

## **BAB II**

### **TINJAUAN PUSTAKA**

Sampai tahun 2006, pemrograman paralel sulit untuk dibuat pada GPU. Ini dikarenakan pemrograman pada GPU harus menggunakan *Application Programming Interface* (API) untuk dapat mengakses inti prosesor yang terdapat pada GPU. Pemrograman pada GPU juga harus menggunakan OpenGL dan Direct3D untuk dapat memprogram inti tersebut sehingga hanya sedikit orang yang menguasai kemampuan ini sehingga dapat menggunakan inti tersebut agar mengoptimalkan kinerja pada aplikasi. Maka pemrograman pada GPU tidak tersebar dengan luas dan tidak digunakan oleh banyak orang (Kirk & Hwu, 2010).

Hal ini berubah dengan diperkenalkannya suatu teknik komputasi yang baru oleh NVIDIA pada tahun 2007. Teknik komputasi tersebut dinamakan CUDA yang membuat GPU tidak hanya digunakan pada game saja tetapi juga untuk melakukan komputasi yang umum (GPGPU – *General Purpose Graphic Processing Unit*). CUDA merupakan pelopor untuk komputasi paralel yang berkembang hingga saat ini. CUDA juga menggunakan bahasa pemrograman yang dikenal umum yaitu C/C++ sehingga mudah untuk digunakan.

Komputasi paralel dengan menggunakan CUDA untuk mempercepat proses telah dilakukan oleh peneliti-peneliti sebelumnya. Dapat dilihat pada penelitian yang dilakukan untuk melihat keuntungan dan kelemahan pada pemrograman

CUDA (Che, et al., 2008) dan fitur-fitur yang dapat digunakan pada CUDA (Garland, et al., 2008). CUDA pun dapat digunakan pada pengolahan citra (Castano-Diez, et al., 2008) (Yang, et al., 2008) (Park, et al., 2011) mendapatkan bahwa dengan mengimplementasikan metode pengolahan citra pada GPU dapat mempercepat kinerja dengan kecepatan yang variatif bergantung pada karakteristik algoritma yang diterapkan, yaitu dapat meningkatkan kecepatan sebanyak 10 – 20x, 40x, 80x, bahkan 200x.

Salah satu bagian dari pengolahan citra adalah segmentasi citra. Segmentasi citra merupakan proses menentukan bagian-bagian citra. Banyak metode yang digunakan untuk segmentasi citra, misalnya metode *edge detection* (Senthilkumaran & Rajesh, 2009), *light refraction* (Guvenc, et al., 2010), *level set* (Li, et al., 2011), *distance regularized level set evolution* (DRLSE) (Li, et al., 2010) dan *deformable model* (Jayadevappa, et al., 2009).

Penelitian segmentasi citra dengan menggunakan DRLSE telah dilakukan (Liu & Liu, 2011) (Rani, et al., 2011) (Kaur & Jindal, 2012). Liu, et al melihat bahwa DRLSE menggunakan *Gaussian filter* yang dapat mengurangi *noise* tetapi membuat gambar menjadi kabur, untuk mengatasi hal tersebut Liu mengadopsi *regularized P-M equation* yang dapat mengatasi *noise* dan tetap menjadi informasi *edge*. Sehingga dapat dengan tepat mengambil bagian dalam citra dan mengurangi waktu komputasi. Rani, et al membuat penelitian segmentasi pada tumor otak dengan menggunakan DRLSE, hasil penelitian didapati bahwa metode ini dapat melakukan segmentasi dengan tepat. Sedangkan penelitian yang

dilakukan oleh Kaur, et al adalah melakukan segmentasi dengan menggunakan *active contour without edges*, *localized active contour without edges*, dan DRLSE serta membandingkan hasilnya. Didapati bahwa DRLSE mempunyai kesalahan komputasi numerik yang sedikit dan tidak perlu dilakukan *reinitialization*, tetapi metode ini rentan terhadap *noise*.

Dalam penelitian ini, metode segmentasi DRLSE diaplikasikan dalam menentukan bagian-bagian pada citra medis. Proses segmentasi tersebut membutuhkan waktu yang relatif lama jika citra medis berukuran besar. Untuk mengatasi hal tersebut metode segmentasi DRLSE dipadukan dengan CUDA, yang merupakan API komputasi paralel pada GPU. Diharapkan dengan menggunakan GPU dapat mempercepat proses komputasi dalam segmentasi. Hasil segmentasi dengan menggunakan CPU dan segmentasi dengan menggunakan GPU dibandingkan dan dianalisa kinerjanya.

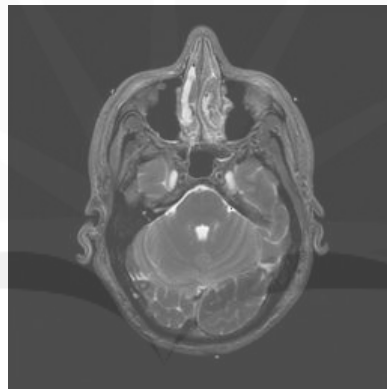
#### **A. Citra Medis**

Citra merupakan gambar yang merepresentasikan sesuatu. Citra dapat berupa gambar dari seseorang, orang banyak atau hewan, atau suatu pemandangan luar, atau *microphotograph* dari suatu komponen elektronik atau juga hasil dari pencitraan medis (McAndrew, 2004)

Pencitraan medis merupakan citra yang dibuat dengan menggunakan beberapa teknologi yang berbeda yang digunakan untuk melihat tubuh manusia bertujuan untuk mengdiagnosa, memonitor dan untuk memeriksa kondisi medis.

Setiap teknologi memberikan informasi yang berbeda mengenai area dari tubuh manusia yang dipelajari atau diperiksa, yang berhubungan dengan penyakit, kecelakaan atau untuk melihat perkembangan pemeriksaan medis.

Salah satu alat yang digunakan untuk membuat citra medis adalah dengan menggunakan MRI. MRI membuat medan magnet yang temporer di sekitar tubuh pasien. Gelombang radio dikirimkan dan diterima dengan *transmitter/receiver* dalam mesin, dan kemudian sinyal tersebut membuat citra digital dari area yang diinginkan.



Gambar 2. 1. Contoh Citra Hasil dari MRI

## **B. Segmentasi Citra**

Segmentasi merupakan proses untuk membagi citra digital menjadi beberapa segmen. Tujuan untuk melakukan segmentasi citra adalah dengan membuat citra tersebut agar dapat lebih berarti dan lebih mudah untuk dianalisa. Terdapat banyak metode untuk melakukan segmentasi citra, yaitu *intensity thresholding*, *region growing and region splitting*, *edge detection*, *interest*

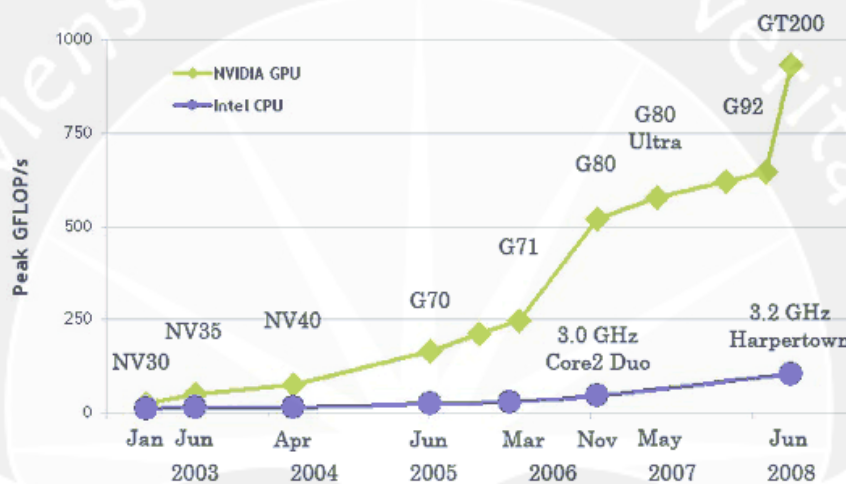
*operators, watershed segmentation, markov random model* (Solomon & Breckon, 2011), metode Generalized Fuzzy C-Means Clustering Algorithm (Hardiyanto, Purwananto and Soelaiman 2012), Threshold-based, Edge-based dan Region-based (Zuva, et al. 2011), Watershed Transform (Richard and Fernandez-Maloigne 2013), Particle Swam Optimization (Mohsen, Hadhound and Amin 2011), Adaptive K-means Clustering Algorithm (Petel and Sinha 2010), Hybrid Technique yang meliputi vector median filtering, Otsu thresholding, dan EnSFCM (Nyma, et al. 2012), Distance Regularization Level Set Evolution (DRLSE) (Li, et al. 2010).

### **C. Komputasi Paralel**

Sejak tahun 2003, ada dua perkembangan utama pada pembuatan mikroprosesor, yaitu *multicore* dan *many-core* (Kirk & Hwu, 2010). *Multicore* dimulai dengan prosesor berinti dua yang kemudian berkembang sampai ke prosesor berinti 8. Tujuan *multicore* adalah mempertahankan kinerja untuk menjalankan program sekuensial. *Multicore* dapat dilihat pada bentuk CPU. Sedangkan *many-core* terfokus pada aplikasi paralel. *Many-core* lebih banyak memiliki unit pemroses lebih banyak dibandingkan *multicore*. *Many-core* banyak dijumpai pada bentuk GPU.

Perbedaan yang mendasar antara CPU (*multicore*) dan GPU (*many-core*) didasarkan pada dasar desain pada kedua tipe prosesor. CPU dibuat untuk mengoptimalkan program sekuensial sedangkan GPU dibuat untuk

mengoptimalkan program parallel. Hal yang lain yang mendasari perbedaan ini adalah *bandwidth* pada memory. GPU mempunyai *bandwidth* 10 kali dibandingkan CPU. Tetapi pada dasarnya GPU tidak dapat menggantikan CPU karena GPU hanya untuk membuat komputasi paralel untuk program paralel dan bukan untuk program sekuensial.



Gambar 2. 2. Perbandingan Performa CPU dan GPU

#### D. OpenGL

OpenGL merupakan bahasa pemrograman umum untuk aplikasi visualisasi 2D maupun 3D. OpenGL adalah bahasa pemrograman yang handal tetapi juga merupakan bahasa pemrograman yang sangat rumit untuk digunakan (Heard, 2008).

Sebagai suatu antar muka perangkat lunak pada perangkat keras, tujuan utama dari OpenGL adalah untuk membuat objek 2 dimensi dan objek 3 dimensi pada bingkai. Objek-objek tersebut berupa deretan dari vector-vektor (yang

didefinisikan sebagai objek geometri) atau piksel (yang didefinisikan sebagai gambar atau citra). OpenGL melakukan beberapa langkah proses pada data-data tersebut dan dirubah ke dalam piksel untuk membuat gambar atau citra yang diinginkan pada sebuah bingkai (Silicon Graphics, 1994).

