

BAB 6

KESIMPULAN DAN SARAN

Pembuatan perangkat lunak “Penyelesaian Masalah Knapsack dengan Menggunakan Multi-Processor pada GPU” dilakukan untuk menunjukkan keuntungan yang dapat diperoleh dalam masalah kecepatan dan efisiensi perhitungan dibandingkan dengan penggunaan CPU konvensional. Seperti yang pernah diulas pada Bagian 1.1, CPU tetap dibutuhkan untuk melakukan pekerjaan-pekerjaan yang kompleks. Meskipun begitu, ketika pekerjaan yang ingin dilakukan banyak, tapi sederhana, CPU menjadi kurang efisien karena jumlah inti yang dapat melakukan pekerjaannya sedikit. Banyak faktor juga yang membuat pekerjaan di CPU semakin lambat, seperti sistem operasi, dan pekerjaan lain yang sedang dilakukan. Dengan bantuan GPU, diharapkan dapat mengurangi beban pekerjaan yang diberikan kepada CPU.

6.1 Kesimpulan

Hasil dari penelitian ini secara keseluruhan mencakup:

1. Cara kerja GPU telah berhasil dipelajari untuk melakukan perhitungan *knapsack* dengan bantuan GPU.
2. Analisis *cost* transaksi dan interaksi antara CPU dengan GPU serta dampak transaksi tersebut terhadap kinerja algoritma telah berhasil dilakukan dan dianalisis.
3. Penyelesaian masalah *knapsack* dengan menggunakan algoritma *Dynamic Programming* telah berhasil diimplementasikan dengan bantuan GPU.
4. Perangkat lunak untuk menyelesaikan masalah *knapsack* dengan bantuan GPU untuk meningkatkan kinerja perhitungan dan mengurangi waktu yang dibutuhkan untuk menyelesaikan perhitungan *knapsack* berhasil dibangun dengan optimasi-optimasi yang telah dijelaskan.
5. Kompresi dan optimasi komunikasi antara CPU dengan GPU telah berhasil dilakukan. Interaksi antara CPU dengan GPU juga telah berhasil dikurangi dengan pengelompokan data tabel keputusan perhitungan.
6. Pengujian algoritma telah dilakukan dan berhasil dibuktikan bahwa kinerja perhitungan yang dibantu GPU untuk hampir keseluruhan kasus uji lebih cepat daripada perhitungan yang dilakukan di CPU saja.

6.2 Saran

Agar penelitian ini dapat lebih lanjut dikembangkan dan diperbaiki, berikut adalah saran-saran yang diusulkan:

1. Membuat perangkat lebih mengetahui karakteristik data yang ingin diproses dan menentukan bilamana data tersebut cocok dihitung oleh CPU atau lebih baik diberikan kepada GPU saat *runtime*.

2. Mengimplementasikan algoritma *knapsack* yang dilakukan secara paralel di CPU dan melakukan perbandingan kinerja dengan perhitungan yang dilakukan di GPU secara paralel.
3. Penelitian dapat dikembangkan lebih jauh dengan memperbesar cakupan masukan yang diberikan agar data lebih akurat. Termasuk ke dalam cakupan ini adalah variasi N dan W yang lebih banyak dan bentuk-bentuk data masukan yang berbeda, seperti data yang menurun, menaik, acak, dan campuran.
4. Memberikan optimasi-optimasi khusus terhadap perangkat keras yang digunakan. Dokumentasi NVIDIA menjelaskan karakteristik perangkat-perangkat rakitannya dengan arsitektur-arsitektur yang berbeda. Masing-masing perangkat tersebut memiliki tujuan dan optimasi masing-masing.
5. Menambahkan fitur yang dapat mengakomodasi teknologi NVIDIA SLI®, yakni penggunaan lebih dari satu GPU dalam waktu yang bersamaan. Tentunya teknologi ini membutuhkan GPU yang mendukung SLI® secara *hardware*. Sebagai contoh, perangkat yang digunakan dalam pengujian dokumen ini, yakni Geforce GTX 1060, **tidak** mendukung teknologi ini.

DAFTAR REFERENSI

- [1] NVIDIA (2017) *CUDA C Programming Guide*. NVIDIA, US.
- [2] Wilt, N. (2013) *The CUDA Handbook*. Addison-Wesley, US.
- [3] Skiena, S. S. (2008) *The Algorithm Design Manual*, 2nd edition. Springer Publishing Company, Incorporated.
- [4] Halim, S. dan Halim, F. (2013) *Competitive Programming 3: The New Lower Bound of Programming Contests*, 3rd edition.
- [5] Martello, S. dan Toth, P. (1990) *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Inc., New York, NY, USA.
- [6] Herlihy, M. dan Shavit, N. (2008) *The Art of Multiprocessor Programming*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [7] Boyer, V., El Baz, D., dan Elkihel, M. (2012) Solving knapsack problems on GPU. *Computers and Operations Research*, **39**, 42–47.