PERBANDINGAN PERFORMA PADA *SINGLE IMAGE DEBLURRING* DENGAN MENERAPKAN *PARALLELISM* DAN *SEQUENTIAL PROGRAMMING*

Muhammad Bagas Nugroho, Muhamad Rizki Triyanto, Fikri Yusrihan, Amalia Rahma Aisyah

Jurusan Ilmu Komputer/ Informatika, Fakultas Sains dan Matematika,
Universitas Diponegoro, Semarang
Email: bagasnugget@gmail.com, muhamadrizkitriyanto41@gmail.com, fikriyusrihan@gmail.com, amaliarahmaaisyah14@gmail.com.

Abstrak

Proses komputasi dalam pemrograman terdiri dari dua macam yakni komputasi paralel dan komputasi sekuensial. Komputasi paralel sendiri telah terbukti dapat meningkatkan performa komputasi dari segi waktu sehingga semakin banyak digunakan pada saat ini. Artikel ini akan membahas tentang perbandingan performa waktu pada program uji *Single Image Deblurring* yang menerapkan dua macam komputasi *parallel programming* dan *sequential programming*. Dari hasil pengujian dapat ditarik kesimpulan bahwa dengan menggunakan komputasi paralel terbukti meningkatkan performa waktu eksekusi. Hal ini dikarenakan dengan menggunakan komputasi paralel, maka proses akan dijalankan secara bersamaan. Sedangkan jika menggunakan komputasi sekuensial, proses akan dijalankan secara berurutan.

Kata kunci: Performa waktu, *Single Image Deblurring*, *Sequential*, *Parallel*

Abstract

The computational process in programming consists of two kinds, namely parallel computing, and sequential computing. Parallel computing itself is already proven to improve computing performance in terms of time so it is widely used at this time. This article will discuss the comparison of time performance on the Single Image Deblurring program that applies two kinds of computations, parallel programming, and sequential programming. From the test results, it can be concluded that using parallel is proven to increase execution time performance. This is because by using parallel computing, the process will be run simultaneously. Meanwhile, if using computational, the process will be executed sequentially.

Keyword: *Time performance, Single Image Deblurring, Sequential, Parallel*

1. Pendahuluan

Di era yang serba digital ini manusia sudah terbiasa menggunakan media komputer untuk menyelesaikan berbagai masalah komputasi, karena penyelesaian yang menggunakan komputer dirasakan lebih cepat dan akurat dibandingkan dengan penyelesaian masalah secara manual. Seiring dengan hal tersebut, tuntutan terhadap kinerja komputasi semakin meningkat.

Komputasi paralel dapat dibangun dari komputer pribadi dengan menggunakan program yang dapat melakukan simulasi suatu lingkup paralel. Pendekatan yang digunakan pada komputasi paralel adalah menggunakan CUDA sebagai alat pemrograman paralel pada GPU untuk mengambil *resources* dari semua *core* yang tersedia. Sedangkan komputasi sequential prosesnya terjadi dalam urutan yang ketat, di mana tidak mungkin ke langkah berikutnya sampai proses yang saat ini selesai.

Artikel ini akan membahas tentang perbandingan performa waktu pada program uji *Single Image Deblurring* yang menerapkan dua macam komputasi *parallel programming* dan *sequential programming*.

Single Image Deblurring bertujuan untuk memperjelas ketajaman gambar dari gambar yang buram atau *blur*. Gambar yang buram bukan hanya tidak menyenangkan secara visual tetapi secara signifikan akan menurunkan kinerja sistem penglihatan.

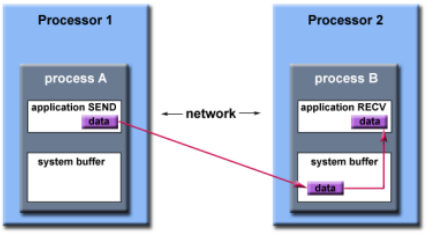
2. Dasar Teori

2.1.Komputasi Paralel

Komputasi paralel adalah metode komputasi untuk menyelesaikan permasalah komputasi dengan membagi beban komputasi ke dalam beberapa bagian kecil sub proses komputasi, dimana sub komputasi tersebut dijalankan pada prosesor yang berbeda secara bersamaan dan saling berinteraksi satu sama lain. Komputasi paralel diperlukan saat kapasitas komputasi yang diperlukan sangat besar, baik karena harus mengolah data maupun karena tuntutan proses komputasi yang banyak. Tujuan dari komputasi paralel adalah meningkatkan kinerja komputer dalam menyelesaikan berbagai masalah dengan membagi sebuah masalah besar ke dalam beberapa masalah kecil, sehingga hal tersebut membuat kinerja menjadi cepat**.**

Komputasi paralel berjalan dengan menggunakan pemrograman Paralel. Pemrograman paralel merupakan teknik pemrograman komputer yang memungkinkan eksekusi perintah/operasi secara simultan (komputasi paralel), baik dalam komputer dengan satu (prosesor tunggal) ataupun banyak (prosesor ganda dengan mesin paralel) CPU. Komunikasi data pada sistem

pemrograman parallel ditunjukkan pada gambar 1 dibawah ini



Path of a message buffered at the receiving process

2.2.Komputasi *Sequential*

Komputasi sekuensial adalah komputasi yang prosesnya terjadi dalam urutan yang ketat, dimana tidak mungkin ke langkah berikutnya sampai proses yang saat ini selesai. Komputasi sekuensial biasanya hanya dijalankan dengan menggunakan sebuah inti/*core* pada CPU.

2.3. CUDA

CUDA singkatan dari *Compute Unified Device Architecture* merupakan arsitektur komputer paralel yang dikembangkan oleh NVIDIA. CUDA memiliki kemampuan untuk melakukan komputasi secara bersamaan yakni pada CPU dan GPU sehingga proses komputasi bisa berjalan lebih cepat. Sebagai contoh proses perhitungan yang dilakukan oleh CPU bisa diselesaikan dalam beberapa detik namun dengan bantuan GPU maka proses bisa menjadi *millisecond*. GPU yang dilengkapi CUDA mampu melakukan simulasi grafis dengan baik daripada hanya menggunakan CPU sehingga GPU yang dilengkapi CUDA mampu melakukan simulasi grafis dengan baik daripada hanya menggunakan CPU.

CUDA merupakan sebuah framework yang bertugas untuk menjembatani proses di CPU dan GPU sehingga keduanya dapat bekerja secara bersamaan (parallel). Seri pertama dari CUDA dikeluarkan oleh NVIDIA pada tahun 2006, yang terus melakukan pengemabnagan pada CUDA sehingga saat ini sudah mencapai versi ke-4. CUDA merupakan sistem properti buatan NVIDIA.

3. Metode Penelitian

3.1. Pemilihan Studi Kasus

Studi kasus yang diamati adalah penerapan *Single Image Deblurring* dengan menerapkan arsitektur *Coarse-to-fine* yang mengubah gambar yang kabur menjadi jelas dan detail. Namun pada *project* ini peneliti telah mengembangkan arsitektur tersebut dengan menggunakan *multi-input multi-output U-net* (MIMO-UNet) sehingga mampu menghasilkan proses *deblurring* yang lebih cepat. Adapun *source code* dari studi kasus ini dapat diakses pada alamat berikut `https://github.com/chosj95/MIMO-UNet`.

3.2. Pemilihan Dataset

Dataset yang digunakan dalam studi kasus ini adalah GOPRO_large dataset yang ditujukan untuk *dynamic*

scene deblurring. Dataset ini tersedia secara publik dan dapat diakses pada `https://seungjunnah.github.io/Datasets/gopro.html`

3.3. Proses Pelatihan

Tahap awal yang harus dilakukan untuk menjalankan *project* ini adalah dengan melakukan proses pelatihan terhadap dataset dengan menggunakan algoritma yang nantinya akan menghasilkan sebuah model. Proses pelatihan ini sendiri memanfaatkan *library machine learning* yaitu PyTorch.

Ketika proses pelatihan, terdapat pilihan untuk menentukan sumber daya komputasi yang kita pilih. Jika kita melakukannya secara serial, maka kita akan memilih sumber daya komputasi berupa CPU. Jika kita memilih untuk melakukannya secara paralel, maka kita akan memilih sumber daya komputasi berupa GPU (CUDA). Untuk mengetahui dan mengamati waktu jalannya eksekusi, diberikan juga sebuah timer pada tiap-tiap perulangan dan epoch yang telah dilakukan.

3.4. Proses Pengujian

Setelah model dihasilkan dari tahap pelatihan, maka saatnya untuk menguji model tersebut untuk mengubah data berupa gambar yang kabur, menjadi gambar yang jelas.

Dalam proses pengujian juga terdapat pilihan sumber daya komputasi yang ingin digunakan. Sehingga kita dapat dengan jelas melihat perbedaan performa ketika menggunakan sekuensial dan paralel.

3.5. Multi-Input Single Encoder

Dalam program ini, Encoder Blocks(EB) menggunakan gambar blur dengan skala yang berbeda sebagai input. Dengan kata lain, selain kita mengambil fitur dari EB diatas, kita juga mengambil fitur dari gambar blur yang di-*downsampled* kemudian menggabungkan dua fitur tersebut.

3.6. Multi Output Single Decoder

Pada MIMO-Unet, *Decoder Block* (DB) yang berbeda memiliki peta fitur dengan ukuran yang berbeda. Fitur-fitur dengan perbedaan skala dapat digunakan untuk meniru model *sub-network* dengan banyak lapisan. Tidak seperti *intermediate supervision* pada *coarse-to-fine network* yang konvensional, proyek ini menggunakan *intermediate supervision* pada setiap *Decoder Block* yang ada.

3.7 Asymmetric Feature Fusion

Asymmetric Feature Fusion (AFF) memungkinkan aliran informasi dari skala yang berbeda dalam satu U-Net. Setiap AFF mengambil output dari semua Encoder Block (EB) sebagai input dan menggabungkan fitur multi-skala menggunakan *convolutional*

layer. Output dari AFF dikirim ke Decoder Block (DB) yang sesuai.

4. Hasil dan Pembahasan

4.1 Implementasi Metodologi

Dalam studi kasus *Single Image Deblurring* menggunakan *multi-input multi-output U-net* (MIMO-UNet) yang menggunakan datasets GOPRO_Large sebesar 8.9 GB.

Proses pelatihan dan pengujian dilakukan dengan menggunakan dua sumber daya yaitu CPU dan GPU. *Hardware* yang digunakan adalah CPU Intel Core i7-9750H dan GPU Nvidia GeForce GTX 1650.

Proses pemilihan metode dilakukan dengan menambahkan beberapa potongan kode sebagai berikut.

```
#Cek apakah CUDA tersedia

device =
torch.device('cuda' if
torch.cuda.is_available()
else 'cpu')
```

Potongan kode di atas menggunakan library PyTorch untuk memilih GPU CUDA jika tersedia dan memilih CPU jika tidak tersedia.

4.2 Pelatihan

Dalam proses pelatihan untuk setiap epoch dengan 1052 iterasi disetiap *epochnya* sebagai berikut didapatkan total waktu yang dibutuhkan sebagai berikut.

Metode	**CPU**	**GPU**
Total waktu*	168.90 s	9.19 s

Keterangan:

* Total waktu untuk 1 epoch.

Proses pelatihan menggunakan GPU CUDA memerlukan 159.71 detik lebih cepat dibandingkan hanya dengan menggunakan CPU.

4.3 Pengujian

Didapatkan rata-rata waktu untuk setiap iterasi pengujian dari data tersebut sebagai berikut.

Metode	**CPU**	**GPU**
PSNR	31.70 dB	31.73 dB
Runtime(s)	17.147 s	0.025 s

Keterangan :

PSNR : *Peak signal-to-noise ratio*

Proses pengujian menggunakan GPU CUDA lebih cepat dan menghasilkan PSNR yang lebih besar dibandingkan dengan hanya menggunakan CPU. Dapat dilihat proses menggunakan CUDA memerlukan waktu 17.122 detik lebih cepat dibandingkan menggunakan CPU.

5. Kesimpulan

Berdasarkan pelatihan dan pengujian yang mengacu terhadap metodologi penelitian maka dapat disimpulkan performa waktu terbaik pada single image deblurring didapatkan dengan menggunakan proses parallelism CUDA yang mana pada proses

pengujian hanya membutuhkan waktu 0.025 detik, hal ini dikarenakan dengan menggunakan parallelism maka proses akan berjalan secara bersamaan antara CPU dan GPU. Proses pelatihan juga menunjukkan bahwa dengan menerapkan CUDA maka waktu yang diperlukan untuk 1 epoch hanya 9.19 detik.

6. Daftar Pustaka

[1] Cho, S. J., Ji, S. W., Hong, J. P., Jung, S. W., & Ko, S. J. (2021). Rethinking coarse-to-fine approach in single image deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 4641-4650).