PARALELISASI NEEDLEMAN-WUNSCH

Asril Adi Sunarto[1)], Siti Muawanah Robial[2)]

[1]Teknik Informatika, Universitas Muhammadiyah Sukabumi
[2]Teknik Sipil, Universitas Muhammadiyah Sukabumi
Jl.Syamsudin S.H No. 5 Kota Sukabumi
Email: [1]asriladi@ummi.ac.id, [2]smuawanah.robial@gmail.com

Abstrak

Peran ilmu komputer diberbagai domain ilmu lain telah banyak membantu dalam memecahkan masalah komputasi seperti dalam microbiologi dalam melakukan proses penjajaranDNA. Salah satu teknik dalam penjajaran urutan DNA adalah Needleman-Wunsch yang menggunakan dynamic programming. Kompleksitas dynamic programming ini mencapai $O(n^2)$. Untuk mengurangi kompleksitas tersebut, maka salah satunya dengan menggunakan komputer pararel. Penelitian ini berfokus pada paralelisasi Needlemen-Wunsch dengan menggunakan dua komputer yang saling terhubung dan sampel DNA dari GenBank: D85708.1 dan X51404.1. Tahap pengisian matrik dengan komputer pertama mengerjakan matriks segitiga atas, sisanya dengan komputer kedua. Hasilnya nilai speed up menurun drastis hingga 0.3 dan efisiensi mencapai 15 %. Besarnya kompleksitas komunikasi saat pemrosesan menjadi penyebab menurunnya performa hingga 3 kali lipat lebih lambat dari komputer tunggal. Buruknya nilai speed up dan efisiensi tersebut mengindikasikan bahwa untuk mempercepat metode Needlemen-Wunsch sangat tidak mungkin dan keliru memilih komputer pararel untuk menjadi suatu solusi.

Kata Kunci: *bioinformatika, dynamic programming, komputer pararel, needlemen-wunsch, sequence alignment.*

1. Pendahuluan

Pendekatan ilmu komputer diberbagai domain ilmu lain telah banyak membantu dalam menyelesaikan masalah seperti dalam ilmu biologi [1]. Hal itu seperti ilmu mikrobiologi dalam melakukan proses penjajaran urutan suatu DNA. Teknik penjajaran urutan DNA bertujuan untuk mengidentifikasi daerah kemiripan antara dua sekuens DNA untuk menganalisis hubungan fungsional, struktural, atau evolusi antara urutan [2]. Hal ini juga memudahkan dalam pembuatan model dari peristiwa evolusi secara komparatif, dan menghasilkan informasi yang banyak dan beragam, dan berguna dalam membangun suatu pohon kekerabatan (filogenetika) pada investigasi wabah penyakit [3].

Similarity (kemiripan) adalah tingkat kesamaan antara satu *sequence* dengan *sequence* yang lain, sedangkan *homology* adalah menunjukan hubungan evolusi antara dua *sequence* atau lebih [4]. Contoh hasil *sequence alignment* di atas merupakan contoh hanya untuk mendapatkan skor *similarity*, sedangkan *homology* tidak dapat disimpulkan, karena hanya memuat sebagian DNA sajaSecara umum penggunaan sekuen DNA memiliki fungsi aplikatif yaitu untuk "memukul jatuh gen" [5] menunjukan hubungan Filogenetik dalam Genus Holothuria berdasarkan Jujukan Gen 16S Mitokondria rRNA [6], dan konfirmasi sampel-sampel tanaman obat [7]. Baik sebagai fungsi umum maupun khusus, penggunaan sekuen DNA memerlukan suatu proses penjajaran sekuen.

Terdapat jumlah data yang sangat besar ketika proses penjajaran sekuen DNA. Isu-isu yang kritis pada proses tersebut adalah akurasi dan kecepatan, yang mana keduanya saling berseberangan. Memilih metode yang lebih cepat akan mengurangi akurasi data, dan sebaliknya [8]. Beragam metode penjajaran sekuen DNA menghasilkan algoritma yang cepat-kurang akurat dan ada lambat-akurat. Maka dari itu perlu suatu metode penjajaran yang akurat.

Salah satu metode dalam penjajaran sekuen DNA yang menitik beratkan pada akurasi data dengan mengesampingkan waktu proses adalah metode Needleman-Wunsch. Metode ini mendefinisikan cara menemukan urutan global terbaik dari dua urutan menggunakan dynamic *programming* (DP) [9]. Karakteristik yang menghitung secara rekursifmenjadi faktor tingginya Komplesitas DP yang mencapai $O(n^2)$ [10].

Untuk mereduksi kompleksitas suatu algoritma memerlukan suatu cara seperti penggunaan komputer pararel. Hal ini seperti suatu pekerjaan dikerjakan oleh beberapa pekerja.Sehingga dengan konsep ini, kompleksitas komputasi dapat dikurangi bahkan secara dramatis. Dimasa depan, setiap node komputasi akan memiliki lebih dari satu perangkat percepatan, hal ini menambah komputer pararel ketingkat level baru [11].

Perlu suatu cara agar penggunaan komputer pararel secara objektif lebih unggul dibandingkan dengan menggunakan satu komputer. Cara tersebut adalah dengan menghitung *speed up (S)* dan efisiensi (E) yang ada dari kinerja komputer paralel. *Speed up (S)* merupakan perbandingan antara waktu eksekusi yang dimiliki oleh satu komputer dengan waktu ekseksusi yang dimiliki oleh banyak komputer. Sedangkan effisiensi (E) merupakan nilai efektivitas dari program paralel yang menggunakan sejumlah *p* processor terhadap program serial.

Terdapat dua tipe arsitektur yang bisa dibangun dalam komputer paralel, yaitu *share memory* yang mana setiap processor dapat mengakses *memory module* sehingga setiap program dapat mengakes data dan

distributed memory yang menggunakan sejumlah komputer yang terhubung dengan jaringan komputer [12]. Untuk menilai kualitas program pararel diperlukan pengukuran kinerjanya. Pengukuran kinerja pemrograman paralel dapat diukur dari seberapa banyak peningkatan kecepatan (speed up) dan efisiensinya.

Meski begitu, desain suatu algoritma sangat dipengaruhi oleh struktur aplikasi yang hasilnya tersebut menyoroti keefektifan algoritma dan penerapan paralelnya. Dengan kata lain, algoritma paralel bukan jaminan dalam penerapannya menghasilkan efektivitas yang tinggi, tergantung dari struktur algoritma yang dibangun baik menggunakan paralel maupun tidak. Oleh karena itu, perlu membandingkan antara program dengan satu komputer dengan lebih dari satu yang dapat membuktikan seberapa besar performanya pada metode Needlemen-Wunsch sehingga layak digunakan.

2. Metode

NW yang mengimplementasikan DP menggunakan matrik (M) dengan beberapa tahapan, yaitu (1) Inisiasi matrik, (2) Pengisian matrik, dan (3) Runut balik. Inisiasi matrik berturut-turut diisi oleh nilai parameter *gap*, *match*, dan *mismatch*. Matrik pada kolom dan baris ke 0 berturut-turut menggunakan Formula (1).

$$\begin{matrix} M\ j,0 & = & gap\ x\ \ posisi\ karakter\ sekuen\ B \\ M\ 0,i & = & gap\ \ x\ posisi\ karakter\ sekuen\ A \end{matrix} \quad (1)$$

Sedangkan pengisian matrik menggunakan Formula NW yang dapat dilihat pada Formula (2).

$$M\ i,j\ = max \begin{matrix} M\ i-1,j-1\ + sub\ A\ i\ ,B\ j\ \ ; \\ M\ i-1,j\ + del\ A\ i\ \ ; \\ M\ i,j-1\ + ins\ B\ j \end{matrix} \quad (2)$$

Sub(A[i],B[j]) merupakan fungsi yang mana jika karakter pada posisi sekuen *A[i]* dengan sekuen *B[j]* itu sama maka *M[i-1,j-1] + match*, jika tidak maka *M[i-1,j-1] + mismatch*. Nilai *del(A[i])* dan *ins(B[j])* merupakan nilai parameter *gap*. Dan terakhir pada tahap runut balik proses dimulai dari posisi akhir *M[i,j]* yang menelususi angka-angka tertinggi menuju ketitik awal M[0,0]. Untuk menghitung similarity dapat menggunakan Formula (3).

$$Similarity = \frac{Jumlahmatchx\ 100\ \%}{jumlahpanjangsequence} \quad (3)$$

Secara konsep, metode paralelisasi NW yang menggunakan multiprocessor adalah sama dengan metode NW yang menggunakan processor tunggal. Paralelisasi NW membagi pekerjaan yang independen dari pengisian matrik. Berdasarkan tahapan pengisian matrik pada Formula (2), bagian yang independen itu ketika memproses berdasarkan baris atau kolom. Untuk mengukur kinerja program pararel perlu formula untuk menghitungnya. *Speed up (S)* dan efisiensi (E) merupakan komponen pengukuran kinerja program pararel yang dapat dilihat berturut-turut pada Formula (4) dan (5).

$$S = \frac{\text{Single Processor Computing } (Ts)}{\text{Multi Processor Computing } (Tp)} \quad (4)$$

$$Tp = communicationtime\ + computationtime$$

$$E = \frac{S}{p} x100\% \quad (5)$$

2.1. Perangkat dan Bahan

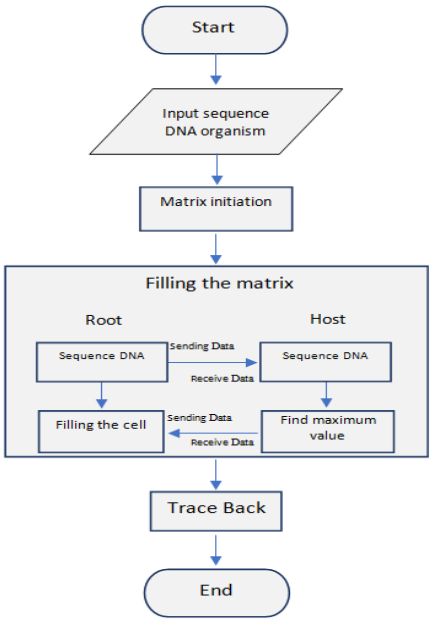
a. Perangkat lunak dipergunakan antara lain:
1) Codeblock 10
2) Microsoft Message Passing Interface (MPI)
3) Sistem Operasi Windows 7

b. Perangkat keras dipergunakan antara lain:
1) Processor Intel i7 2.4 GHz dan i5 230 GHz
2) Ram 4 Gb
3) Hardisk 320 Gb

Bahan dipergunakan adalah sepasang data *sequence* diambil dari NCBI (*https://www.ncbi.nlm.nih.gov*) yaitu *sequence* dari organisme Chlamydia caviae (Genbank: D85708.1) dan Chlamydophila caviae (Genbank: X51404.1).

2.2. Metode Penelitian

Metode penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Metode Penelitian Paralelsiasi NW

Setelah dua sekuen dimasukan, maka *root* mengatur dan menghitung guna membagi data.Hasil desain ini, matrik segitiga bawah itu diproses oleh *host*, sedangkan sisanya oleh *root*.Setelah itu tahapan yang berhubungan dalam pengiriman data, seperti komunikasi data, aglomerasi, dan *mapping*. Pengiriman data dalam pemograman paralel ini menggunakan komunikasi *point to point* dengan parameter *MPI_Send()* disisi pengirim dan parameter *MPI_Recv()* disisi penerima.

3. Hasil dan Pembahasan

3.1. Pemrosesan Tunggal

Metode NW yang menggunakan DP menghasilkan kompleksitas $O(n^2)$ ini diperlihatkan ketika pengisian matrik. Lebih lanjut kaitannya dengan Formula (2) dengan pengisian matrik adalah seperti misalnya terdapat karakter DNA Sekuen 1 "ATGGCT" dan karakter DNA Sekuen 2 "CGTGAA" yang akan di sejajarkan. Maka tahap pertama inisiasi matrik dengan member parameter match = 7, mismatch = -3, dan gap = -2. Hasilnya seperti Gambar 2.

Posisi		0	1	2	3	4	5	6
			A	T	G	G	C	T
0		0	-2	-4	-6	-8	-10	-12
1	C	-2						
2	G	-4						
3	T	-6						
4	G	-8						
5	A	-10						
6	A	-12						

Gambar 2. Tahap inisiasi matrik

Terlihat bahwa posisi dari baris pertama (*M[0,0], M[0,1], M[0,2],...M[0,i]*) dan kolom pertama (*M[1,0], M[2,0], M[3,0],...M[j,0]*) menggunakan Formula (2) di atas. Kemudian tahap selanjutnya adalah pengisian matrik yang menggunakan Formula (3) yang dimulai dari posisi *M[1,1]* hingga posisi *M[i,j]* dalam contoh di atas adalah posisi M[6,6].

Perhitungan untuk mengisi posisi *M[1,1], M[1,2], M[1,3], M[1,4], M[1,5], M[1,6]* berturut-turut adalah:

$$\begin{array}{lll} 0+-3; & -2+-3; & -4+-3; \\ \max\ -2+-2;, \max & -3+-2, \max & -5+-2; \\ -2+-2 & -4+-2 & -6+-2 \\ -6+-3; & -8+7; & -10+-3; \\ \max\ -7+-2;, \max & -9+-2, \max & -1+-2; \\ -8+-2 & -10+-2 & -12+-2 \end{array}$$

dan seterusnya.

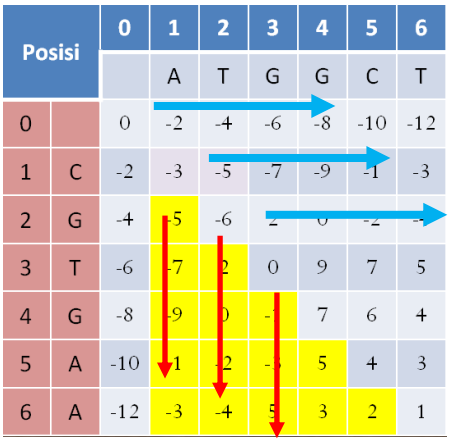
Hasilnya dapat dilihat pada Gambar 3.

Posisi		0	1	2	3	4	5	6
			A	T	G	G	C	T
0		0	-2	-4	-6	-8	-10	-12
1	C	-2	-3	-5	-7	-9	-1	-3
2	G	-4	-5	-6	2	0	-2	-4
3	T	-6	-7	2	0	9	7	5
4	G	-8	-9	0	-1	7	6	4
5	A	-10	-1	-2	-3	5	4	3
6	A	-12	-3	-4	5	3	2	1

Gambar 3. Tahap pengisian matrik

3.2. Pemrosesan Pararel

Pertama dalam komputasi pararel adalah cara pempartisian dengan melihat setiap tugas satu sama lainnya dalam kondisi independen. Isu dalam memproses pengisian matrik posisi M[i,j] adalah posisi sebelumnya seperti M[i-1,j-1], M[i-1,j], M[i,j-1] menjadi syarat mutlah harus selesai terlebih dahulu (i dan j >= 0). Hal ini membuat pembagian tugas pengisian matrik menjadi perbaris.Yang mana setiap baris atau kolomnya memungkinkan untuk diproses oleh processor yang berbeda.Seperti contoh desain yang membag berdasarkan pemrosesan kolom dapat dilihat pada Gambar 4.



Gambar 4. Desain partisi pararel (warna kuning diproses oleh processor yang berbeda)

Pada Gambar 4 di atas, pemrosesan pengisian matrik dibagi menjadi dua bagian, yang mana pemrosesan secara horizontal dilakukan oleh *root process*, sedangkan secara vertical dilakukan oleh host process. Konsekuensinya biaya komunikasi meningkat tajam. Biaya komunikasi yang diperlihatkan oleh Gambar 4 di atas merupakan:

a. pengiriman Sekuen A (*n*=panjang sekuen A) dan Sekuen B (*m* = panjang sekuen B) dari *root* ke *host*, dengan panjang sekuen disamakan dengan Sekuen yang lebih pendek. Kompleksitasnya Komunikasi ini mencapai $\frac{m(m+1)}{2}$. Kompleksitas ini menjadi tiga kali karena biaya komunikasi dari *host* ke *root* sama dengan *root* ke *hostdan* kompleksitas komputasi disisi *host*.
b. Kompleksitas komputasi pada root menjadi $\frac{n(n+1)}{2} + m(\mathrm{n}-\mathrm{m})$.
c. Sehingga total kompleksitas pararel menjadi $\frac{m^2+n^2+3m+2mn}{2}$

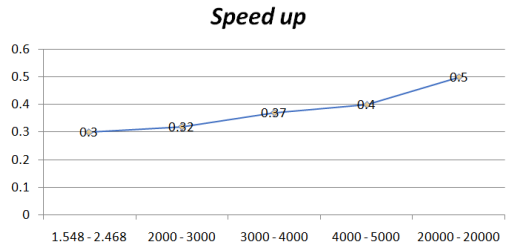
Hasil perhitungan kompleksitas pararel di atas dapat digunakan dalam membandingkan kompleksitas antara menggunakan *processor* tunggal (n^2) dan *multiprocessor* ($\frac{m^2+n^2+3m+2mn}{2}$) dapat dilihat pada Tabel 1.

Table 1. Perbandingan kompleksitas antara penggunaan satu processor dengan dua processor

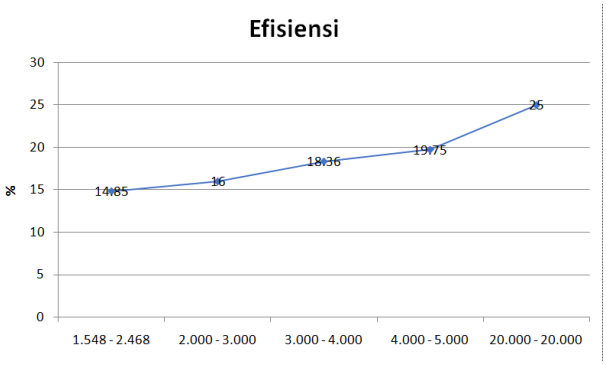
m	n	Complexity		Speed up	Efficiency
		Tunggal	Multiprocessor		
1.548	2.468	2.396.304	8.066.450	0.30	14.85
2.000	3.000	4.000.000	12.503.000	0.32	16.00
3.000	4.000	9.000.000	24.504.500	0.37	18.36
4.000	5.000	16.000.000	40.506.000	0.40	19.75
20.000	20.000	400.000.000	800.030.000	0.50	25.00

n = number of sequence A m= number of sequence B

Berdasarkan Tabel 1 di atas, grafik*speed up* dan efisiensi dapat dilihat pada Gambar 5 dan Gambar 6.



Gambar 5. Grafik *Speed up*



Gambar 6. Grafik Efisiensi

Sedangkan yang dijalankan dengan menggunakan program satu *PC* dan *pararel*, didapatkan seperti pada Tabel 2.

Tabel 2. Hasil eksekusi program pararel dan program serial

Chlamydia caviae (Genbank:D85708.1)	1548
Chlamydophila caviae (Genbank: X51404.1)	2468
Execution time single processor	0.078 s
Execution time two processors	0.25 s
Speedup	0.312
Efficiency	15.6%

4. Kesimpulan

Berdasarkan hasil pengujian kinerja program pararel mendapatkan nilai speed up sebesar 0.3 dan efisiensi sebesar 15.6%. Nilai ini mendekati hasil pengujian kompleksitas yang telah dikemukakan pada Tabel 1 di atas. Besaran *speed up* yang hanya mencapai 0.3 mengindikasikan bahwa paralelisasi NW jauh lebih lambat tiga kali lipatnya dari program yang menggunakan satu processor, sehingga langkah ini tidaklah feasibel untuk digunakan. Sebagai saran, maka perlu suatu upaya penelitian lanjutan untuk mengurangi kompleksitas yang dimiliki oleh komputer pararel.

Ucapan Terimakasih

Terima kasih sebesar-besarnya kepada Kementerian Riset dan Pendidikan Tinggi Republik Indonesa yang telahmendukung pendanaan penelitian ini, LPPM Universitas Muhammadiyah Sukabumi, dan rekan-rekanfakultasyangtelahmembantumenyelesaikan penelitian ini hingga selesai.

Daftar Pustaka

[1] Yunita Irma, Kristian Tjandradiredja, Seng Hansun. "Perkembangan Bioinformatics dalam Ruang Lingkup Ilmu Komputer". ULTIMATICS, Vol. VIII, No. 1 | Juni 2016, hal 65-69. 2016.

[2] Singh V K, Singh A K, Chand R., and Kushwaha C. "Role of Bioinformatics in Agriculture and Sustainable Development". Banaras Hindu University, India. 2011.

[3] Wibawa Hendra, Walujo Budi Prijono, Ni Luh Putu Indi Dharmayanti, Sri Handayani Irianingsih , Yuli Miswati, Anieka Rohmah, Ernes Andesyha, Romlah, Rosmalina Sari Dewi Daulay, dan Kiki Safitria. "Investigasi Wabah Penyakit Pada Itik Di Jawa Tengah, Yogyakarta, Dan Jawa Timur : Identifikasi Sebuah Clade Baru Virus Avian Influenza Subtipe H5n1 Di Indonesia". BULETIN Laboratorium Veteriner Vol : 12 No : 4 Tahun 2012, hal 2-9. 2012.

[4] Baxevanis AD, Ouellette BFF. "Bioinformatics A Practical Guide to the Analyses of gene and Proteins". A John Wiley & Sons, Inc., Publication. 2001.

[5] Supatmi. "Bioteknologi Crispr/Cas9: Cara terbaru untuk "Memukul Jatuh Gen". Bio*Trends* Vol.7 No.2 Tahun 2016, hal 31-36. 2016.

[6] Kamarudin Kamarul Rahim, Rehan, Hashim, Gires Usup dan Maryam Mohamed Rehan. "Phylogenetic Relationships within the Genus Holothuria Inferred from 16S Mitochondiral rRNA Gene Sequences". Sains Malaysiana 45(7)(2016): hal 1079–1087. 2016.

[7] Xue, C.Y. dan Li, D.Z. "Use of dna barcode sensu lato to identify traditional Tibetan medicinal plant *Gentianopsis paludosa* (Gentianaceae)". *J. Sys. Evol*, 49 (3): 267-270. 2011.

[8] Sunarto Asril Adi, W. A. Kusuma and H. Sukoco, "Parallelization of star alignment". *3rd International Conference on Instrumentation, Communications, Information Technology and Biomedical Engineering (ICICI-BME)*, Bandung, 2013, pp. 167-171. doi: 10.1109/ICICI-BME.2013.6698486. 2013.

[9] Needleman, Saul B. and Christian D Wunsch. "A general method applicable to the search for similarities in the amino acid sequence of two proteins". J. Mol. Biol. 48, 443-453. 1970.

[10] SandAndreas, Morten K. Holt,Jens Johansen, Rolf Fagerberg, Gerth Stølting Brodal, Christian N. S. Pedersenand Thomas Mailund. "Algorithms for Computing the Triplet and Quartet Distances for Binary General Trees". *Biology* 2013, *2*(4), 1189-1209; doi:10.3390/biology2041189. 2013.

[11] Sourouri Mohammed, Tor Gillberg, Scott B. Baden, and Xing Cai. "Effective Multi-GPU Communication Using Multiple CUDA Streams and Threads". Conference: 20th International Conference on Parallel and Distributed Systems (ICPADS 2014), DOI: 10.1109/PADSW.2014.7097919. 2015.

[12] Wilkinson Barry and Michael Allen. "Parallel Programming (Techniques and Applications Using a Network of Workstations and Parallel Computers)". Andi Yogyakarta. 2010.