

GPU ACCELERATED FUZZY C-MEANS (FCM) COLOR IMAGE SEGMENTATION

Mutaqin Akbar¹⁾, Arita Witanti²⁾, Indah Susilawati³⁾

^{1,2,3}Teknik Informatika, Fakultas Teknologi Informasi

Universitas Mercu Buana Yogyakarta

Jl. Wates Km. 10 Yogyakarta 55753

Email : ¹mutaqin@mercubuana-yogya.ac.id, ²arita@mercubuana-yogya.ac.id,
³indah@mercubuana-yogya.ac.id

Abstract

In this paper, computational acceleration of color image segmentation using fuzzy c-means (FCM) algorithm has been presented. The color image is first converted from the Red Green Blue (RGB) color space to the YUV color space. Then, the luma (Y) information values are grouped according to the desired number of clusters using the FCM algorithm. The FCM algorithm is implemented on a Graphical Processing Unit (GPU) using the Compute Unified Device Library (CUDA) library which is developed by NVidia to speed up the computing time. Images used in this research are red blood cell images, geometry images and leaf images. The results of segmented images processed using GPU were seen identic to the results of segmented images processed using the Central Processing Unit (CPU). The computational time of the FCM algorithm can be accelerated by speed-up to 5,628 times faster and the average speed-up of all simulations done is 5,517 times faster.

Keywords : computational acceleration, FCM, GPU, image segmentation

1. Latar Belakang Masalah

Segmentasi citra merupakan proses yang ditujukan untuk mendapatkan objek-objek yang terkandung didalam citra atau membagi citra kedalam beberapa daerah dengan setiap objek atau daerah memiliki kemiripan atribut[1]. Segmentasi citra sering digunakan pada bidang medis contohnya citra *scan* otak, tulang, maupun citra mikroskopis seperti sel darah atau sel mikroba. Segmentasi citra juga sering digunakan untuk preprocessing dalam hal pengenalan pola untuk *machine learning* dan sejenisnya.

Segmentasi citra berdasarkan pada dua dasar, yaitu ketidaksinambungan dan kesamaan antar pixel. Yang pertama adalah pemisahan citra yang berdasar pada perubahan mendadak pada aras keabuan. Contohnya: *edge detection* dan *line detection*[2]. Cara kedua berdasarkan pada kesamaan antar pixel didalam suatu area. Contoh cara kedua ini yaitu: *thresholding*, *region growing*, *region merging and splitting*, dan *clustering*.

Proses *clustering* memisahkan intensitas warna piksel kedalam kelompok dengan menentukan kesamaan piksel dalam satu kelompok dengan kelompok lainnya. Untuk intensitas warna citra, sebuah ruang warna adalah ruang alam, dan menerapkan warna citra untuk segmentasi dengan pendekatan *clustering* adalah ide yang sederhana, karena warna cenderung membentuk kelompok diruang warna. Segmentasi citra berbasis ruang warna dianggap mampu mengurangi waktu komputasi untuk menghindari perhitungan fitur[3]. Sementara Wicaksono [4] menambahkan tekstur menggunakan filter gabor sehingga secara persepsi manusia, hasil segmentasi lebih halus.

Beberapa algoritma *clustering* yang sering digunakan adalah *K-Means* (KM) dan *Fuzzy C-Means* (FCM). Menurut Cebeci (dan kawan-kawan) [5] KM memiliki keunggulan yaitu eksekusinya lebih cepat dibandingkan dengan FCM. Sedangkan FCM

lebih unggul dalam hal akurasi tetapi waktu komputasinya besar. Safitri (dan kawan-kawan) [6] membandingkan algoritma FCM dan spatial FCM (sFCM), tetapi tidak didapati kesimpulan mutlak mengenai algoritma yang menghasilkan segmentasi terbaik.

Kelemahan FCM dalam hal waktu komputasi dapat diatasi dengan melakukan akselerasi proses komputasi dengan bantuan *Graphical Processing Unit* (GPU). GPU biasa dimanfaatkan untuk tujuan umum yang dikena juga dengan sebutan *General Purpose Graphic Processing Unit* (GPGPU). Pengurutan, pencarian, *query* basis data, persamaan diferensial, maupun aljabar linier merupakan beberapa aplikasi komputasi pada GPU[7]. GPU mempunyai banyak *thread*, yang dapat bekerja bersama mempercepat proses yang terdapat pada algoritma FCM untuk setiap informasi warna dalam sebuah citra secara serentak, dibandingkan dengan proses sekuensial yang dilakukan oleh *Central Processing Unit* (CPU). Akbar (dan kawan-kawan) [8] melakukan akselerasi komputasi pada algoritma Alternating Direction-Implicit (ADI) untuk inpainting citra dan mendapatkan percepatan sebesar 5.86 kali dengan citra berresolusi 2736x1824.

Proses *clustering* menggunakan FCM hanya menggunakan masukan berupa 1 nilai, sehingga dibutuhkan proses transformasi dari ruang warna standar *Red Green Blue* (RGB) kedalam sebuah informasi warna. Beberapa ruang warna yaitu *Hue Saturation Intensity* (HIS), *Hue Saturation Value* (HSV), $L^*u^*v^*$, $L^*a^*b^*$, YUV, dan lain sebagainya. Kesemua ruang warna dapat digunakan untuk segmentasi dengan kelemahan dan kelebihan tersendiri. Ruang warna YUV mudah diimplementasikan, sehingga diambil informasi luma (Y) untuk dijadikan masukan pada proses *clustering* menggunakan FCM.

2. Metodologi Penelitian

2.1 Tahapan Penelitian

Tahapan penelitian yang dilakukan terdiri dari :

1. Studi literatur, dilakukan dengan mempelajari jurnal dan buku yang berhubungan dengan pengolahan citra[1][2], FCM[3][4][6], dan pemrograman berbasis GPU[7][8][9][10].
2. Akuisisi dan analisa data, dilakukan dengan mendapatkan data citra yang akan diteliti dari berbagai sumber seperti :
 - a. Citra sel darah merah[11],
 - b. Citra geometri,
 - c. Citra daun.
3. Perancangan, membuat dokumentasi alur program.
4. Implementasi dan pengujian, dilakukan dengan membuat 2 program yaitu program berbasis CPU dan program berbasis GPU, kemudian dilakukan pengujian terhadap hasil dan juga kecepatan eksekusi terhadap data yang menjadi masukan.

2.2 Prinsip Kerja Sistem

2.2.1. Fuzzy C-Means (FCM)

Algoritma FCM menetapkan piksel untuk setiap kelompok dengan menggunakan keanggotaan *fuzzy* u_{ij}^m . Setiap citra mempunyai n piksel dengan ditulis simbol x_i $\{i = 1, 2, 3, \dots, n\}$ yang akan dibagi menjadi c kelompok (*cluster*). Pada metode ini juga digunakan suatu variabel m yang merupakan *weighting exponent* dari keanggotaan fuzzy. Variabel m mempunyai wilayah nilai $m > 1$. Nilai keanggotaan *fuzzy* menentukan pusat

cluster y_j maka digunakan untuk memperoleh nilai fungsi obyektif yang minimal dari rumus sebagai berikut :

$$J_m = \sum_{j=1}^c \sum_{i=1}^n u_{ij}^m d_{ij}^2 \quad (1)$$

Dimana,

$$d_{ij} = \|x_i - y_j\|$$

Fungsi keanggotaan u_{ij} dapat dijabarkan sebagai berikut,

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{d_{ij}}{d_{kj}}\right)^{2/(m-1)}} \quad (2)$$

y_j adalah centroid dari j th *cluster*, yang dihasilkan dari :

$$y_j = \frac{\sum_{i=1}^n u_{ij}^m x_i}{\sum_{i=1}^n u_{ij}^m} \quad (3)$$

2.2.2. Ruang Warna

Warna dipersepsikan oleh manusia sebagai kombinasi dari *tristimuli Red* (R), *Green* (G), dan *Blue* (B), yang biasanya disebut dengan tiga warna utama. Dari ruang warna RGB, kita dapat menghitung berbagai ruang warna lain dengan menggunakan baik transformasi linier maupun non-linier. Beberapa ruang warna yaitu *Hue Saturation Intensity* (HIS), *Hue Saturation Value* (HSV), $L^*u^*v^*$, $L^*a^*b^*$, YUV, dan lain sebagainya. Kesemua ruang warna dapat digunakan untuk segmentasi dengan kelemahan dan kelebihan tersendiri.

Ruang warna YUV sering digunakan oleh standar video warna *Phase Alternation Line* (PAL), *National Television System Committee* (NTSC), dan *Sequentiel Couleur Avec Memoire or Sequential Color with Memory* (SECAM). System warna hitam dan putih hanya menggunakan informasi luma (Y). Konversi dari ruang warna RGB ke YUV dapat dilihat pada rumus berikut[12] :

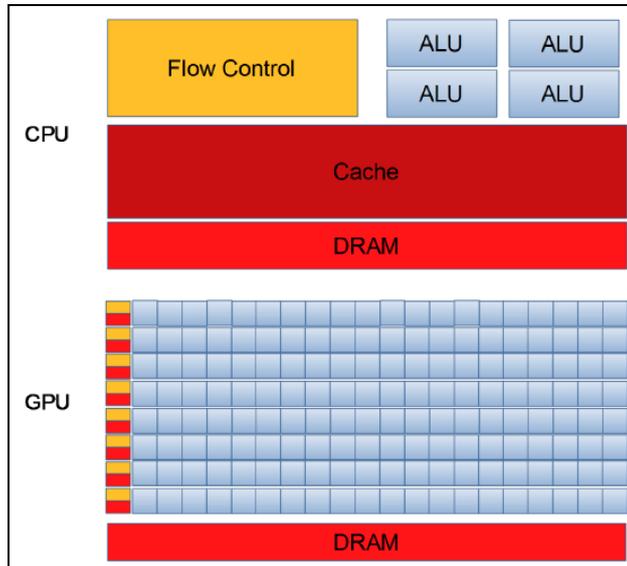
$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ U &= 0.565(B - Y) \\ V &= 0.713(R - Y) \end{aligned} \quad (4)$$

Hanya informasi luma (Y) yang akan digunakan sebagai masukan pada proses *clustering* menggunakan FCM.

2.2.3. Komputasi Paralel Berbasis GPU

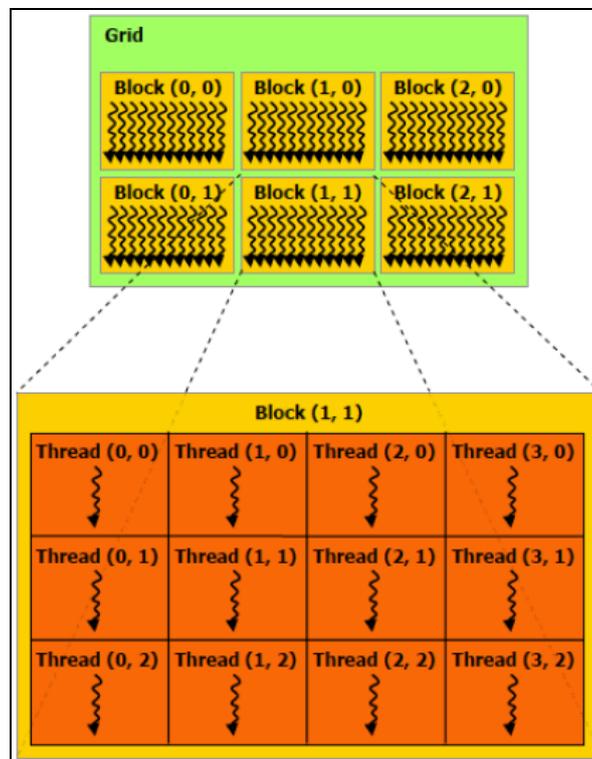
Dewasa ini, paralelisme menjadi tenar dimana-mana, dan pemrograman paralel menjadi perbincangan utama di dunia pemrograman. Terdapat dua jenis paralelisme dasar dalam aplikasi, yaitu: *task parallelism* dan *data parallelism*. *Task parallelism* muncul ketika terdapat banyak tugas atau fungsi yang dapat dijalankan secara independen dan sebagian besar paralel. *Task parallelism* fokus pada distribusi fungsi ke beberapa inti (*cores*) yang ada. *Data parallelism* muncul ketika ada banyak data yang dapat dioperasikan disaat yang sama. *Data parallelism* berfokus pada pendistribusian data lintas

banyak inti. Pemrograman CUDA sangat cocok untuk mengatasi masalah yang dapat dinyatakan sebagai komputasi *data-parallel*.



Gambar 1. Perbedaan CPU dan GPU[9]

GPU memiliki banyak *thread*, yang dapat dikombinasi menjadi *block*, dimana *block* tersebut melakukan proses terhadap bagian tertentu dari banyak data. Kombinasi banyak *block* disebut juga *grid*. *Thread*, *block* dan *grid* tersusun sebagai berikut :



Gambar 2. Grid, Block, dan Thread[10]

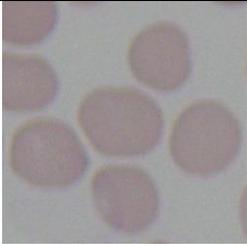
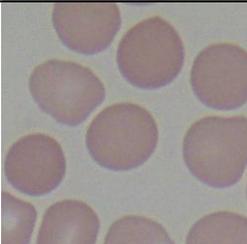
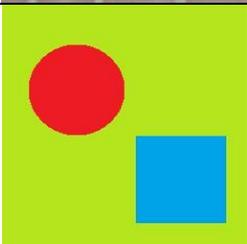
3. Hasil dan Pembahasan

Simulasi-simulasi sudah dilakukan menggunakan sebuah computer personal dengan spesifikasi sebagai berikut :

- CPU Intel(R) Core(TM) i3-4005U @1.70GHz
- RAM 4GB
- GPU GeForce 930M

Bahan penelitian yang digunakan berupa citra berwarna dengan format *.png* maupun *.jpg* dan memiliki resolusi citra sebesar 256 x 256. Beberapa citra berwarna yang digunakan dalam simulasi dapat dilihat pada Tabel 1 dibawah ini.

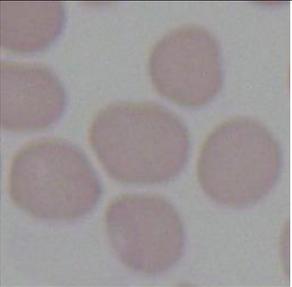
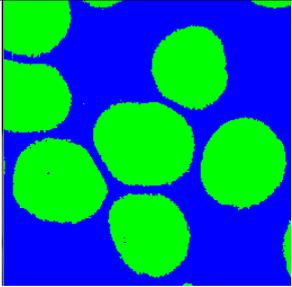
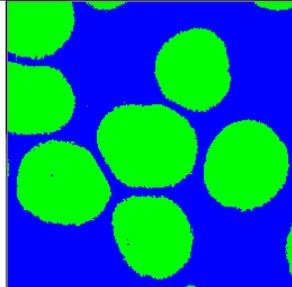
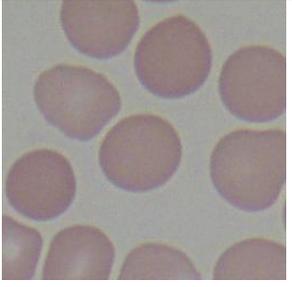
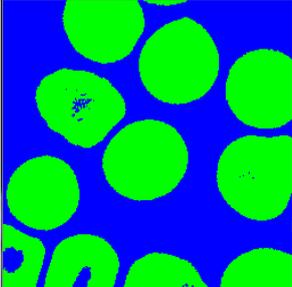
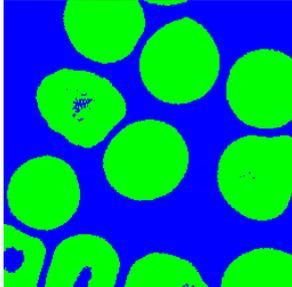
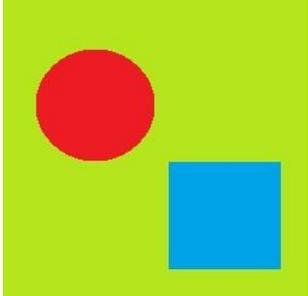
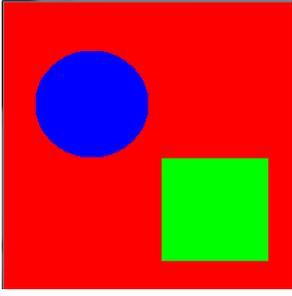
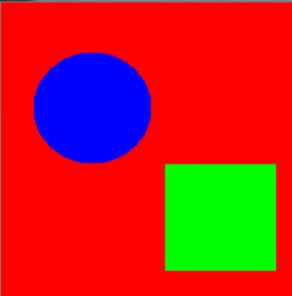
Tabel 1. Citra Berwarna yang Digunakan untuk Simulasi

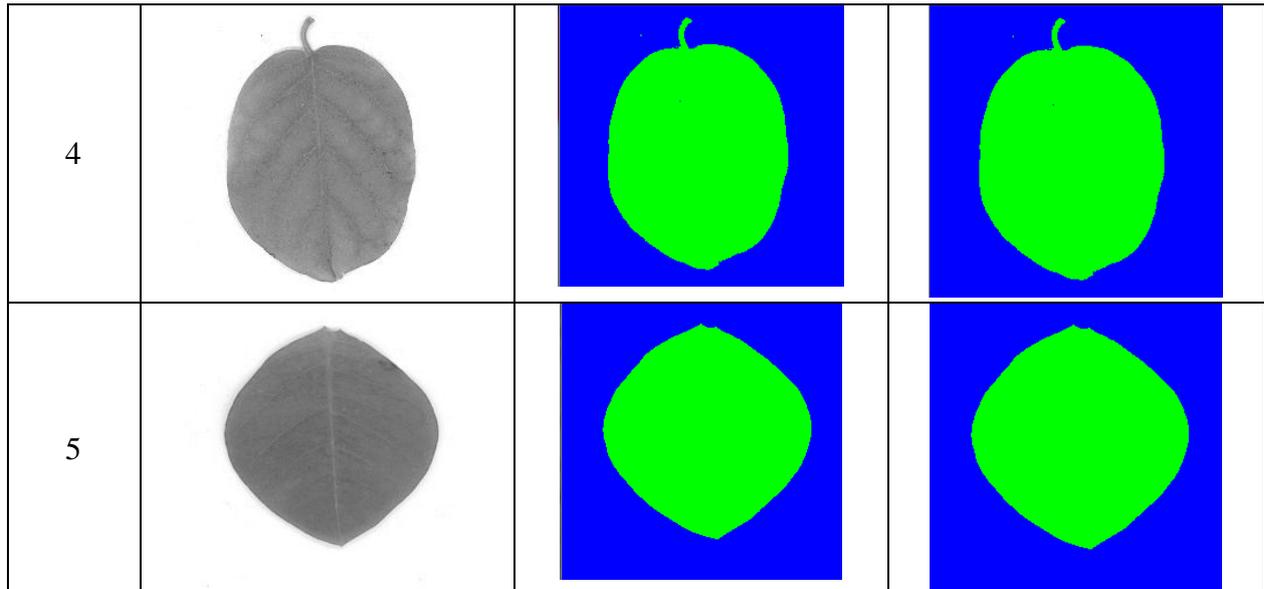
No. Citra	Resolusi Citra	Citra Berwarna
1	256 x 256	
2	256 x 256	
3	256 x 256	
4	256 x 256	
5	256 x 256	

Citra pada nomor 1 dan 2 merupakan citra sel darah merah, dimana sel darah merah pada citra nomor 1 dapat diamati berbentuk bulat penuh. Sedangkan pada citra nomor 2 sel darah merah terdapat beberapa sel yang dibagian tengahnya berwarna menyerupai latar belakang citra tersebut, sehingga dapat kita amati perbedaan hasil segmentasi antara citra nomor 1 dan nomor 2. Dan pada citra nomor 3, kita temui citra uji berupa bentuk geometri lingkaran berwarna merah, bujursangkar berwarna biru dan latar belakang berwarna hijau. Untuk citra nomor 4 dan 5 merupakan citra daun dengan bentuk yang berbeda.

Setelah melalui simulasi, maka hasil dari proses clustering menggunakan FCM pada citra uji dengan komputasi berbasis GPU dapat dilihat pada Tabel 2 dibawah. Sebagai informasi hasil simulasi, maka tiap *cluster* diberi informasi warna tersendiri dengan rincian yaitu: warna biru (B) menunjukkan *cluster* ke-1, hijau (G) menunjukkan *cluster* ke-2 dan merah (R) menunjukkan *cluster* ke-3.

Tabel 2. Hasil Simulasi Proses *Clustering* menggunakan FCM dengan Komputasi Berbasis GPU

No. Citra	Citra Asli	Citra Tersegmentasi (CPU)	Citra Tersegmentasi (GPU)
1			
2			
3			



Dapat diamati pada Tabel 2 nomor citra 1 citra sel darah merah dapat tersegmentasi dengan baik untuk setiap sel darah merah. Tetapi pada citra nomor 2, terdapat beberapa sel darah merah dimana pada bagian tengah selnya masuk kedalam *cluster* warna latar belakang citra. Sedangkan pada citra uji berbentuk geometri, citra daun nomor 4 dan citra daun nomor 5, jelas sekali bentuk-bentuk baik geometri maupun daun tersegmentasi dengan baik. Dari hasil simulasi menggunakan CPU dan GPU, tidak terdapat perbedaan mencolok mengenai hasil citra tersegmentasi. Dapat diamati hasil antara keduanya terlihat sama persis secara kasat mata.

Untuk performa waktu komputasi baik dengan komputasi dengan CPU maupun dengan GPU dapat dilihat pada Tabel 3 berikut.

Tabel 3 Performa Waktu Komputasi CPU dan GPU

Nomor Citra	Jumlah Iterasi	Jumlah Cluster	Waktu CPU (ms)	Waktu GPU (ms)	Speed-up
1	15	2	957.000	195.694	4.890
2	15	2	941.000	189.999	4.953
3	15	3	1990.000	353.568	5.628
4	15	2	936.000	196.003	4.775
5	15	2	936.000	197.422	4.741
Rata-rata			1296.000	246.420	4.997

Beberapa hal yang dapat dilihat dari hasil simulasi pada Tabel 3, waktu komputasi dengan CPU pada citra nomor 1, citra nomor 2, citra nomor 4, dan citra nomor 5 tidak terpaut terlalu jauh yaitu masing-masing 957.000 ms, 941.000 ms, 936.000 ms, dan 936.000 ms, tetapi pada citra nomor 3 dapat kita lihat waktu proses komputasinya lebih besar yaitu 1990.000 ms. Hal serupa juga terjadi pada waktu komputasi berbasis GPU, dimana waktu komputasi untuk citra nomor 1, citra nomor 2, citra nomor 4, citra nomor 5 masing-masing yaitu 193.694 ms, 189.999 ms, 196.003 ms, dan 197.422 ms. Hal ini disebabkan karena jumlah *cluster* pada citra nomor 1 dan citra nomor 2 sama yaitu sebanyak 2 buah *cluster*, sedangkan pada citra nomor 3 sebanyak 3 buah *cluster*. Jumlah *cluster* mempengaruhi lamanya waktu proses komputasi baik untuk CPU maupun GPU.

Kemudian dengan membandingkan waktu simulasi yang sudah dilakukan dengan komputasi berbasis CPU dan komputasi paralel berbasis GPU, maka dapat dikatakan bahwa percepatan waktu komputasi dengan komputasi paralel berbasis GPU dapat mencapai 5.628 kali, dengan rata-rata 4.997 kali.

4. Kesimpulan

Makalah ini menyajikan sebuah akselerasi komputasi segmentasi citra berwarna menggunakan pemrosesan paralel berbasis GPU dengan bantuan *library* CUDA besutan NVidia. Pemrosesan paralel digunakan untuk mempercepat proses *clustering* menggunakan algoritma FCM. Waktu proses komputasi dapat dipercepat (*speed-up*) hingga mencapai 5.628 kali dengan rata-rata *speed-up* yaitu 4.997 kali.

Sebagai pertimbangan untuk pengembangan lebih lanjut dapat digunakan citra dengan objek lebih banyak lagi, citra dengan resolusi lebih besar lagi, dan juga menggunakan GPU dengan spesifikasi yang lebih tinggi lagi.

5. Ucapan Terimakasih

Penulis mengucapkan terima kasih kepada Universitas Mercu Buana Yogyakarta yang telah mendanai penelitian ini dan Fabio Scotti atas ketersediaan data citra sel darah merah.

Daftar Pustaka

- [1] Kadir, A., & Susanto, A. (2013). Teori dan Aplikasi Pengolahan Citra. Yogyakarta: ANDI
- [2] Muthukrishnan, R., & Radha, M. (2011). Edge detection techniques for image segmentation. *International Journal of Computer Science & Information Technology*, 3(6), 259
- [3] Wicaksono, Y. Segmentasi Citra Warna Dan Tekstur Menggunakan Fuzzy C-Means Dan Filter Gabor
- [4] Saikumar, T., Yugander, P., Sreenivasa, P., & Smitha, B. (2011). Colour based image segmentation using fuzzy c-means clustering. In *Proceedings of International Conference on Computer and Software Modeling, Singapore* (pp. 180-5)
- [5] Cebeci, Z., & Yildiz, F. (2015). Comparison of K-Means and Fuzzy C-Means Algorithms on Different Cluster Structures. *Journal of Agricultural Informatics* , 6 (3), 13-23
- [6] Safitri, Q. U., Huda, A. F., & A., A. S. (2017). Segmentasi Citra Menggunakan Algoritma Fuzzy C-Means (FCM) dan Spatial Fuzzy C-Means (sFCM). *Jurnal Kubik* , 2 (1), 22-34
- [7] Owens, J. D., Houston, M., Luebke, D., Green, S., Stones, J. E., & Phillips, J. C. (2008). GPU Computing. *Proceedings of the IEEE* , 879-899
- [8] Akbar, M., Pranowo, & Suyoto. (2017). Computational Acceleration of Image Inpainting Alternating-Direction Implicit (ADI) Method Using GPU CUDA. *The 2017 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)* (pp. 186-190). Yogyakarta: Telkom University
- [9] Xu, L. (2011). Parallel Computing based on GPGPU using Compute Unified Device Architecture. Royal Institute of Technology (KTH)
- [10] Tse, J. J. (2012). Image processing with CUDA.

- [11] Labati, R. D., Piuri, V., & Scotti, F. (2011). ALL-IDB: The Acute Lymphoblastic Leukemia Image Database for Image Processing. The 2011 IEEE International Conference on Image Processing (ICIP 2011) (pp. 2045-2048). Brussels: IEEE
- [12] Saravanan, C. (2010). Color Image to Grayscale Image Conversion. The 2010 Second International Conference on Computer Engineering and Applications (ICCEA). Bali: Seoul National University

