

Penerapan Multi-threading untuk Meningkatkan Kinerja Pengolahan Citra Digital

Megah Mulya¹, dan Abdiansah²

^{1,2}Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Sriwijaya
Jl. Raya Palembang-Prabumulih Kampus UNSRI Inderalaya
E-mail: ¹megahmulya@yahoo.com, dan ²abdiansah84@gmail.com

Abstrak

Penelitian ini menggunakan teknik multi-threading dalam pengolahan citra digital untuk menghasilkan pengolahan citra digital yang lebih cepat. Selain itu, penelitian ini membandingkan kecepatan akurasi antara single-threading dengan multi-threading. Hasil pengujian menggunakan teknik multi-threading memperlihatkan waktu proses semakin bertambah cepat, bila jumlah sampel bertambah banyak. Hal ini menunjukkan bahwa teknik multi-threading memiliki waktu proses yang optimal dalam pengolahan citra digital dibandingkan dengan single-threading.

Kata kunci: waktu proses, single-threading, multi-threading.

Abstract

This research used multi-threading technique for digital image processing to achieve better digital image processing. Besides, this research was to compare the velocity of accuracy between single-threading and multi-threading. The testing results of multi-threading technique showed the processing time more quickly, when using more number of samples. It showed that multi-threading technique had optimal time processing for digital image processing than single-threading.

Keywords: time processing, single-threading, multi-threading.

1. Pendahuluan

Pengolahan citra digital memiliki peranan yang cukup penting dalam mengawal perkembangan teknologi komputer. Dewasa ini, citra digital memegang peranan cukup vital sebagai media yang sering digunakan oleh pengguna teknologi komputer, seperti *handset* dan *digital camera* yang dapat menghasilkan citra digital, dan teknologinya terus menerus dalam pengembangan. Beragam aplikasi komputer dapat digunakan untuk memanipulasi citra digital bertujuan untuk menghasilkan karya seni dan juga untuk kepentingan-kepentingan lain.

Citra digital memiliki beberapa karakteristik yang berbeda dengan citra analog. Citra digital dan analog sama-sama memiliki dimensi panjang dan lebar serta komponen warna, tetapi citra analog tidak memiliki ukuran *file* dan kedalaman warna yang hanya dimiliki oleh citra digital. 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. Salah satu indikator kerja proses komputasi citra digital dapat dilihat berdasarkan lamanya proses pengolahan citra. Pemrosesan yang lebih cepat sangat

diharapkan daripada pemrosesan yang lambat. Kecepatan pengolahan suatu citra menjadi topik penting mengingat pengolahannya sering dilakukan secara non-paralel.

Disamping itu, sejarah perkembangan citra digital cukup panjang yang awalnya hanya berwarna hitam-putih (*monochrome*) beralih menjadi abu-abu (*grayscale*) dan kemudian menjadi berwarna yang memiliki beragam tipe. Selain itu ukuran citra digital beragam dari yang hanya kb (*kilobyte*) sampai giga (*gigabyte*), seperti citra satelit yang digunakan oleh NASA.

Multi-threading merupakan salah satu teknik dalam pemrograman komputer yang bertujuan untuk membuat proses berjalan secara paralel (bersamaan). Dengan teknik ini, dapat dimungkinkan untuk membuat program komputer melakukan pekerjaan yang banyak dalam satu waktu. Penggunaan teknik ini memiliki keuntungan dan kerugian tergantung dari kasus yang dihadapi. Keuntungannya adalah dapat membuat pekerjaan lebih cepat karena dilakukan secara paralel sedangkan kerugiannya adalah teknik ini memerlukan *cost* yang sangat besar karena menggunakan *resource* yang lebih banyak dibandingkan dengan *single-threading*. Kendala *cost* yang tinggi tidak terlalu berdampak dikarenakan kapasitas memori yang sudah semakin besar dan harga yang terjangkau. Salah satu bidang yang memerlukan pengolahan citra dengan kecepatan yang tinggi adalah bidang pengolahan citra dengan waktu nyata (*realtime image processing*), seperti: *video conference*, *satellite imaging* dan lainnya.

Penelitian bidang citra digital masih terus dikembangkan, hal ini disebabkan oleh karakteristik citra digital yang terus berkembang. Oleh karena itu, penelitian ini menggunakan teknik *multi-threading* dalam pengolahan citra digital untuk mendapatkan hasil pengolahan citra yang lebih cepat dan melihat signifikansi kecepatan antara *single-threading* dengan *multi-threading*.

2. Pengolahann Citra

Proses pengolahan citra sangat beragam bentuknya, ada tiga bentuk umum proses pengolahan citra digital, yaitu operasi tingkat rendah (*low level operation*), operasi tingkat menengah (*intermediate level operation*) dan operasi tingkat tinggi (*high level operation*) [1]. Pada operasi tingkat rendah, citra ditransformasikan ke dalam citra yang siap dimodifikasi. Struktur citra dimodifikasi secara keseluruhan dan menghasilkan citra baru, vector atau nilai tunggal. Pada tingkatan ini operasi citra dilakukan secara langsung piksel per piksel, contoh operasinya adalah *smoothing*, *convolution*, *histogram generation* dan lainnya.

Pada tingkat menengah, citra ditransformasikan ke dalam struktur data. Struktur citra dimodifikasi sebagian karena operasi ini hanya mengambil objek-objek yang dianggap penting untuk diolah atau yang sering disebut dengan ROI (*Region of Interest*). Contoh operasi ini adalah *region labeling*, *motion analysis* dan lainnya. Terakhir operasi tingkat tinggi, pada tingkat ini citra diekstraksi dan diambil informasinya. Informasi tersebut diolah lagi untuk mendapatkan aksi atau membuat keputusan, tingkat ini sering disebut juga dengan pemrosesan simbolik (*symbolic processing*). Contohnya adalah *object recognition*, *face recognition* dan lainnya.

Setiap pengolahan citra memerlukan serangkaian proses komputasi yang tidak sebentar. Contohnya, seperti proses konvolusi sederhana untuk meningkatkan kecerahan citra dimana masing-masing piksel dioperasikan dengan *template operator* kecerahan. Jika citra tersebut mempunyai piksel sebanyak 1000 piksel (100x100 piksel) maka

terjadi komputasi piksel sebanyak 1000 kali juga. Demikian pula pada operasi citra kelas berat seperti pengolahan citra pada sidik jari (*fingerprint*) memerlukan beberapa rangkaian komputasi. Penelitian [2] memberikan contoh pengolahan citra sidik jari. Hasil penelitian tersebut hanya untuk pra-pengolahan (*pre-processing*) dan meningkatkan kualitas citra (*image enhancement*) sidik jari belum sampai ke tahap identifikasi sidik jari. Dalam penelitian tersebut digunakan lima tahap komputasi: *normalization*, *orientation image estimation*, *frequency image estimation*, *region mask generation* dan *filtering*. Masing-masing tahapan tadi memiliki komputasi lebih dari sekali, seperti tahap *normalization* terdapat tiga kali komputasi untuk menghitung *mean*, *varian* dan *normalisasi*.

Sama halnya dengan penelitian [3,4,5,6,7,8,9,10,11,12,13,14] memerlukan lebih banyak komputasi citra dibandingkan dengan [2]. Oleh karena itu, diperlukan teknik yang dapat mempercepat proses pengolahan citra apalagi untuk pengolahan citra dengan waktu nyata (*realtime*) diperlukan batasan waktu yang sangat ketat (*time bounds*) pada saat proses pengolahan citra seperti deteksi wajah secara langsung menggunakan video. Penerapan *multi-threading* dalam pengolahan citra bukanlah hal yang baru, aplikasi-aplikasi pengolahan citra komersil telah menggunakan teknik ini untuk mempercepat proses komputasi [15] akan tetapi aplikasi-aplikasi tersebut hanya digunakan untuk memanipulasi citra yang bertujuan untuk memperbaiki kualitas citra seperti *enhancement*, *cropping*, *noise removal* dan sebagainya dan juga produk komersil tersebut bersifat *close-source* sehingga masih sedikit pustaka kode program yang tersedia.

Penelitian *multi-threading* dalam bidang pengolahan citra digital telah dilakukan oleh [1,15] sedangkan penelitian *multi-threading* telah dilakukan oleh [16,17] dan terus dalam pengembangan. Penelitian ini membahas bagaimana teknik *multi-threading* diterapkan untuk pengolahan citra digital dengan bahasa pemrograman java. *Multi-threading* menggunakan bahasa pemrograman java juga telah diteliti oleh [18,19].

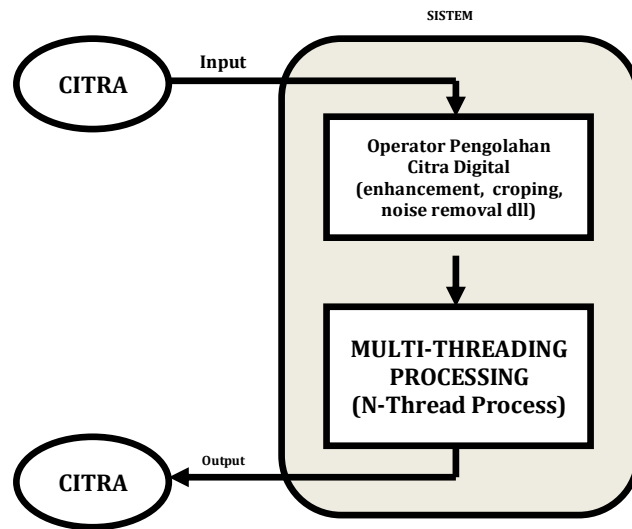
3. Multi-Threading

Multi-threading adalah teknik pemrograman yang memungkinkan beberapa sub proses dalam program dapat berjalan secara paralel. Sebuah *thread* adalah sebuah bagian program yang dapat berjalan mandiri, sehingga dua atau lebih *thread* dapat berjalan bersamaan, tanpa yang satu harus menunggu selesainya yang lain. Akibatnya penggunaan *multi-thread* membuat GUI lebih responsif dan menggunakan *resources* menjadi lebih efektif.

Keuntungan menggunakan *multi-threading* diantaranya adalah responsif, berbagi sumber daya, utilisasi arsitektur *multiprosesor*, dan ekonomis.

4. Arsitektur Sistem Multi-Threading

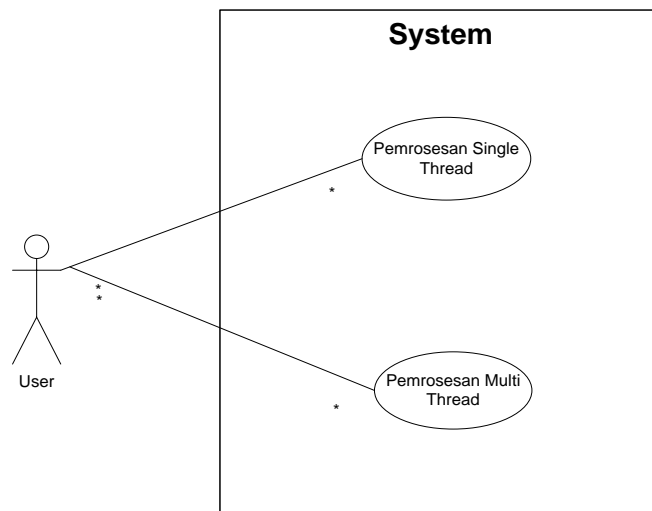
Gambar 1 berikut ini memperlihatkan gambaran dari arsitektur sistem *multi-threading* yang dibangun. Pada gambar tersebut memperlihatkan *input-process-output* dari sistem.



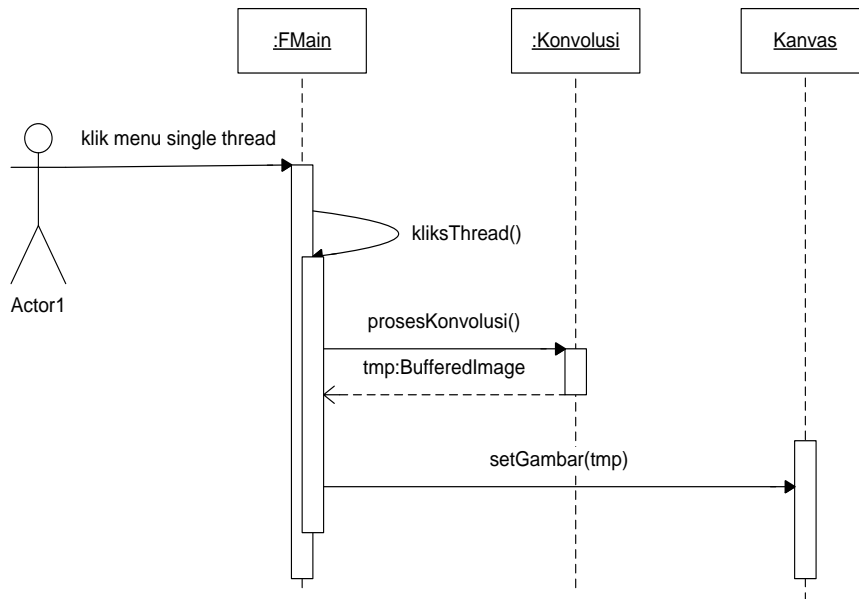
Gambar 1: Arsitektur Sistem

5. Analisis dan Perancangan Perangkat Lunak

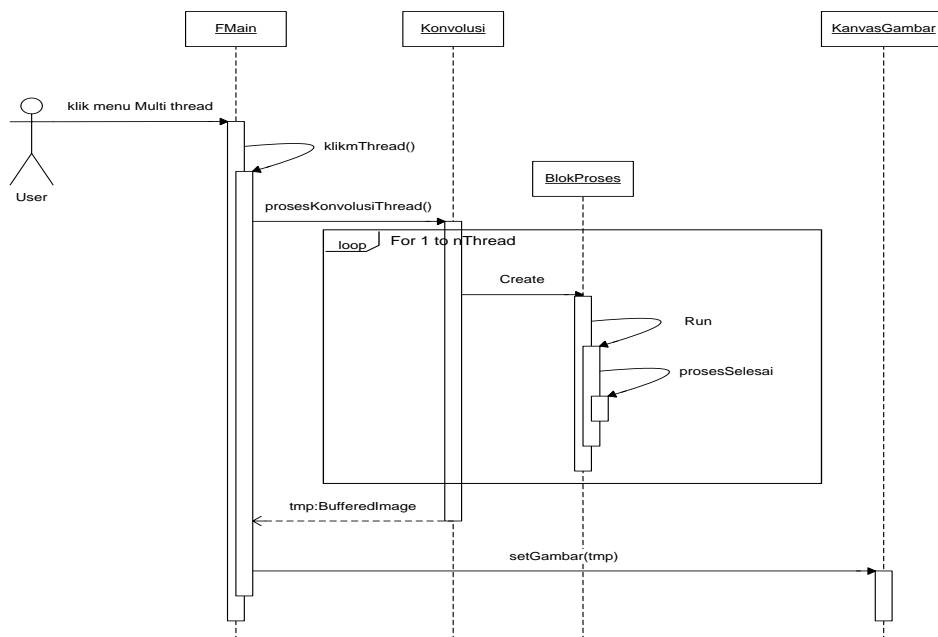
Perangkat lunak yang dikembangkan menggunakan 2 use-case, yaitu pemrosesan single-thread dan multi-thread seperti diperlihatkan pada Gambar 2.



Gambar 2: Diagram use-case perangkat lunak



Gambar 3: Diagram sekuen proses *single thread*



Gambar 4: Diagram sekuen proses *Multi-Thread*

6. Hasil Pengujian Perangkat Lunak

Perangkat lunak yang dihasilkan dalam penelitian ini telah diuji dengan hasil seperti tampak pada Tabel 1 berikut.

Tabel 1: Rata-rata hasil pengukuran waktu proses untuk tiga sampel

Sampel	Waktu penyelesaian proses (mili detik)		
	Single-Thread	Multi-Thread	Selisih
Sampel 1	562	496,8	65,2
Sampel 2	1984	1922,0	62,0
Sampel 3	7772	6906,4	865,6

7. Analisis Hasil Pengujian

Perangkat lunak yang dihasilkan dalam penelitian ini dikembangkan dengan paradigma berorientasi obyek dengan bahasa pemrograman Java dalam IDE NetBeans dengan dua buah fitur utama yaitu pemrosesan secara *single-thread* dan pemrosesan secara *multi-thread* maka dapat dilakukan pengujian untuk tiga buah sampel citra bahkan jika diperlukan jumlah sampel dapat diperbanyak lagi.

Berdasarkan hasil pengujian perangkat lunak untuk rata-rata waktu proses seperti terlihat pada Tabel 1 menunjukkan teknik pemrograman *multi-threading* menghasilkan pemrosesan yang lebih cepat. Berdasarkan selisih waktu pemrosesan secara *single-thread* dengan *multi-thread* menunjukkan bahwa perbedaan waktu keduanya semakin signifikan untuk citra yang berukuran semakin besar. Hal ini disebabkan karena teknik pemrograman *multi-threading* dapat lebih mengoptimalkan kinerja prosesor terutama untuk komputer dengan prosesor ganda.

Pada komputer dengan prosesor tunggal teknik *multi-threading* tidak menghasilkan proses yang benar-benar paralel karena banyak *thread* tetapi dikerjakan oleh satu prosesor dengan cara *time slicing*. Sedangkan untuk komputer dengan prosesor ganda maka banyak *thread* benar-benar dapat didistribusikan pada prosesor-prosesor tersebut sehingga tercipta pemrosesan secara paralel yang sesungguhnya. Hasil pengujian perangkat lunak tersebut bisa berbeda untuk waktu pengujian yang berbeda. Hal ini disebabkan beban prosesor yang berbeda untuk situasi perangkat lunak aplikasi yang dieksekusi yang sedang dieksekusi oleh sistem operasi. Namun demikian hasil pengujian tersebut dapat mewakili perbandingan *single-thread* dan *multi-thread* karena saat pengujian perangkat lunak ini keduanya dijalankan pada situasi yang sama.

8. Kesimpulan

Kesimpulan penelitian yang telah dilakukan adalah sebagai berikut:

- Perangkat lunak yang dikembangkan pada penelitian ini dapat digunakan untuk membandingkan kecepatan pemrosesan citran secara *single-thread* dan *multi-thread*, dan
- Teknik pemrograman *multi-threading* dapat mempercepat proses pengolahan citra terutama untuk citra dengan ukuran besar.

Referensi

- [1] A. K Manjunathachari, K. Satya Prasad, 2006, "*Implementation of Image Processing Operations Using Simultaneous Multithreading and Buffer Processing*", GVIP Journal, Volume 6, Issue 3.
- [2] Lin Hong, Yifei Wan, Anil Jain, 1998, "Fingerprint Image Enhancement: Algorithm and kPerformance Evaluation", IEEE Transaction on Pattern Analysis and Machine Intelligence Vol.20 No. 8.
- [3] Atipat Julasayvake, Somsak Choomchuay, 2007, "*An Algorithm for Fingerprint Core Point Detection*", Signal Processing and Its Applications, 2007. ISSPA 2007, 9th International Symposium on Digital Object Identifier, Page(s): 1 – 4.
- [4] Ching-Tang Hsieh, Shys-RongShyu, and Chia-Shing Hu, 2005, "*An Effective Method of Fingerprint Classification Combined with AFIS*", IFIP – International Federation for Information Processing, pp 1107 – 1122.
- [5] Rein-Lien Hsu, Mohamed Abdel-Mottaleb, Anil K. Jain, 2002, "*Face Detection in Color Images*", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.24 No. 5.
- [6] Qinzi Zhang, Kai Huang, Hong Yan, 2002, "*Fingerprint Classification Based on Extraction and Analysis of Singularities and Pseudoridges*", Research and Practice and Information Technology, Vol. 11.
- [7] Zhi Han, Chang Ping Liu, 2005, "*Fingerprint Classification Based on Statistical Features and Singular Point Information*", IWBRIS 2005, LNCS 3781 pp. 119 – 126.
- [8] Sen Wang, Wei Wei Zhang, Yang Sheng Wang, 2002, "*Fingerprint Classification by Directional Fields*", ICMI '02 (IEEE International Conference on Multimodal Interfaces).
- [9] Wei Wang, Jianwei Li, Weimin Chen, 2006, "*Fingerprint Classification Using Improved Directional Field and Fuzzy Wavelet Neural Network*", Proceedings World Congress on Intelligent Control and Automation.
- [10] Sarat C. Dass, Anil K Jain, 2001, "*Fingerprint Classification Using Orientation Field Flow Curves*" Medical Imaging, IEEE Transactions on Volume: 20, Issue:12, Digital Object Identifier, Page(s): 1215 – 1227.
- [11] Raymond Thai, 2003, "*Fingerprint Image Enhancement and Minutiae Extraction*", The University of Western Australia.
- [12] E. Canto, N.Canyellas, M.Fons, M.Lopez, 2004, "*FPGA Implementation of the Ridge Line Following Fingerprint Algorithm*", FPL 2004, LNCS 3203, pp. 1087 – 1089.
- [13] Liana M.Lorigo, VenuGovindaraju, May.2006, "*Offline arabic handwritten recognition: a survey*", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.28, No.5.
- [14] Chul Hyun Park, Joon Jae Lee, Mark JT Smith, KilHoumPark, 2006, "*Singular point detection by shape analysis of directional fields in fingerprints*", Journal of Pattern Recognition Society, vol.39 pp.839 – 855.
- [15] PawelCzarnul, AndrzejCiereszko, MarcinFraczak, ,"*Towards Eficient Parallel Image Processing on Cluster Grids using GIMP*".

-
- [16] Todd C Mowry, Charles Q.C. Chan, Adley K.W. Lo, ,“*comparative evaluation of latency tolerance techniques for software distributed shared memory*” High-Performance Computer Architecture, 1998. Proceedings, Fourth International Symposium on Digital Object Identifier: 10.1109/HPCA.1998.650569 Publication Year: 1998, Page(s): 300 – 311.
- [17] Wei-Chun Ku, Shu-Hsuan Chou, Jui-Chin Chu, Chi-Lin Liu, Tien-Fu Chen, Jiun-In Guo, Jinn-Shyan Wang, Nov.2009, “*VisoMT: A Collaborative Multithreading Multicore Processor for Multimedia Applications with a Fast Data Switching Mechanism*”, IEEE Transactions on Circuits and Systems for Video Technology, Vol.19, No.11
- [18] Iffat H. Kazi, David J. Lilja, 2000, “*JavaSpMT A Speculative Thread Pipelining Parallelization Model for Java Programs*”, Parallel and Distributed Processing Symposium, 2000. IPDPS 2000. Proceedings 14th International, Digital Object Identifier:10.1109/IPDPS.2000.846035 Publication Year: 2000, Page(s): 559 – 564.
- [19] Wanming Chu, Yamin Li, 2003, “*An Instruction Cache Architecture for Parallel Execution of Java Threads*”, Parallel and Distributed Computing, Applications and Technologies, 2003 (PDCAT 2003), Proceedings of the Fourth International Conference on Publication Year: 2003, Page(s): 180 – 184.