian tepi pada gambar
- c. Bagaimana pengaruh *speedup* pada pemrograman paralel dalam *cluster* yang berbeda
- d. Bagaimana mengukur *benchmark* pada *cluster* untuk mengetahui performa *cluster* yang telah dibangun

## 1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah :

- a. Menggunakan sistem operasi Linux.
- b. Menggunakan bahasa pemrograman c++ dalam membangun *code* paralel.
- c. Menggunakan pustaka OpenMPI sebagai aplikasi yang mendukung pemindahan tugas.
- d. Melakukan pendeteksian tepi pada gambar dengan menggunakan operasi Sobel
- e. Menggunakan *library* OpenCV untuk pengolahan gambar
- f. Input deteksi tepi berupa citra warna RGB

## 1.4 Tujuan Penelitian

Tujuan yang ingin dicapai adalah mengetahui *speedup* dalam menjalankan komputasi paralel dan pengaruh *speedup switch* dalam melakukan *image processing*



yakni dengan menjalankan deteksi tepi secara paralel pada *cluster* yang berbeda dan mengetahui *benchmark* pada *cluster*.

## 1.5 Sistematika Penulisan

Sistematika penulisan laporan tugas akhir ini terdiri dari pokok-pokok permasalahan yang dibahas pada masing-masing bab yang diurai menjadi beberapa bagian :

### BAB I PENDAHULUAN

Bab ini membahas tentang gambaran umum penelitian yang meliputi latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian dan sistematika penulisan.

### BAB II LANDASAN TEORI

Bab ini berisi tentang dasar teori yang berhubungan dengan penelitian yang terdiri dari penjelasan mengenai *cluster* komputer, komputasi paralel, teori tentang MPI, OpenMPI.

### BAB III METODOLOGI PENELITIAN

Bab ini berisi tentang metodologi penelitian yang terdiri dari studi pustaka / *literature*, perancangan *cluster*, implementasi dan pengujian terhadap *cluster*.

### BAB IV ANALISA DAN PERANCANGAN *CLUSTER*

Bab ini berisi tentang tahapan tahapan merancang sebuah *cluster*. Dari tahapan instalasi sistem operasi keseluruhan *nodes* hingga mengintegrasikan keseluruhan *nodes*.

### BAB V IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi tentang implementasi bahasa pemrograman c++ menggunakan *cluster* yang telah dibangun dan melakukan pengujian terhadap *cluster* yang telah dibangun.

### BAB VI PENUTUP

Bab ini berisi tentang kesimpulan yang didapatkan dari hasil penelitian yang telah dilakukan serta berisi saran-saran sebagai hasil akhir yang diperlukan untuk proses pengembangan selanjutnya.

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.