Modifikasi *String* dan *Pattern* untuk Mempercepat Pencocokan Rantai Asam Amino pada Rantai DNA

Septu Jamasoka - 13509080
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13509080@std.stei.itb.ac.id

***Abstract*—Saat ini DNA banyak dipergunakan oleh kedokteran. Salah satu kegunaan dari DNA adalah untuk mencari apakah seseorang mengalami kelainan berdasarkan kandungan rantai asam amino (protein) tertentu yang terdapat pada DNA seseorang yang merupakan rantai asam amino yang menyebabkan kelainan pada seseorang.**

Dengan adanya komputer, pencarian terhadap rantai asam amino pada rantai DNA dapat dipercepat terutama dengan adanya algoritma pencocokan string. Akan tetapi, masalah pencarian rantai asam amino pada DNA dapat dipercepat dengan mengkompresi rantai DNA dan rantai asam amino sehingga waktu pencarian dapat dipercepat lagi.

Kompresi dapat dilakukan dengan memanfaatkan sifat istimewa rantai asam amino dan rantai DNA, yaitu bahwa asam amino merupakan triplet basa sehingga DNA dapat dibagi menjadi triplet-triplet basa dan ukuran DNA akan berkurang dan dengan menggunakan algoritma *brute force* dapat mempercepat kinerja algoritma *brute force*.

***Index Terms*—DNA, asam amino, algoritma pencocokan triplet basa, algoritma *brute force*.**

I. PENDAHULUAN

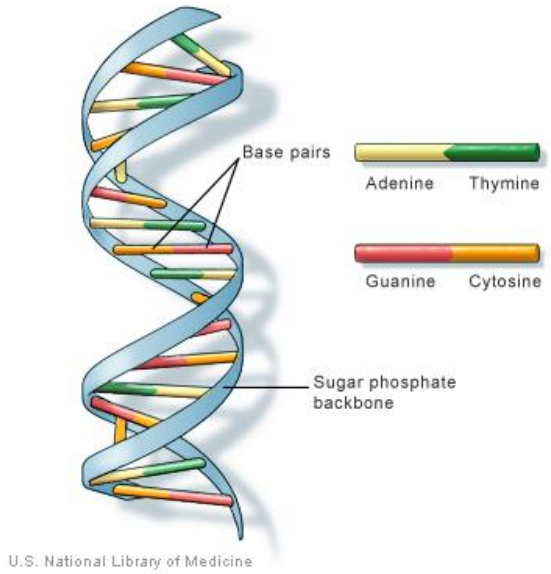
Saat ini dunia kedokteran semakin berkembang, terutama ketika ditemukannya DNA dari seseorang yang bertindak layaknya identitas seseorang. Dalam DNA, terkandung sejumlah banyak data-data yang dapat diekstraksi dan akan menunjukkan ciri-ciri dari seseorang.

Dengan perkembangan teknologi yang semakin canggih, DNA semakin mudah diekstraksi dari dalam diri manusia, terutama dapat berasal dari kulit-kulit yang terkelupas, atau bisa saja dari rambut manusia. Akan tetapi, karena panjangnya rantai DNA yang ada pada manusia, pencarian terhadap rantai protein tertentu akan menjadi cukup sulit.

Saat ini, pencarian terhadap suatu rantai asam amino pada DNA sering dilakukan untuk mengecek apakah seseorang memiliki kelainan karena terdapatnya rantai protein yang menunjukkan adanya kelainan pada protein yang dihasilkan. Selain itu, rantai protein juga dapat digunakan untuk membandingkan seseorang dengan orang lainnya (dalam hal ini digunakan untuk mengecek apakah seseorang dengan orang lain punya orang tua-anak atau tidak).

Dengan berkembangnya teknologi komputer saat ini, pencarian semakin dipermudah dengan adanya komputer. Dengan mekanisme pencarian yang ada, kita dapat dengan mudah mencari suatu protein pada DNA. Akan tetapi, pencarian tersebut ternyata masih bisa dipercepat dengan sifat khusus pada DNA. Salah satunya bahwa DNA terdiri atas rantai asam nukleid yang hanya terdiri atas empat macam asam, dan pencarian terhadap protein pada DNA biasanya dilakukan dalam paket tiga asam.

Dengan adanya keistemewaan itu, bisa diterapkan sedikit algoritma kompresi sehingga panjang untaian dan pola yang dicari dapat dikurangi dan tentunya akan mengurangi lamanya pencarian apabila menggunakan algoritma *brute force* dalam melakukan pencarian pola protein yang ingin dicari pada potongan DNA.



Gambar 1. Contoh potongan rantai DNA[2]

II. DASAR TEORI

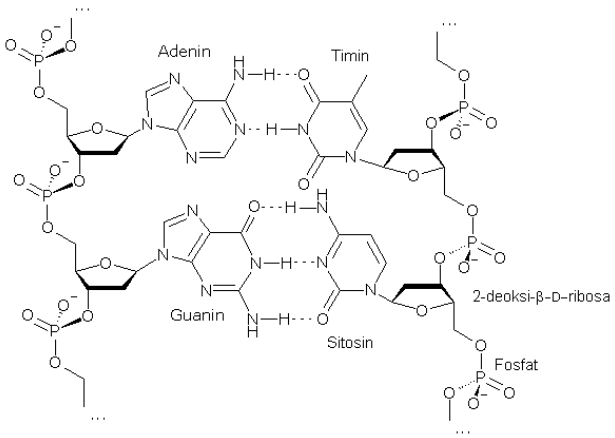
A. Deoxyribonucleic Acid

Deoxyribonucleic Acid atau asam deoksiribonukleat atau lebih dikenal sebagai DNA adalah sejenis asam

nukleat yang tergolong biomolekul utama penyusun berat kering setiap organisme. Di dalam sel, DNA umumnya terletak di dalam inti sel, tetapi dapat juga ditemukan pada mitokondria (dalam hal ini disebut sebagai mtDNA). Secara garis besar, peran DNA di dalam sebuah sel adalah sebagai materi genetik; artinya, DNA menyimpan cetak biru bagi segala aktivitas sel. Ini berlaku umum bagi setiap organisme. Di antara perkecualian yang menonjol adalah beberapa jenis virus (dan virus tidak termasuk organisme) seperti HIV (*Human Immunodeficiency Virus*).

Struktur untai komplementer DNA menunjukkan pasangan basa (adenin dengan timin dan guanin dengan sitosin) yang membentuk DNA beruntai ganda. DNA merupakan polimer yang terdiri dari tiga komponen utama,

1) gugus fosfat
2) gula deoksiribosa
3) basa nitrogen, yang terdiri dari:
    a. Purin
        i. Adenin (A)
        ii. Guanin (G)
    b. Pirimidin
        i. Sitosin (C)
        ii. Timin (T)



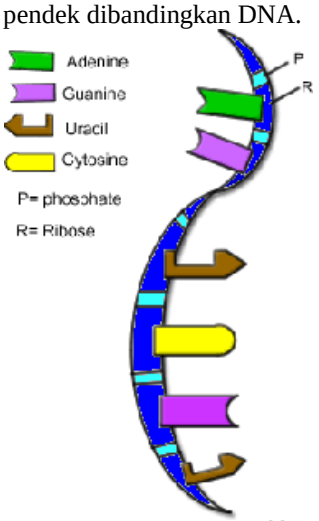
Gambar 2. Struktur dasar DNA[4]

B. Ribonucleic acid

Ribonucleic Acid atau asam ribonukleat atau lebih dikenal sebagai RNA berperan sebagai pembawa bahan genetik dan memainkan peran utama dalam ekspresi genetik. Dalam dogma pokok (*central dogma*) genetika molekular, RNA menjadi perantara antara informasi yang dibawa DNA dan ekspresi fenotipik yang diwujudkan dalam bentuk protein. Struktur untai RNA terdiri atas tiga komponen utama, yaitu

1) gugus fosfat
2) gula ribosa
3) basa nitrogen, yang terdiri dari:
    a. Purin
        i. Adenin (A)
        ii. Guanin (G)
    b. Pirimidin
        i. Sitosin (C)
        ii. Urasil (U)

Purin dan pirimidin yang berkaitan dengan ribosa membentuk suatu molekul yang dinamakan nukleosida atau ribonukleosida, yang merupakan prekursor dasar untuk sintesis DNA. Ribonukleosida yang berkaitan dengan gugus fosfat membentuk suatu nukleotida atau ribonukleotida.RNA merupakan hasil transkripsi dari suatu fragmen DNA, sehingga RNA merupakan polimer yang jauh lebih pendek dibandingkan DNA.

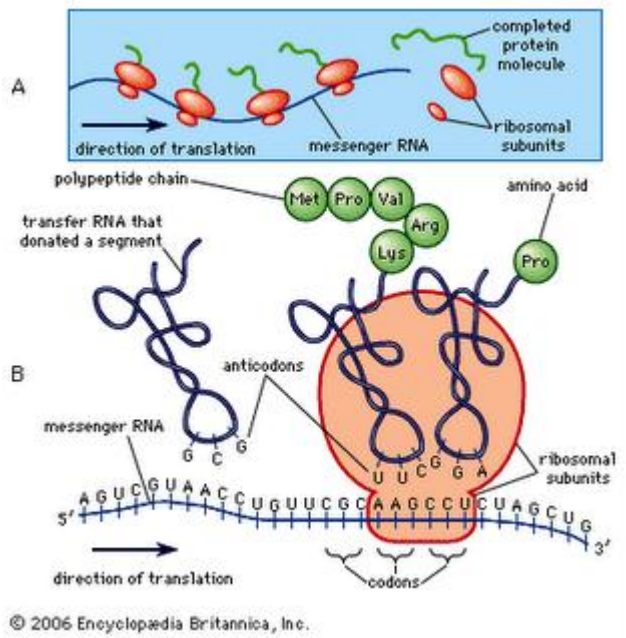


Gambar 3. Struktur RNA[5]

C. Sintesis Protein

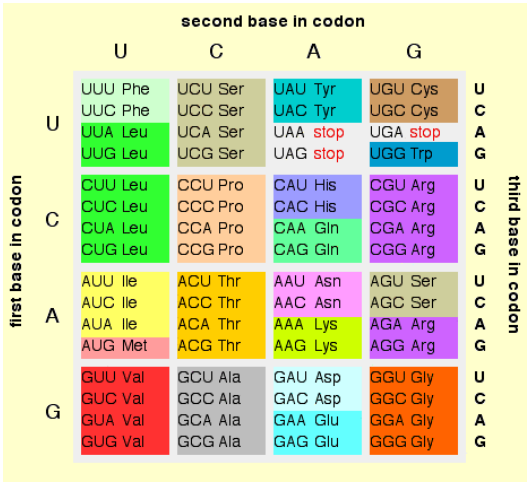
intesis protein adalah proses pembentukan protein dari monomer peptida yang diatur susunannya oleh kode genetik. Sintesis protein dimulai dari anak inti sel, sitoplasma dan ribosom. Sintesis protein terdiri dari dua tahapan besar yaitu:

1) Transkripsi
   DNA akan membuka rantai menjadi dua rantai terpisah. Karena mRNA berantai tunggal, maka salah satu rantai DNA ditranskripsi (di-*copy*). Rantai yang ditranskripsi dinamakan DNA sense atau template dan kode genetik yang dikode disebut kodogen. Sedangkan yang tidak ditranskripsi disebut DNA antisense/komplementer. RNA Polimerase membuka pilinan rantai DNA dan memasukkan nukleotida-nukleotida untuk berpasangan dengan DNA sense sehingga terbentuklah rantai mRNA.
2) Translasi
   mRNA / RNAd yang sudah terbentuk keluar dari anak inti sel menuju rRNA. Disana mRNA masuk ke rRNA / RNAr diikuti oleh tRNA / RNAt. Ketika antikodon pada tRNA cocok dengan kodon mRNA kemudian rantai bergeser ke tengah. Kodon mRNA berikutnya dicocokkan dengan tRNA kemudian asam amino yang pertama berikatan dengan asam amino kedua. tRNA pertama keluar dari rRNA. Proses ini berlangsung hingga kodon stop, ribosom subunit besar dan kecil terpisah, mRNA dan tRNA keluar dari ribosom.



Gambar 4. Translasi mRNA menjadi tRNA yang kemudian menjadi rantai protein

Adapun kode protein hasil translasi yang merupakan rantai kodon pada mRNA dapat dilihat pada gambar berikut ini.



Gambar 5. Tabel protein beserta triplet basa pada mRNA

Dari proses sintesis protein ini dapat diperoleh kesimpulan bahwa kode protein dapat dilihat secara langsung pada mRNA yang merupakan hasil transkripsi dari DNA sense atau dapat langsung berupa DNA antisense itu sendiri.

D. Algoritma Pencocokan String

Algoritma pencarian string atau sering disebut juga pencocokan string adalah algoritma untuk melakukan pencarian semua kemunculan string pendek *pattern*[0..*n* – 1] yang disebut pattern di string yang lebih panjang *teks*[0..*m* − 1] yang disebut teks.

Pencocokkan string merupakan permasalahan paling sederhana dari semua permasalahan string lainnya, dan dianggap sebagai bagian dari pemrosesan data, pengkompresian data, analisis leksikal, dan temu balik informasi. Teknik untuk menyelesaikan permasalahan pencocokkan string biasanya akan menghasilkan implikasi langsung ke aplikasi string lainnya. Algoritma-algoritma pencocokkan string dapat diklasifikasikan menjadi tiga bagian menurut arah pencariannya. Tiga kategori itu adalah :

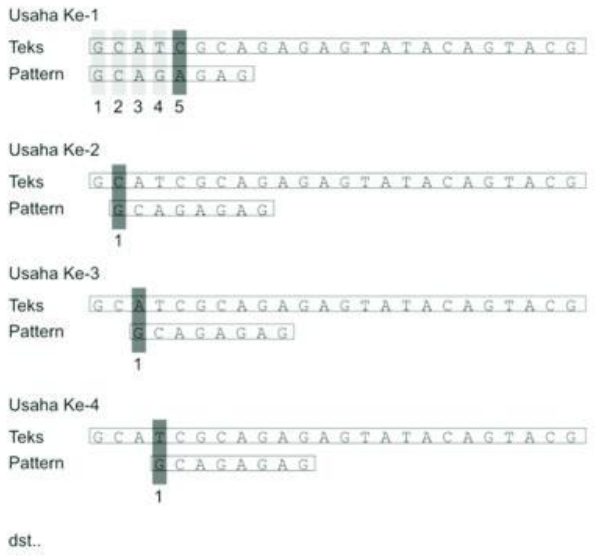
- Dari arah yang paling alami, dari kiri ke kanan, yang merupakan arah untuk membaca, algoritma yang termasuk kategori ini adalah:
  1. Algoritma *Brute Force*
  2. Algoritma dari *Morris* dan *Pratt*, yang kemudian dikembangkan oleh *Knuth*, *Morris*, dan *Pratt*
- Dari kanan ke kiri, arah yang biasanya menghasilkan hasil terbaik secara praktikal, contohnya adalah:
  1. Algoritma dari *Boyer* dan *Moore*, yang kemudian banyak dikembangkan, menjadi Algoritma turbo *Boyer-Moore*, Algoritma tuned *Boyer-Moore*, dan Algoritma *Zhu-Takaoka*;
- Dan kategori terakhir, dari arah yang ditentukan secara spesifik oleh algoritma tersebut, arah ini menghasilkan hasil terbaik secara teoritis, algoritma yang termasuk kategori ini adalah:
  1. Algoritma *Colussi*
  2. Algoritma *Crochemore-Perrin*

E. Algoritma Brute Force

Algoritma *brute force* merupakan algoritma pencocokan string yang ditulis tanpa memikirkan peningkatan performa. Algoritma ini sangat jarang dipakai dalam praktik, namun berguna dalam studi pembanding dan studi-studi lainnya. Secara sistematis, langkah-langkah yang dilakukan algoritma brute force pada saat mencocokkan string adalah:

1) Algoritma *brute force* mulai mencocokkan pattern pada awal teks.
2) Dari kiri ke kanan, algoritma ini akan mencocokkan karakter per karakter *pattern* dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi:
   a. Karakter di pattern dan di teks yang dibandingkan tidak cocok (*mismatch*).
   b. Semua karakter di *pattern* cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.
   c. Algoritma kemudian terus menggeser pattern sebesar satu ke kanan, dan mengulangi langkah ke-2 sampai *pattern* berada di ujung teks.

Berikut adalah algoritma *brute force* yang sedang bekerja mencari *string*:



Gambar 6. Langkah algoritma *brute force*

Kompleksitas waktu untuk algoritma *brute force* sebagai algoritma pencarian string adalah *O(mn)*[1] untuk kasus terburuk dengan kasus pencarian tidak ditemukan dan pada setiap tahap pencarian diperlukan pencocokan terhadap seluruh *pattern* dengan hanya bagian akhir dari *pattern* yang tidak sesuai, contohnya dapat dilihat pada gambar berikut.

String = bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
Pattern = bbbbbbc

Gambar 7. Contoh kasus terburuk untuk algoritma *brute force*

III. ANALISIS

A. Kompresi String *dan* Pattern

Pada masalah pencarian rangkaian asam amino pada DNA, komponen *string* pada algoritma pencarian *string* merupakan rantai DNA itu sendiri yang terdapat pada bagian DNA *antisense* dan *pattern* merupakan rangkaian asam amino yang dicari dalam bentuk kode protein (kodon). Dengan menggunakan algoritma pencarian string biasa seperti *brute force*, pencarian terhadap rantai asam amino tertentu dapat dilakukan dengan mudah. Akan tetapi, waktu pencariannya masih dapat dikatakan cukup lama karena harus menbandingkan satu per satu basa dari DNA yang sudah ditranskripsi dengan rantai kode protein yang ingin dicari.

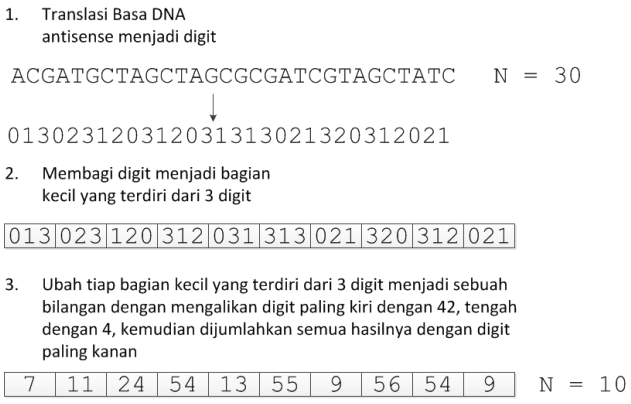
Beberapa cara yang dapat diterapkan untuk mempercepat proses kerja dari algoritma pencarian *string* adalah dengan mengompresi ukuran dari *string* dan ukuran dari *pattern* sehingga waktu untuk perbandingan berkurang karena terjadi pemendekan *string* yang harus dibandingkan. Untuk kasus DNA ini, kompresi terhadap rantai DNA ataupun rantai asam amino dapat dilakukan sehingga ukuran string ataupun pattern dapat berkurang dan kecepatan pencarian akan menjadi lebih cepat.

Kompresi ini dapat dilakukan karena adanya beberapa sifat istimewa yang dapat diterapkan untuk DNA yang ingin dicari berdasarkan rantai asam amino tertentu untuk mengetahui apakah rantai asam amino tersebut terdapat pada rantai DNA tersebut. Pertama, DNA ataupun kode protein hanya terdiri atas empat jenis huruf yang menandakan basa pada DNA atau kode protein. Kedua, asam amino selalu terdiri dari triplet basa sehingga baik rantai asam amino ataupun DNA pasti merupakan kelipatan dari 3[4,5], dan tiap tiga basa dapat direpresentasikan sebagai sebuah integer sehingga terjadi kompresi panjang string ataupun pattern.

Dari dua hal tersebut, maka dapat ditarik pemikiran bahwa proses kompresi untuk DNA dapat dilakukan dengan cara:

1) Setiap basa pada DNA dapat direpresentasikan sebagai angka ‘0’ sampai ‘3’ dengan ‘0’ untuk ‘A’, ‘1’ untuk ‘C’, ‘2’ untuk ‘T’ atau ‘U’, dan ‘3’ untuk ‘G’.
2) Rantai DNA ataupun pattern kemudian dibagi-bagi menjadi bagian-bagian kecil yang terdiri atas digit angka.
3) Setiap bagian yang terdiri dari tiga digit angka, digit paling kiri dikalikan dengan 4^2, kemudian digit tengah dikalikan dengan 4, kemudian dijumlahkan keseluruhan hasilnya dengan digit paling kanan dan disimpan dalam bentuk integer.
4) Lakukan langkah 3) untuk semua bagian baik pada rantai DNA ataupun rantai asam amino.
5) Hasil akhirnya, *string* atau *pattern* akan memendek hingga 1/3 kali dari ukuran aslinya.

Untuk memperjelas langkah-langkah yang dijelaskan di atas dapat dilihat pada gambar berikut.



Gambar 8. Langkah kompresi DNA

B. Modifikasi Algoritma Brute Force

Dengan diterapkannya metode kompresi pada bagian sebelumnya, hampir semua algoritma untuk pencocokan *string* dapat menggunakan teknik kompresi tersebut. Kompresi dilakukan sebelum algoritma utama dijalankan sehingga ketika memasuki bagian utama dari algoritma yang digunakan, *string* dan *pattern* telah mengalami kompresi untuk mempercepat waktu pencarian yang dibutuhkan.

Sebagai contoh, penulis menggunakan algoritma *brute*

force untuk melakukan analisis ketika suatu DNA *antisense* dan rantai asam amino yang ingin dicari kecocokannya dikompresi dibandingkan dengan pencarian dengan menggunakan algoritma *brute force* biasa. Adapun modifikasi yang dilakukan yaitu dengan menambahkan fungsi kompresi sebelum dilakukan pencarian *string*. Fungsi kompresi pada DNA *antisense* ataupun rantai kode protein dapat dilihat pada pseudocode di bawah ini.

```
function kompresi (input str: string) → array of integer
KAMUS
temp : array of integer
i, count, tempvalue, len : integer
ALGORITMA
i ← 0
count ← 0
tempvalue ← 0
len ← length(str)

while (count < len) do
    tempvalue ← 0
    tempvalue ← tempvalue + digit(str_count) * 4 *
                4
    count ← count + 1
    tempvalue ← tempvalue + digit(str_count) * 4
    count ← count + 1
    tempvalue ← tempvalue + digit(str_count)
    count ← count + 1
    temp_i ← tempvalue
    i ← i + 1
endwhile

→ tempvalue
```

Gambar 9. *Pseudocode* fungsi kompresi

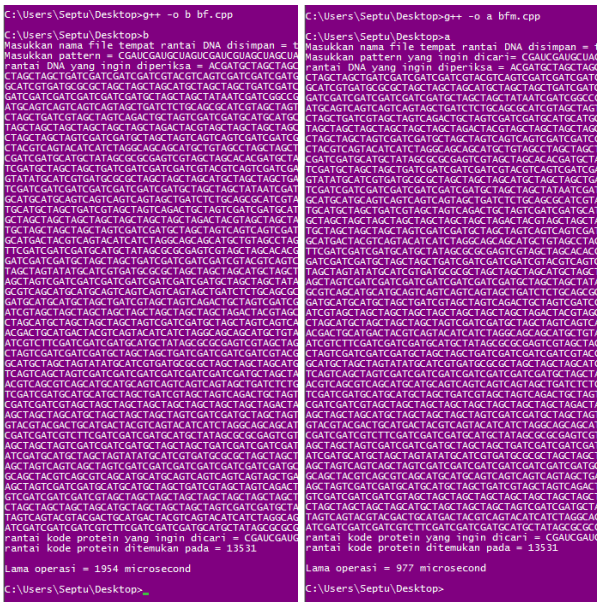
Modifikasi dari algoritma *brute force* dengan penggunaan fungsi kompresi pada Gambar 10 dapat dilihat pada pseudocode berikut ini.

```
function BruteForceMod (input str, pat : string) → integer
KAMUS
tempstr, temppat : array of integer
lenstr, lenpat, i, j, temp : integer
found : Boolean
ALGORITMA
i ← 0
j ← 0
found ← false
tempstr ← kompresi(str)
temppat ← kompresi(pat)
lenstr ← length(tempstr)
lenpat ← length(temppat)
while (not found and i < lenstr) do
    j ← 0
    temp ← i
    while (not found and j < lenpat) do
        if (temppat_j = tempstr_temp) then
            j ← j + 1
            temp ← temp + 1
        else
            found ← true
        endif
    endwhile
    if (found) then
        found ← false
        i ← i + 1
    else
        found ← true
    endif
endwhile

if (not found) then
    i ← -1
endif

→ i
```

Gambar 10. *Pseudocode* algoritma *brute force*.

C. Hasil Analisis

Dengan menggunakan *pseudocode* yang didefinisikan pada Gambar 7 dan Gambar 8 untuk modifikasi pada algoritma *brute force*, maka diperoleh perbandingan kecepatan proses antara dengan menggunakan algoritma *brute force* biasa dibandingkan dengan algoritma *brute force* yang sudah dimodifikasi dengan kasus panjang rantai DNA yang harus dibaca sebanyak 13671 basa dengan pencarian rantai kode protein sebanyak 141 basa dapat dilihat pada gambar berikut ini.



(a) **(b)**

Gambar 11. (a) Hasil pencarian dengan menggunakan algoritma *brute force* biasa, (b) Hasil pencarian dengan menggunakan algoritma *brute force* yang dimodifikasi

Dari hasil percobaan di atas, diperoleh bahwa waktu untuk eksekusi pencocokan rantai DNA dengan menggunakan algoritma *brute force* pada umumnya dibutuhkan waktu selama 1954 *microsecond*, sedangkan untuk algoritma *brute force* yang sudah dimodifikasi dengan adanya pengompresan rantai DNA dan rantai kode protein hanya diperlukan waktu sebesar 977 *microsecond*. Hal ini membuktikan bahwa terdapatnya pengurangan waktu pencarian yang cukup signifikan hingga setengah kali waktu dengan menggunakan algoritma *brute force* yang dimodifikasi. Pada kasus terburuk, algoritma brute force yang dimodifikasi ini akan melakukan perbandingan sebanyak panjang rantai protein(m) / 3 x (panjang rantai

DNA / 3 – m/3 + 1). Kompleksitas waktu untuk algoritma *brute force* yang dimodifikasi ini dalam notasi *big-O* adalah *O(mn/9).*

IV. KESIMPULAN

Untuk kasus pencarian rantai kode protein pada suatu rantai DNA, dapat digunakan kompresi terhadap kedua rantai tersebut dengan memotong menjadi rantai tersebut menjadi bagian-bagian kecil yang terdiri dari tiga basa yang kemudian ditranslasikan menjadi sebuah bilangan sehingga panjang rantai berkurang hingga menjadi 1/3 kali dari panjang semula. Dengan pengurangan panjang tersebut, waktu untuk pencocokan tentunya akan berkurang dan mempercepat pencarian rantai kode protein yang diperlukan. Kompleksitas waktu untuk algoritma brute force yang dimodifikasi dalam notasi *big-O* adalah *O(mn/9)* yang lebih kecil dibandingkan *O(mn)* untuk algoritma brute force biasa.

REFERENCES

[1] Rinaldi Munir, "Diktat Kuliah IF3051 Strategi Algoritma," Teknik Informatika Institut Teknologi Bandung, Bandung, 2009.

[2] http://ghr.nlm.nih.gov/handbook/basics/dna, diakses pada tanggal 7 Desember 2011, pukul 17.20 WIB

[3] http://id.wikipedia.org/wiki/Algoritma_pencarian_string, diakses pada tanggal 7 Desember 2011, pukul 19.50 WIB

[4] http://id.wikipedia.org/wiki/Asam_deoksiribonukleat, diakses pada tanggal 7 Desember 2011, pukul 17.17 WIB

[5] http://id.wikipedia.org/wiki/Asam_ribonukleat, diakses pada tanggal 7 Desember 2011, pukul 19.20 WIB

[6] http://kbs.jogja.go.id/upload/DNA.doc, diakses pada tanggal 7 Desember 2011, pukul 18.17 WIB

[7] http://mr-fabio2.blogspot.com/2009/02/sintesis-protein.html, diakses pada tanggal 7 Desember 2011, pukul 19.30 WIB

[8] http://prestasiherfen.blogspot.com/2009/09/normal-0-false-false-false-en-us-x-none.html, diakses pada tanggal 7 Desember 2011, pukul 18.44 WIB

[9] http://widagdomahendro.wordpress.com/2010/07/23/asam-deoksiribonukleat-dna/, diakses pada tanggal 7 Desember 2011, pukul 18.17 WIB

[10] http://www.chemguide.co.uk/organicprops/aminoacids/dna5.html, diakses pada tanggal 7 Desember 2011, pukul 18.44 WIB

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, 8 Desember 2011

Septu Jamasoka