

Penerapan Algoritma Brute Force pada Formulasi Jumlah Varian Fragmen DNA (*Partial Digest Problem*)

Andzarrahim, Tara Baskara

*Laboratorium Ilmu dan Rekayasa Komputasi
Departemen Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung*

E-mail : if14013@students.if.itb.ac.id, if14042@students.if.itb.ac.id

Abstraksi

DNA merupakan substansi yang terkecil dalam tubuh makhluk hidup. Akan tetapi, pada setiap DNA yang terdapat dalam tubuh makhluk hidup tersebut mengandung sangat banyak informasi penting yang menyusun struktur sel yang akan membangun seluruh organ pada makhluk hidup tersebut. Sekarang ini, teknologi berkembang dengan sangat pesat terutama dalam bidang rekayasa genetika, seperti : *cloning, teknologi DNA rekombinan, dan DNA mapping*. Hampir setiap teknologi di atas menggunakan metode pemecahan DNA (*DNA Fragmentation*) dalam proses pengembangannya. Oleh karena itu, dalam makalah ini kami berusaha menjelaskan proses pencarian jumlah varian dalam pemecahan DNA tersebut dengan menerapkan algoritma Brute Force. **Algoritma Brute Force** merupakan algoritma yang paling sederhana pada metode pencarian jumlah varian dalam pemecahan DNA, walaupun dalam segi waktu, algoritma ini terbilang algoritma yang terlamban.

Kata kunci : *DNA fragmentation, Brute Force, DNA*

1. Pendahuluan

DNA merupakan substansi terkecil dalam makhluk hidup yang mengandung karakteristik dan komposisi tertentu. Setiap karakteristik maupun komposisi yang disusun dari kombinasi ATSG nantinya akan memberikan informasi khusus pada kelangsungan makhluk hidup tersebut..

Pada zaman sekarang ini, terutama dalam bidang rekayasa teknologi, telah banyak teknik yang dikembangkan dalam memanipulasi baik komposisi maupun karakteristik dari DNA tersebut, sebagai contoh dalam teknologi *cloning*, teknologi ini menggunakan metode DNA Fragmentation sebagai landasan fungsinya dan panjang dari DNA tersebut sebagai parameter fungsi tersebut. Permasalahan dalam pemecahan DNA bisa dipecah mejadi beberapa bagian kecil, sebagai contoh dalam memecah potongan DNA untuk digabung, karena dalam proses pemotongan tersebut kadar enzim restriksi tersebut bisa berbeda sehingga

membedakan panjang dari setiap potongan dari DNA tersebut.

Permasalahan dalam DNA fragmentasi terutama dalam memprediksi jumlah varian dari fragmen DNA tersebut bisa dibilang masalah efektifitas waktu dan tenaga. Oleh karena itu, banyak ilmuwan terkenal seperti Werner Arber (penemu enzim restriksi), Daniel Nathans (peraih nobel di bidang BioInformatika), dan Hamilton Smith (ilmuwan pertama yang membuktikan enzim restriksi dalam memotong DNA) mencoba meyelesaikan masalah pemecahan DNA (DNA Fragmentation) dengan menerapkan berbagai algoritma yang ada terutama dalam kasus ini menggunakan *algoritma brute force*.

Tujuan pembuatan makalah :

1. Memprediksi jumlah varian dalam proses pemecahan DNA (*DNA Fragmentation*).
2. Menerapkan algoritma Brute Force dalam proses pemecahan DNA (*DNA Fragmentation*)

2. Algoritma Brute Force

Algoritma Brute Force dikenal algoritma yang paling tidak cerdas dan tidak mangkus dibandingkan algoritma lainnya karena dalam masalah berukuran besar algoritma ini membutuhkan juga banyak langkah penyelesaian. Akan tetapi dalam kasus ini algoritma brute force terbilang sangat efisien walaupun jarang digunakan.

Algoritma ini bekerja dengan mencocokkan setiap data L dengan DX (yang didapatkan dari set X), apabila satu saja data yang dicocokkan tidak sesuai maka data L yang dimasukkan bukan merupakan hasil “spekulasi” dari DX. Dalam proses pengerjaannya, algoritma ini terdiri dari empat tahap, pertama mengurutkan jsj dari array L (karena masukan nilai bisa acak), lalu algoritma ini mengkalkulasikan nilai set DX dari array X dengan cara mencari selisih dari isi nilai yang bernilai selain nol. Setelah itu algoritma ini akan menyusun nilai dari set DX yang terbentuk dari proses perhitungan sebelumnya, dan yang terakhir dari kedua array yang dibentuk di atas (DX dan L) akan dicocokkan nilainya satu dengan yang lain dan apabila ada satu saja nilai yang tidak sama maka akan keluar komentar “tidak ketemu”.

Algoritma ini memiliki kompleksitas waktu dan ruang $O(M^{n2-n})$, ini disebabkan karena pada proses pencarian semua kemungkinan set (DX), algoritma ini mencoba semua selisih dari nilai yang ada di set X dan memasukkannya ke dalam array.

2.1 Pseudo – code Algoritma Brute Force

Procedure **SortArray** (L : array of integer)

Deklarasi

X : integer
Y : integer

Algoritma

```
X <- 0
While (L[X] /= nil) do
  Y <- X + 1
  Temp <- L[X]
  While (L[Y] /= nil) do
```

```
    If (L[Y] <= L[X])
      Temp <- L[X]
      L[X] <- L[Y]
      L[Y] <- temp
    Y++
  X++
```

Function **CreateDX** (X: array of integer) -> L

Deklarasi

L : array of integer
K : integer
V : integer
J : integer

Algoritma

```
K <- 0
J <- 0
While (X[K] /= nil) do
  V <- K + 1
  While (X[V] /= nil) do
    If (X[V] - X[K] /= 0)
      L[J] <- X[V] -
X[K]
      J++
    V++
  K++
----->L
```

Function **VarianMatch** (L : array of integer, X : array of integer) -> X

{Prekondisi : jika array tidak cocok maka X akan di- nil kan}

Deklarasi

DX : array of integer
J : integer

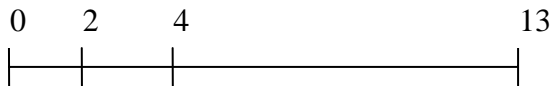
Algoritma

```
SortArray (L)
DX <- CreateDX(X)
SortArray (DX)

If (DX = L) then
  -----> X
Else
  Write ("hasil tidak sama")
  X <- nil
  -----> X
```

2.2 Penerapan Algoritma Brute Force pada Proses Pencarian Jumlah Varian DNA

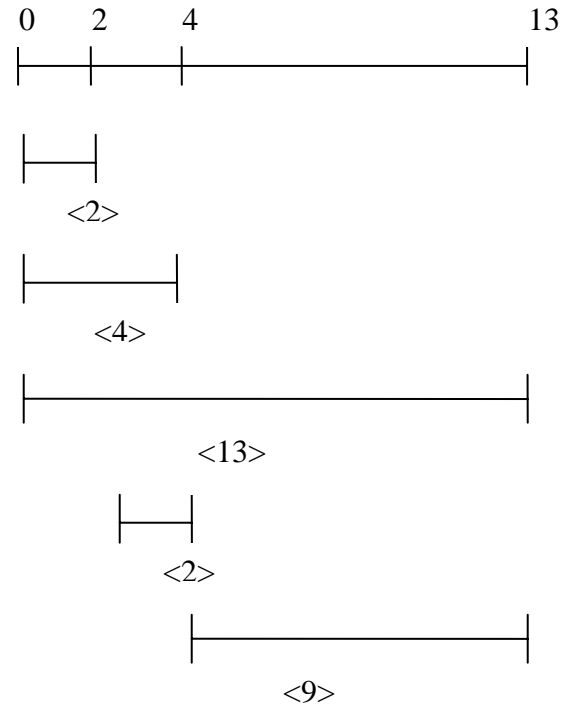
Andaikan ada sebuah pita DNA yang memiliki kecenderungan untuk membelah pada titik tertentu, misal ada sebuah set $X = \{0,2,4,13\}$ (ket : angka angka adalah panjang titik dimana pita memiliki kecenderungan untuk membelah).



Kemudian ada nilai L yang berisikan $\{9,13,4,2,11,2\}$ (belum tersusun) yang akan dicocokkan dengan nilai kemungkinan dari multiset DX . Berdasarkan nilai X dapat diciptakan DX dengan mencari nilai selain nol dari isi elemen X .

X	0	2	4	13
0	-	2	4	13
2	-	-	2	11
4	-	-	-	9
13	-	-	-	-

Berdasarkan nilai tabel di atas, maka $DX = \{2,4,13,2,11,9\}$ (belum tersusun). Nilai nilai ini didapatkan dengan mencari selisih dari setiap elemen X , misal $11 = 13 - 2$, $9 = 13 - 4$.



Berdasarkan algoritma di atas nilai dari setiap elemen DX dan L akan disilangkan dan akan ditarik kesimpulan apakah nilai L yang dimasukkan memenuhi semua nilai DX yang didapat dari X .

$$L = \{9,13,4,2,11,2\}$$

$$X = \{0,2,4,13\}$$

$$DX = \{2,4,13,2,11,9\}$$

Setelah di sort, kedua array tersebut akan disamakan dan apabila keduanya (DX dan L) sama maka X adalah set nilai yang mungkin untuk membelah bagi sebuah pita DNA.

3. Kesimpulan dan Saran

Dari makalah ini, kita bisa menarik kesimpulan bahwa algoritma brute force yang diterapkan pada proses pemecahan DNA sangat efisien dibanding algoritma lainnya. Selain itu, algoritma yang diimplementasikan terlihat sangat sederhana dan sangat mudah untuk dipelajari (bisa dilihat dalam fungsi (CreateDX)).

Saran untuk masalah pemecahan DNA ini adalah adanya penelitian lebih lanjut tentang penggunaan algoritma lain selain algoritma brute force untuk membuktikan kebenaran bahwa algoritma brute force merupakan algoritma yang paling efisien dalam masalah ini.

4. Referensi

- [1] Smith, Hamilton. An Introduction to Bioinformatics Algorithms. www.bioalgorithms.info. Diakses tanggal 16 Mei 2006.
- [2] Munir, Rinaldi, 2005. Strategi Algoritmik. Teknik Informatika ITB : Bandung.