

IMPLEMENTASI DAN ANALISIS BIT-PARALLELISM UNTUK MENGEFISIENKAN KOMPUTASI DYNAMIC PROGRAMMING

Iqbal Fauzi¹, Setyorini², Andrian Rakhmatsyah³

^{1,2,3} Universitas Telkom, Bandung

iqbalfau@telkomuniversity.ac.id¹, setyorini@telkomuniversity.ac.id², kangandrian@telkomuniversity.ac.id³

Abstrak

Terdapat kebutuhan analisis biologi dengan melakukan pencarian similaritas pada sequence biologi (RNA, DNA, protein). Pencarian similaritas dalam permasalahan ilmu komputer adalah permasalahan string matching. String matching pada kebutuhan analisis biologi bisa dilakukan dengan membandingkan pasangan string atau sequence biologi (pairwise sequence alignment) atau membandingkan kelompok sequence biologi (multiple sequence alignment). Algoritma utama dalam pairwise alignment ataupun multiple sequence alignment adalah Dynamic programming (DP).

Bit-parallelism atau Bit-Vector merupakan salah satu algoritma yang dikembangkan dari algoritma DP, yang mengadaptasi cara kerja pemrosesan word di dalam sistem komputer. Word merupakan unit yang terdiri dari kumpulan data biner. Bit-parallelism melakukan transformasi unit komputasi pada DP matrix menjadi unit-word.

Dalam penelitian ini telah diimplementasikan algoritma Bit-parallelism dan dibandingkan dengan algoritma DP Matrix dan diteliti efek efisiensi bit-parallelism jika diterapkan pada suatu lingkungan komputasi tertentu.

Hasil dari penelitian ini adalah bit-parallelism memiliki waktu yang lebih cepat dibandingkan dengan DP Matrix dan mendapatkan efisiensi 98% lebih baik.

Kata Kunci : Pencocokan kata, Bit-Parallelism, Dynamic Programming Matrix.

Abstract

There is a need for biological analysis by searching for similarities in biological sequences (RNA, DNA, protein). The search for similarity in a computer science problem is the string-matching problem. String matching for biological analysis needs can be done by comparing pairs of strings or biological sequences (pairwise sequence alignment) or comparing groups of biological sequences (multiple sequence alignment). The main algorithm in pairwise alignment or multiple sequence alignment is Dynamic Programming (DP).

Bit-parallelism or Bit-Vector is one of the algorithms developed from the DP algorithm, which adapts the way words work in computer systems. Word is a unit consisting of a binary data set. Bit parallelism transforms the computational units in the dp matrix into word units.

In this research, the Bit-parallelism algorithm is implemented and compared with the DP Matrix algorithm and research has been carried out on the efficiency effect of bit-parallelism when applied to a certain computing environment.

The result of this research is bit-parallelism has a faster time than DP Matrix and gets 98% better efficiency.

Keywords: String Matching, Bit-Parallelism, Dynamic Programming Matrix.

1. Pendahuluan

Terdapat kebutuhan analisis biologi dengan melakukan pencarian similaritas pada sequence biologi seperti RNA (Ribonucleic Acid), DNA (deoxyribonucleic acid), dan protein. Pencarian similaritas dalam permasalahan ilmu komputer adalah permasalahan string matching. String matching pada kebutuhan analisis biologi bisa dilakukan dengan membandingkan pasangan string atau sequence biologi (pairwise sequence alignment) atau membandingkan kelompok sequence biologi (multiple sequence alignment). Algoritma utama dalam pairwise alignment ataupun multiple sequence alignment adalah Dynamic programming[1].

Dynamic programming merupakan salah satu algoritma yang bisa menangani berbagai masalah salah satunya masalah pencocokan kata dengan menggunakan Dynamic Programming Matrix (DP Matrix) dengan kompleksitas $O(mn)$ [2]. Dynamic programming melakukan komputasi perbandingan sequence secara exhaustive, karena semua kemungkinan solusi ditelusuri sehingga pertumbuhan ukuran sequence maupun jumlah sequence akan berpengaruh secara signifikan pada waktu komputasinya.

Bit-parallelism atau Bit-Vector merupakan salah satu algoritma yang dikembangkan dari algoritma DP, yang mengadaptasi cara kerja pemrosesan word di dalam sistem komputer. Word merupakan unit yang terdiri dari kumpulan data biner. Bit-parallelism melakukan transformasi unit komputasi pada DP matrix menjadi unit-unit word. Sehingga kompleksitas komputasi DP matrix yang semula $O(mn)$ dimana m dan n ukuran sequence dapat diefisienkan menjadi $O((mn)/w)$. Dengan menggunakan Bit-parallelism tingkat kompleksitas mengecil dan waktunya menjadi lebih singkat[3]. Pada penelitian ini diimplementasikan algoritma bit-parallelism yang tingkat kompleksitasnya menjadi $O((mn)/w)$.

Rumusan Permasalahan yang dibahas pada penelitian ini adalah Bagaimana bit-parallelism

memberikan efek efisiensi jika diterapkan pada suatu lingkungan komputasi yang berbeda.

Tujuan dari penelitian ini adalah mengimplementasikan algoritma Bit-parallelism dan membandingkannya dengan algoritma DP Matrix dan meneliti efek efisiensi bit-parallelism jika diterapkan pada suatu lingkungan komputasi tertentu.

Batasan masalah pada penelitian ini adalah Spesifikasi Komputer: Processor Intel Core i3-9100f. CPU @ 3.60GHz (4 CPUs), ~3,6GHz. Memory 8192 RAM. Operating System: Windows 10 Pro 64-bit. Menggunakan input single word berukuran 8-bit dan 64-bit string untuk kedua jenis algoritma.

2. Studi Terkait

Pada penelitian Myer [3] disebutkan bahwa bit vector dapat mendapatkan hasil yang sama Dengan Dynamic Programming matrix tapi dengan kompleksitas lebih kecil yaitu $O(mn/w)$.

3. Metode Penelitian

Dynamic Programming ialah metode untuk menentukan kecocokan yang optimal dengan mencocokkan dua string untuk semua kemungkinan pasangan karakter antara kedua string[4]. Dynamic programming bekerja dengan membuat sebuah matriks dua dimensi terlebih dahulu. Dimana sumbu adalah kedua alignment yang dibandingkan. Lalu kolom pertama akan diinisialisasi dengan angka 0, 1, 2, ..., n atau sampai dengan panjang string kedua. Lalu baris pertama akan diisi dengan nol. Matriks akan diisi dengan rumus berikut:

Jika yang di bandingkan sama maka:

$$C[i,j]=C[i-1,j-1] \quad (1)$$

Jika yang dibandingkan berbeda maka:

$$C[i,j]=\min\{C[i-1,j-1]+1,C[i,j-1]+1,C[i,j-1]+1\} \quad (2)$$

Score dihitung satu per satu dalam satu baris lalu ke baris selanjutnya sampai semua baris terisi. Score terbaik berada di pojok kanan bawah.

Bit-parallelism merupakan salah satu algoritma lanjutan dari Dynamic Programming yang dikembangkan oleh Myers. Dimana dalam algoritma ini kompleksitasnya menjadi $O(mn/w)$ dikarenakan pemrosesannya dilakukan pada level

bit[3]. Meskipun kompleksitasnya lebih kecil tapi scorenya akan sama dengan score yang dihasilkan dari DP Matrix.

Dalam Bit-parallelism terdapat dua tahap algoritma yaitu pre-processing dan algoritma utama seperti berikut:

```

1 Preprocessing
2 for c ∈ Σ do B [c] = 0m od
3 for j ∈ Σ 1 ... m do B [pj] = B [pj] | 0m-j
10j-1 od
    
```

Gambar 1 Algoritma pre-processing

```

1 Scanning the text
2 for pos ∈ 1...n do
3     X = B[tpos] | VN;
4     D0 = ((VP + (X&VP)) ^ VP | X;
5     HN = VP&D0;
6     HP = X&D0;
7     X = HP << 1;
8     VN = X&D0;
9     VP = (HN << 1) | ~(X|D0)
10    if HP&10m-1 ≠ 0m
11        then score +=1;
12    else if HN&10m-1 ≠ 0m
13        then score -=1;
14    fi
15 fi
16 if score ≤ k report occurrence
at pos fi;
17 od
    
```

4. Hasil dan Pembahasan

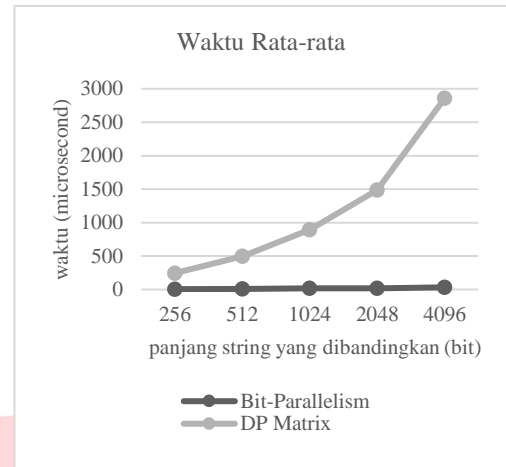
Dibuat data string random dengan Panjang

Gambar 1 Algoritma Bit-Parallelism

word 64bit dengan jumlah 100data dibandingkan 256-bit, 512-bit, 1024-bit, 2048-bit, dan 4096-bit. Dengan percobaan menggunakan dua jenis metode yaitu DP Matrix dan Bit-Parallelism. Dan mendapatkan hasil waktu sebagai berikut:

Tabel 1 Waktu rata-rata

	Bit-Parallelism (ms)	DP Matrix (ms)
256	5.537778	239.252222
512	6.647778	490.710000
1024	16.620000	877.351111
2048	15.514444	1468.620000
4096	32.13888889	2824.89333



Gambar 3 Grafik waktu rata-rata

dapat dilihat bahwa DP matrix menghasilkan waktu rata-rata yang terus meningkat tajam sesuai dengan panjang string yang diinputkan sedangkan pada Bit-Parallelism waktu tetap mengalami peningkatan tapi tidak se ekstrim DP matrix.

Dari hasil sebelumnya dihitung lah tingkat efisiensi waktunya dengan rumus:

$$\text{tingkat efisiensi} = \frac{\text{waktu dp} - \text{waktu bitpar}}{\text{waktu dp}} \times 100 \% \tag{3}$$

Didapatkanlah hasil berikut:

Tabel 2 Hasil Efisiensi

	Bit-Parallelism	DP Matrix	efisiensi
256	5.537778	239.252222	98%
512	6.647778	490.710000	99%
1024	16.620000	877.351111	98%
2048	15.514444	1468.620000	99%
4096	32.13888889	2824.89333	99%

5. Kesimpulan

Pada penelitian ini telah dilakukan percobaan untuk meminimalkan waktu yang dibutuhkan untuk menyelesaikan masalah pencocokan string. dihasilkan bahwa bit-parallelism memiliki waktu yang lebih cepat dibandingkan dengan DP Matrix dan mendapatkan efisiensi 98% lebih baik.

Referensi

[1] Setyorini, Kuspriyanto, D. H. Widyantoro, and A. Pancoro, "The implementation of bit-parallelism for

- DNA sequence alignment,” *J. Phys. Conf. Ser.*, vol. 835, no. 1, 2017, doi: 10.1088/1742-6596/835/1/012004.
- [2] J. Gregor and M. G. Thomason, “Dynamic Programming Alignment of Sequences Representing Cyclic Patterns,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 2, pp. 129–135, 1993, doi: 10.1109/34.192484.
- [3] G. Myers, “A fast bit-vector algorithm for approximate string matching based on dynamic programming,” *J. ACM*, vol. 46, no. 3, pp. 395–415, 1999, doi: 10.1145/316542.316550.
- [4] J. Xiong, *Essential bioinformatics*, vol. 148. 2006.

