

Pencocokan DNA dengan Pencocokan String (Algoritma *Needleman-Wunsch*)

Jovian Christianto - 13514101
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jln. Ganesha 10 Bandung 40132, Indonesia
13514101@std.stei.itb.ac.id

Abstract—Makalah ini berisi aplikasi pencocokan string dalam bidang keilmuan lain. Secara khusus, makalah ini membahas mengenai pencocokan 2 DNA. Kecocokan 2 DNA tersebut mempunyai suatu nilai kecocokan. Dari nilai kecocokan tersebut bisa didapatkan bagaimana kombinasi 2 DNA tersebut agar nilai kecocokannya maksimal. Pencocokan 2 DNA ini menggunakan suatu algoritma, yaitu algoritma *Needleman-Wunsch*.

Keywords—*algoritma, Needleman-Wunsch, pencocokan, string, DNA, aplikasi*

I. PENDAHULUAN

Zaman semakin berkembang, ilmu pengetahuan semakin berkembang. Dalam bidang Biologi, banyak ditemukan makhluk hidup baru yang belum pernah ditemukan sebelumnya. Para ilmuwan meneliti makhluk hidup baru tersebut dan akan dikelompokkan ke dalam kelompok-kelompok yang mempunyai ciri-ciri mirip dengan anggota kelompok tersebut. Selain mempelajari ciri-ciri dan perilaku makhluk hidup baru tersebut, ilmuwan juga mulai mencocokkan DNA makhluk hidup baru tersebut dengan DNA makhluk hidup lain. Jika kecocokannya besar, maka kemungkinan makhluk hidup tersebut satu kelompok dengan makhluk hidup yang diuji semakin besar.

Pencocokan DNA bisa dilakukan dengan berbagai macam cara. Salah satunya adalah dengan mencocokkan susunan kimia DNA. Biasanya susunan kimia DNA yang dicocokkan adalah bagian basa nitrogennya yang terdiri dari Guanin, Adenin, Timin, dan Cytosin. Basa nitrogen tersebut biasa disingkat dengan G, A, T, dan C. Dengan singkatan seperti itu, basa nitrogen dalam susunan kimia DNA dapat dianggap sebagai sebuah string yang berisi basa-basa nitrogen tersebut. Oleh sebab itu pencocokan DNA dapat dilakukan dengan metode pencocokan string. Pencocokan DNA dengan metode pencocokan string yang akan dibahas lebih mendalam dalam makalah ini adalah dengan menggunakan algoritma *Needleman-Wunsch*.

II. DASAR TEORI

A. Pencocokan String

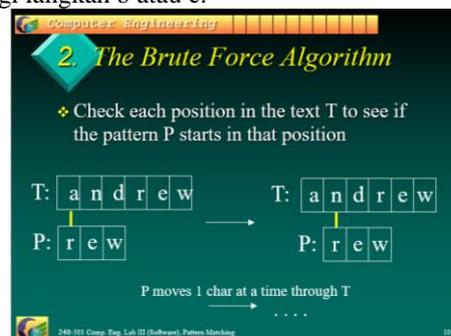
Algoritma pencocokan string merupakan salah satu algoritma yang sering digunakan dalam dunia keilmuan, baik di bidang informatika ataupun di bidang lainnya. Aplikasi

pencocokan string yang paling dasar adalah mengenai pencarian kata di dalam *editor text*. Aplikasi lainnya adalah pencarian kata dalam web (*web search engine*), analisis citra, *bioinformatics*, dll. Pada umumnya pencocokan string menggunakan 3 algoritma, yaitu algoritma *Brute-Force*, algoritma *Knuth-Morris-Pratt*, dan algoritma *Boyer-Moore*. Ketiga algoritma tersebut digunakan untuk mencocokkan suatu string, apakah ada string yang cocok atau tidak. Biasanya dalam ketiga algoritma ini ada *text* yang merupakan string panjang dan *pattern* yang merupakan string pendek. Dalam ketiga algoritma ini dicocokkan apakah ada *pattern* dalam *text*. Ada 1 lagi algoritma yang dapat menentukan kecocokan suatu string berdasarkan suatu nilai, yaitu algoritma *Needleman-Wunsch*. Untuk lebih jelasnya, algoritma-algoritma tersebut akan dijelaskan secara singkat.

1. Algoritma *Brute-Force*

Pencocokan string dengan algoritma *Brute-Force* dilakukan dengan mencocokkan karakter *text* dengan karakter *pattern* satu per satu dari karakter paling kiri (depan). Pencocokan dilakukan dengan langkah-langkah sebagai berikut.

- Cocokkan karakter pertama *pattern* dengan *text*
- Jika sama, maju ke karakter berikutnya. Jika semua karakter *pattern* telah cocok dengan karakter pada *text*, maka *pattern* ditemukan. Proses selesai.
- Jika tidak sama, maka pencocokan karakter pada *pattern* maju 1 karakter pada *text*. Pencocokan karakter dilakukan lagi dari karakter paling kiri (depan) *pattern*.
- Ulangi langkah b atau c.

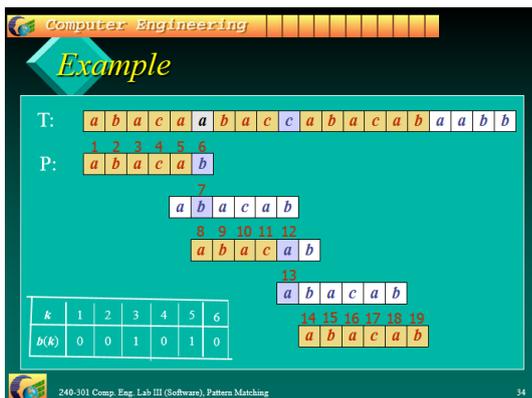


Gambar 1. Contoh algoritma *Brute-Force*

2. Algoritma *Knuth-Morris-Pratt (KMP)*

Pencocokan string dengan algoritma *KMP* dilakukan dengan mencocokkan karakter *text* dengan karakter *pattern* dari karakter paling kiri (depan), sama seperti algoritma *Brute Force*. Pada algoritma *KMP* ada fungsi pinggiran setiap karakter pada *pattern*. Fungsi pinggiran merupakan jumlah karakter terpanjang yang didapatkan jika *prefix* dan *suffix* dari *pattern* sama. Pencocokan dilakukan dengan langkah-langkah sebagai berikut.

- Cocokkan karakter pertama *pattern* dengan *text*
- Jika sama, maju ke karakter berikutnya. Jika semua karakter *pattern* telah cocok dengan karakter pada *text*, maka *pattern* ditemukan. Proses selesai.
- Jika tidak sama, maka ambil fungsi pinggiran karakter sebelum karakter *pattern* yang dibandingkan. Lalu fungsi pinggiran ditambah 1. Itulah posisi karakter *pattern* yang selanjutnya dibandingkan dengan karakter *text*.
- Ulangi langkah b atau c



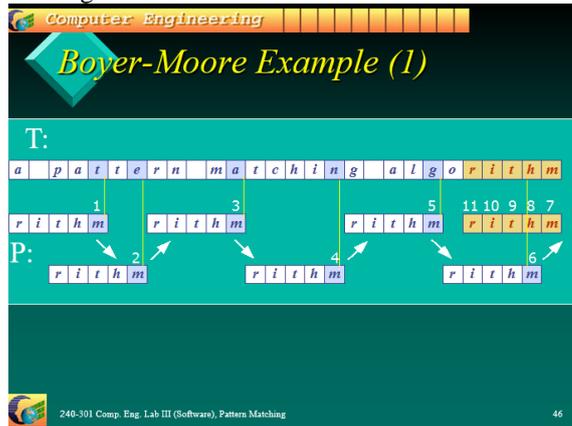
Gambar 2. Contoh algoritma *KMP*

3. Algoritma *Boyer-Moore*

Pencocokan string dengan algoritma *BM* dilakukan dengan mencocokkan karakter *pattern* dengan karakter *text* dari paling kanan (belakang). Di dalam algoritma *BM* ini terdapat 3 kasus khusus. Pencocokan dilakukan dengan langkah-langkah sebagai berikut.

- Cocokkan karakter terakhir *pattern* dengan *text*
- Jika sama, mundur ke karakter sebelumnya. Jika semua karakter *pattern* telah cocok dengan karakter *text*, maka *pattern* ditemukan. Proses selesai.
- Jika tidak sama, maka ada 3 kasus khusus :
 - Jika karakter yang dibandingkan pada *text* ada pada *pattern* dan letaknya di sebelah kiri karakter yang sedang dibandingkan pada *pattern*, maka *pattern* maju sampai karakter tersebut
 - Jika karakter yang dibandingkan pada *text* ada pada *pattern* dan letaknya di sebelah kanan karakter yang sedang dibandingkan pada *pattern*, maka *pattern* maju 1.
 - Jika karakter yang dibandingkan pada *text* tidak ada pada *pattern*, maka *pattern* maju sejauh panjang *pattern*.

- Perbandingan kembali dilakukan dari belakang, ulangi langkah b atau c.



Gambar 3. Contoh algoritma *BM*

4. Algoritma *Needleman-Wunsch*

Pencocokan string dengan algoritma *NW* berbeda dengan algoritma-algoritma sebelumnya. Pada algoritma *NW* ini kecocokan string mempunyai suatu nilai. Biasanya algoritma *NW* ini digunakan dalam bidang *bioinformatics*, dalam pencocokan DNA suatu makhluk hidup. Pada algoritma *NW* ini, nilai kecocokan suatu string ditentukan sendiri. Ada 3 nilai, yaitu nilai jika cocok, jika tidak cocok, dan jika ada gap (jarak). Nilai-nilai tersebut ditentukan sendiri nilainya. Pencocokan dilakukan dengan langkah-langkah sebagai berikut.

- Tentukan nilai cocok, tidak cocok, dan gap
- Buat matriks yang berisikan 2 string yang akan dicocokkan, dan berisikan gap pada setiap awal string.
 - Isikan nilai pada kolom gap dengan nilai gap dikalikan indeks kolom atau baris. Lalu isikan elemen-elemen lainnya dengan nilai maksimum dari 3 kemungkinan nilai. 3 kemungkinan nilai tersebut adalah sebagai berikut.
 - Nilai dari elemen $i-1, j-1$ (diagonal) ditambah dengan nilai asli elemen tersebut, yaitu nilai pencocokannya, cocok atau tidak cocok.
 - Nilai dari elemen $i-1, j$ ditambah dengan nilai gap
 - Nilai dari elemen $i, j-1$ ditambah dengan nilai gap
 - Lakukan langkah a sampai semua matriks terisi.
- Nilai pada elemen paling kanan bawah merupakan nilai dari kecocokan string
- Memperoleh string yang nilai kecocokan yang paling baik (nomor 3). Untuk memperoleh string tersebut, caranya adalah dengan menelusuri kembali nilai-nilai elemen tersebut didapatkan dari mana. Dimulai dari elemen paling kanan bawah, lalu ditelusuri balik sampai pada elemen paling kiri atas. Hasil penelusuran tidaklah unik, hasilnya bisa lebih dari satu. Hal ini disebabkan karena nilai suatu elemen matriks bisa didapatkan dari 2 atau 3 kemungkinan nilai yang menghasilkan nilai yang sama. Dalam memperoleh

string tersebut, terdapat 3 kemungkinan kasus sebagai berikut.

- i dimulai dari panjang string Y dan j dimulai dari panjang string X.
- Jika didapat dari elemen diagonal, maka string X ditambah karakter X_j dan string Y ditambah karakter Y_i . i dan j berkurang 1.
- Jika didapat dari elemen sebelah atas, maka string X ditambah karakter '-' dan string Y ditambah karakter Y_i . i berkurang 1.
- Jika didapat dari elemen sebelah kiri, maka string X ditambah karakter X_j dan string Y ditambah karakter '-'. j berkurang 1.

Contoh :

DNA 1 : AAGT

DNA 2 : AT

Langkah 1.

Nilai cocok : 1

Nilai tidak cocok : -1

Nilai gap : 0

Langkah 2.

Matriks :

		A	A	G	T
	0	0	0	0	0
A	0	1	1	1	1
T	0	1	1	1	2

Pengisian matriks menggunakan langkah-langkah yang telah dijelaskan sebelumnya.

Misal untuk elemen (2,4), dilihat dahulu kemungkinan nilai-nilainya :

- Dari elemen (2,3) : $(2,3) + \text{gap} = 1 + 0 = 1$
- Dari elemen (1,3) : $(1,2) + \text{perbandingan elemen} = 0 + (-1) = -1$
- Dari elemen (1,4) : $(1,4) + \text{gap} = 0 + 0 = 0$

Diambil nilai yang paling besar dari ketiga kemungkinan tersebut, maka diambil nilai 1.

Langkah 3.

		A	A	G	T
	0	0	0	0	0
A	0	1	1	1	1
T	0	1	1	1	2

Nilai kecocokan adalah 2.

Langkah 4.

Elemen matriks yang diwarnai merupakan hasil telusuran asal nilai-nilai elemen yang bermula dari nilai kecocokan di kanan bawah matriks.

Kemungkinan 1 :

		A	A	G	T
	0	0	0	0	0
A	0	1	1	1	1
T	0	1	1	1	2

DNA 1 : AAGT

DNA 2 : A - - T

Kemungkinan 2 :

		A	A	G	T
	0	0	0	0	0
A	0	1	1	1	1
T	0	1	1	1	2

DNA 1 : AAGT

DNA 2 : - A - T

B. DNA

DNA merupakan asam nukleat yang menyimpan segala informasi biologis yang unik dari setiap makhluk hidup. DNA merupakan singkatan dari *deoxyribonucleic acid* atau dalam Bahasa Indonesia disebut dengan Asam Deoksiribosa Nukleat (AND). Susunan kimia DNA adalah polimer berupa rantai panjang dari nukleotida. Nukleotida sendiri terdiri dari 3 komponen dasar, yaitu deoksiribosa, gugus fosfat, dan basa nitrogen.

Satu-satunya pembeda tiap nukleotidan adalah basa nitrogennya. Basa nitrogen terdiri dari kelompok Purin dan Pirimidin. Purin berupa Guanin dan Adenin, sementara Pirimidin berupa Timin dan Cytosin. Secara umum, basa nitrogen terdiri dari Guanin, Adenin, Timin, dan Cytosin. Untuk mempermudah penulisan, biasanya basa-basa tersebut disingkat menjadi G untuk Guanin, A untuk Adenin, T untuk Timin, dan C untuk Cytosin. Variasi urutan dari keempat basa-basa tersebut membentuk kode genetik pada sel. Variasi tersebut yang membedakan makhluk hidup satu dengan makhluk hidup lainnya.

III. ANALISIS PENCOCOKAN DNA

Misalkan dalam bidang kedokteran terdapat sebuah masalah untuk mencocokkan 2 DNA makhluk hidup. Kedua DNA ini akan diteliti apakah berasal dari nenek moyang yang sama atau tidak. Para peneliti ingin membandingkan kecocokan kedua DNA ini untuk mengetahui jawaban dari masalah tersebut. Berikut susunan basa nitrogen dari suatu makhluk hidup yang akan diteliti.

DNA-A : GAATTCAGTTA

DNA-B : GGATCGA

Untuk menentukan nilai kecocokan matriks ini, digunakan algoritma *Needleman-Wunsch* seperti telah dijelaskan di bagian sebelumnya. Secara umum, pencocokan DNA dengan menggunakan algoritma *Needleman-Wunsch* adalah sebagai berikut.

1. Menentukan nilai cocok, tidak cocok, dan gap
2. Membuat matriks
3. Menentukan nilai kecocokan
4. Membuat kemungkinan pencocokan DNA paling baik

1. Menentukan nilai cocok, tidak cocok, dan gap

Nilai cocok = 1
 Nilai tidak cocok = -1
 Nilai gap = 0

2. Membuat matriks

Pseudocode yang menggambarkan pengisian matriks adalah sebagai berikut.

```
//PENGISIAN BARIS DAN KOLOM GAP
for i = 0 to length(DNA_1)
    matriks(i,0) <- gap * i
for j = 0 to length(DNA_2)
    matriks(0,j) <- gap * j

//PENGISIAN BARIS DAN KOLOM LAIN
for i = 1 to length(DNA_1)
    for j = 1 to length(DNA_2)
        nilai1 <- matriks(i-1,j-1) + nilai(DNA_1i,
DNA_2j)
        nilai2 <- matriks(i-1,j) + gap
        nilai3 <- matriks(i,j-1) + gap
        matriks(i,j) <- max(nilai1, nilai2, nilai3)
```

Dengan menggunakan algoritma demikian, maka matriks hasilnya adalah sebagai berikut.

		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0	0	0	0	0
G	0	1	1	1	1	1	1	1	1	1	1	1
G	0	1	1	1	1	1	1	1	1	1	1	1
A	0	1	2	2	2	2	2	2	2	2	2	2
T	0	1	2	2	3	3	3	3	3	3	3	3
C	0	1	2	2	3	3	4	4	4	4	4	4
G	0	1	2	2	3	3	4	4	5	5	5	5
A	0	1	2	3	3	3	4	5	5	5	5	6

Contoh cara pengisian matriks :

1. matriks(3,2)
 - nilai 1 = matriks(2,1) + nilai(A,A) = 1 + 1 = 2
 - nilai 2 = matriks(2,2) + gap = 1 + 0 = 1
 - nilai3 = matriks(3,1) + gap = 1 + 0 = 1
 - max(2,1,1) = 2
2. matriks(5,4)

$$\begin{aligned} \text{nilai1} &= \text{matriks}(4,3) + \text{nilai}(C,T) = 2 + (-1) = 1 \\ \text{nilai 2} &= \text{matriks}(4,4) + \text{gap} = 3 + 0 = 3 \\ \text{nilai3} &= \text{matriks}(5,3) + \text{gap} = 2 + 0 = 2 \\ \text{max}(1,3,2) &= 3 \end{aligned}$$

3. Menentukan nilai kecocokan

		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0	0	0	0	0
G	0	1	1	1	1	1	1	1	1	1	1	1
G	0	1	1	1	1	1	1	1	1	1	1	1
A	0	1	2	2	2	2	2	2	2	2	2	2
T	0	1	2	2	3	3	3	3	3	3	3	3
C	0	1	2	2	3	3	4	4	4	4	4	4
G	0	1	2	2	3	3	4	4	5	5	5	5
A	0	1	2	3	3	3	4	5	5	5	5	6

Dari matriks yang telah dibuat sebelumnya, didapatkan nilai kecocokan adalah 6.

4. Membuat kemungkinan pencocokan DNA paling baik

Pseudocode yang menggambarkan cara mencari kemungkinan pencocokan DNA paling baik

```
i <- length(DNA_1)
j <- length(DNA_2)
while (i > 0 or j > 0)
    if (i > 0 and j > 0 and matriks(i,j) == matriks(i-1,j-1) + nilai(DNA_1i,DNA_2j)) then
        hasil1 <- DNA_1i + hasil1
        hasil2 <- DNA_2j + hasil2
        i <- i -1
        j <- j -1
    else if (i > 0 and matriks(i,j) == matriks(i-1,j) + gap) then
        hasil1 <- DNA_1i + hasil1
        hasil2 <- "-" + hasil2
        i <- i -1
    else
        hasil1 <- "-" + hasil1
        hasil2 <- DNA_2j + hasil2
        j <- j -1
```

Dengan menggunakan algoritma demikian, didapatkan bahwa DNA dengan nilai pencocokan terbaik adalah sebagai berikut.

Kemungkinan 1.

		G	A	A	T	T	C	A	G	T	T	A	
	0	0	0	0	0	0	0	0	0	0	0	0	
G	0	1	1	1	1	1	1	1	1	1	1	1	
G	0	1	1	1	1	1	1	1	1	1	1	1	
A	0	1	2	2	2	2	2	2	2	2	2	2	
T	0	1	2	2	3	3	3	3	3	3	3	3	
C	0	1	2	2	3	3	4	4	4	4	4	4	
G	0	1	2	2	3	3	4	4	5	5	5	5	
A	0	1	2	3	3	3	4	5	5	5	5	5	6

Hasil telusuran :

hasil1 = - GAATTCAGTTA
 hasil2 = GGA - - TC - G - - A

hasil1	-	G	A	A	T	T	C	A	G	T	T	A	TOTAL
hasil2	G	G	A	-	-	T	C	-	G	-	-	A	
nilai	0	1	1	0	0	1	1	0	1	0	0	1	6

Kemungkinan 2.

		G	A	A	T	T	C	A	G	T	T	A	
	0	0	0	0	0	0	0	0	0	0	0	0	
G	0	1	1	1	1	1	1	1	1	1	1	1	
G	0	1	1	1	1	1	1	1	1	1	1	1	
A	0	1	2	2	2	2	2	2	2	2	2	2	
T	0	1	2	2	3	3	3	3	3	3	3	3	
C	0	1	2	2	3	3	4	4	4	4	4	4	
G	0	1	2	2	3	3	4	4	5	5	5	5	
A	0	1	2	3	3	3	4	5	5	5	5	5	6

Hasil telusuran :

hasil1 = G - AATTCAGTTA
 hasil2 = GGA - - TC - G - - A

hasil1	G	-	A	A	T	T	C	A	G	T	T	A	TOTAL
hasil2	G	G	A	-	-	T	C	-	G	-	-	A	
nilai	1	0	1	0	0	1	1	0	1	0	0	1	6

Kemungkinan 3.

		G	A	A	T	T	C	A	G	T	T	A	
	0	0	0	0	0	0	0	0	0	0	0	0	
G	0	1	1	1	1	1	1	1	1	1	1	1	
G	0	1	1	1	1	1	1	1	1	1	1	1	
A	0	1	2	2	2	2	2	2	2	2	2	2	
T	0	1	2	2	3	3	3	3	3	3	3	3	
C	0	1	2	2	3	3	4	4	4	4	4	4	
G	0	1	2	2	3	3	4	4	5	5	5	5	
A	0	1	2	3	3	3	4	5	5	5	5	5	6

Hasil telusuran :

hasil1 = G - AATTCAGTTA
 hasil2 = GG - A - TC - G - - A

hasil1	G	-	A	A	T	T	C	A	G	T	T	A	TOTAL
hasil2	G	G	-	A	-	T	C	-	G	-	-	A	
nilai	1	0	0	1	0	1	1	0	1	0	0	1	6

Kemungkinan 4.

		G	A	A	T	T	C	A	G	T	T	A	
	0	0	0	0	0	0	0	0	0	0	0	0	
G	0	1	1	1	1	1	1	1	1	1	1	1	
G	0	1	1	1	1	1	1	1	1	1	1	1	
A	0	1	2	2	2	2	2	2	2	2	2	2	
T	0	1	2	2	3	3	3	3	3	3	3	3	
C	0	1	2	2	3	3	4	4	4	4	4	4	
G	0	1	2	2	3	3	4	4	5	5	5	5	
A	0	1	2	3	3	3	4	5	5	5	5	5	6

Hasil telusuran :

hasil1 = GA - ATTCAGTTA

hasil2 = G - GA - TC - G - - A

hasil1	G	A	-	A	T	T	C	A	G	T	T	A	T O T A L
hasil2	G	-	G	A	-	T	C	-	G	-	-	A	
nilai	1	0	0	1	0	1	1	0	1	0	0	1	6

Nilai kecocokan yang didapat bergantung dari nilai yang ditetapkan di awal, yaitu nilai cocok, nilai tidak cocok, dan nilai gap. Jika dipilih nilai cocok yang terbesar, maka semakin besar nilai kecocokan, semakin cocok kedua DNA yang diteliti. Sebaliknya jika dipilih nilai tidak cocok yang terbesar, maka semakin besar nilai kecocokan, semakin tidak cocok kedua DNA yang diteliti.

Dari keempat kemungkinan tersebut didapatkan bahwa nilai kecocokannya adalah 6. Nilai 6 adalah nilai maksimum untuk kedua DNA makhluk hidup yang diteliti. Nilai 6 mempunyai arti bahwa kedua DNA makhluk hidup yang diteliti cukup cocok. Hal ini karena nilai cocok ditetapkan menjadi nilai terbesar dibandingkan dengan nilai lainnya. Dari data-data yang telah didapatkan sebelumnya, nilai 6 didapatkan dari 6 basa nitrogen DNA yang cocok, dan sisanya merupakan gap. Hal ini wajar karena kecocokan akan memberikan nilai 1, gap memberikan nilai 0, dan ketidakcocokan memberikan nilai -1. Algoritma *Needleman-Wunsch* akan menghindari penggunaan nilai dengan nilai terendah. Jika nilai-nilai tersebut diubah, maka nilai kecocokan akan berubah, dan hasilnya pun mungkin berubah. Jadi algoritma *Needleman-Wunsch* ini bersifat fleksibel, dapat disesuaikan dengan kebutuhan pengguna

IV. KESIMPULAN

Pencocokan string tidak hanya dapat diaplikasikan pada pencarian text, melainkan dapat diaplikasikan juga pada pencocokan DNA. Analisis pencocokan DNA makhluk hidup pada bagian sebelumnya hanyalah permasalahan. Basa nitrogen dari makhluk hidup cukup panjang sehingga tidak dapat dijelaskan semua pada makalah ini. Langkah-langkah pengerjaannya seperti pada bagian-bagian sebelumnya, hanya saja ukuran matriks dan pengerjaan pengisian matriksnya cukup panjang. Dari makalah ini juga didapatkan bahwa ada algoritma lain dalam pencocokan string selain algoritma *Brute-Force*, *Knutt-Morris-Prath*, dan *Boyer-Moore*, yaitu algoritma *Needleman-Wunsch* yang menghasilkan nilai kecocokan string yang dibandingkan berdasarkan nilai yang ditentukan sendiri.

V. UCAPAN TERIMA KASIH

Ucapan terima kasih saya ucapkan kepada Tuhan Yang Maha Esa karena rahmat-Nya saya dapat menyelesaikan tugas makalah ini dengan baik. Selanjutnya saya juga mengucapkan terima kasih kepada Ibu Ulfa dan Bapak Rinaldi atas bimbingan dan ajarannya selama 1 semester ini sehingga saya dapat membuat makalah ini dan mengetahui aplikasinya dalam bidang keilmuan lain. Tidak lupa saya juga mengucapkan terima kasih kepada orang tua saya yang telah mendukung saya menyelesaikan makalah ini. Saya juga mengucapkan terima kasih kepada teman-teman dan orang-orang yang tidak bisa saya sebutkan satu per satu yang telah membantu saya menyelesaikan makalah ini dengan baik.

VI. REFERENSI

- [1] Munir, Rinaldi. 2009. *Strategi Algoritma*. Bandung : Institut Teknologi Bandung.
- [2] Munir, Rinaldi. 2015. *Slide Kuliah IF2211 Strategi Algoritma : Pencocokan String (2015)*. Bandung : Institut Teknologi Bandung.
- [3] Needleman, Saul B.; and Wunsch, Christian D. 1970. "A general method applicable to the search for similarities in the amino acid sequence of two proteins". *Journal of Molecular Biology* 48 (3): 443-53K. Elissa, "Title of paper if known," unpublished.
- [4] <http://www.avatar.se/molbioinfo2001/dynprog/dynamic.html>, diakses tanggal 06 Mei 2016
- [5] <http://vlab.amrita.edu/?sub=3&brch=274&sim=1431&cnt=1>, diakses tanggal 06 Mei 2016
- [6] <http://www.slideshare.net/avrilcoghlan/the-needleman-wunsch-algorithm>, diakses tanggal 06 Mei 2016
- [7] <https://www.youtube.com/watch?v=vqxc2EfPWdk>, diakses tanggal 07 Mei 2016
- [8] https://www.youtube.com/watch?v=zwA-6_1bLgE, diakses tanggal 07 Mei 2016
- [9] <http://informasitips.com/dna-pengertian-struktur-dan-fungsinya>, diakses tanggal 07 Mei 2016

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 08 Mei 2016



Jovian Christianto - 13514101