

Perbandingan Algoritma Brute Force ,KMP dan Boyer-Moore untuk Pencarian Gugus Asam Amino pada Rantai Double Helix

Pratama Nugraha Damanik , 13513001
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
tamadamnik@students.itb.ac.id

Abstract— Saat ini, perkembangan algoritma semakin memudahkan manusia untuk menyelesaikan beberapa persoalan yang ada, salah satunya algoritma String Matching . Algoritma String Matching bertujuan untuk mendapatkan kemunculan suatu pattern pada sebuah text . Algoritma ini banyak digunakan di berbagai bidang, salah satunya di bidang kedokteran . Di bidang kedokteran, algoritma ini dapat digunakan untuk pencocokan DNA . Dengan algoritma ini, dapat dicari asam amino tertentu dalam satu rantai DNA . Dari berbagai macam algoritma pencocokan string, dalam makalah ini akan dibandingkan algoritma KMP dan algoritma Boyer-Moore untuk pencarian asam amino dalam gugus rantai DNA .

Index Terms—String Matching, Knuth-Morris-Pratt,Boyer-Moore,DNA .

I. PENDAHULUAN

Di era globalisasi saat ini, perkembangan teknologi tidak hanya berdampak pada bidang teknologi itu sendiri, namun berdampak ke bidang lain seperti keamanan, transportasi, bahkan kesehatan . Banyak penemuan penemuan baru di bidang teknologi yang berdampak pada kehidupan manusia, seperti penemuan alat baru, teknik tertentu, atau algoritma baru. Dengan perkembangan teknologi saat ini, berbagai permasalahan manusia di berbagai bidang dapat diselesaikan dengan mudah .

Salah satu perkembangan teknologi, yaitu perkembangan algoritma, sudah banyak berdampak pada kehidupan manusia . Misalnya, adanya algoritma Greedy memudahkan manusia untuk menemukan solusi yang baik untuk suatu persoalan dengan waktu yang relative singkat . Contoh lain Algoritma Divide and Conquer yang mempercepat proses pengurutan suatu array bilangan atau pemecahan masalah Closest Pair . Selain itu adanya Algoritma Branch and Bound yang mempercepat penyelesaian masalah TSP . Lalu Algoritma String Matching yang memudahkan kita untuk menemukan suatu pattern tertentu dalam suatu String .

Salah satu algoritma yang banyak digunakan saat ini adalah algoritma String Matching . Algoritma ini adalah algoritma pencocokan pattern tertentu dalam suatu string .

Pada kehidupan sehari – hari, algoritma ini banyak diimplementasikan di berbagai bidang . Contoh yang kita sering temukan adalah ketika kita menggunakan fitur “Find” pada Microsoft Word, atau pencarian file dalam computer, bahkan pencarian kata pada Google juga menggunakan String Matching .

Dalam dunia medis, Algoritma String Matching dapat diaplikasikan pada pencarian suatu asam amino dalam rantai DNA . Apabila dengan Brute Force, pencarian suatu gugus asam amino dalam rantai DNA akan memakan waktu yang lama, namun dengan menerapkan algoritma String matching, proses pencarian ini akan berlangsung dengan cepat .

Suatu rantai DNA mengandung berbagai jenis asam amino . Susunan DNA akan di translasi oleh RNA polymerase sehingga sebagian rantai double helix DNA akan membuka . Lalu susunan kode genetik pada DNA akan disalin menjadi basa N lalu membentuk rantai Mrna . Rantai mRNA inilah yang mengandung berbagai jenis asam amino . Asam amino sendiri terdiri atas 20 asam amino, 1 asam amino mulai dan 1 asam amino berhenti . Setiap asam amino terdiri dari 3 kode genetik yang merepresentasikan jenis asam amino yang dikandungnya .

II. DASAR TEORI

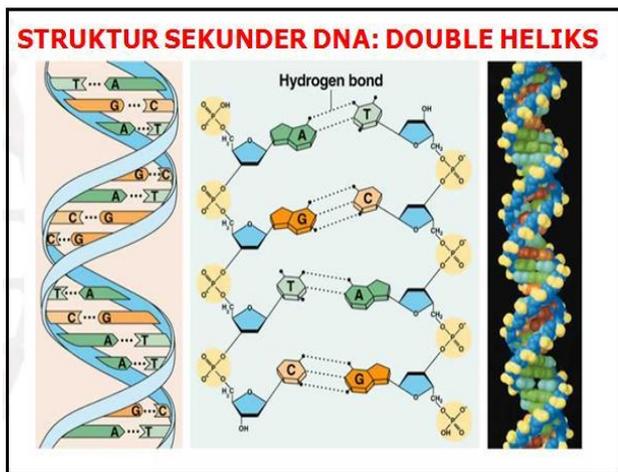
A. DNA

DNA atau *deoxyribonucleic acid* , adalah sejenis biomolekul yang menyimpan dan menyandi instruksi – instruksi genetika setiap organisme dan banyak jenis virus . DNA merupakan asam nukleat bersamaan dengan protein dan karbohidrat . Asam nukleat adalah makromolekul esensial bagi seluruh makhluk hidup . Kebanyakan molekul DNA terdiri dari dua ulir biopolymer yang berpilin satu sama lainnya . Dua ulir DNA ini dikenal sebagai polinukleotida karena keduanya terdiri dari satuan satuan molekul yang disebut nukleotida . Nukleotida terdiri dari atas salah satu jenis basa nitrogen, yaitu :

- Guanin (G)

- Adenin (A)
- Timin (T)
- Sitosin (S)

Tiap tiap nukleotida akan tersambung dalam satu rantai ikatan kovalen antara gula satu nukleotida dengan fosfat nukleotida lainnya. Hasilnya adalah rantai punggung gula fosfat yang berselang seling. Pasangan basa (Adenin dengan Timin; Sitosin dengan Guanin), ikatan hydrogen mengikat basa-basa dari kedua ulir polinukleotida membentuk DNA ulir ganda. Rantai DNA sendiri memiliki lebar 22-24 Å. Sementara, panjang satu unit nukleotida adalah 3,3 Å. Meskipun unit monomer ini sangat kecil, satu DNA dapat memiliki jutaan nukleotida yang terangkai menjadi rantai. Misalnya, pada manusia, kromosom terbesarnya memiliki 220 juta nukleotida.



Gambar : Struktur Sekunder DNA Double Helix

Sumber : <http://adilarosyida.blogspot.com/>

B. RNA

RNA atau *ribonucleic acid* adalah satu dari tiga makromolekul utama (bersama dengan DNA dan protein) yang berperan penting dalam segala bentuk kehidupan. Asam ribonukleat berperan sebagai pembawa kode genetik dan memegang peran utama dalam ekspresi genetik. Dalam genetika molekular, RNA menjadi perantara antara informasi yang dibawa DNA dan ekspresi fenotipik yang diwujudkan dalam bentuk protein.

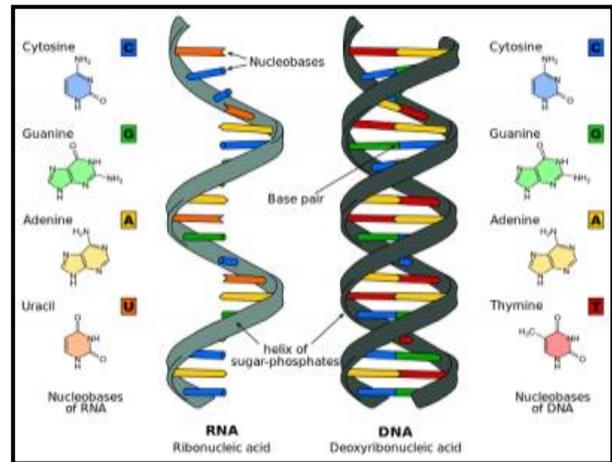
Struktur dasar RNA sendiri mirip dengan DNA. RNA merupakan polimer yang tersusun dari sejumlah nukleotida. Setiap nukleotida pada RNA memiliki satu gugus fosfat, satu gugus pentose dan satu gugus basa nitrogen. Polimer tersusun dari ikatan berselang seling antara gugus fosfat dari satu nukleotida dengan gugus pentosa dari nukleotida yang lain. Basa nitrogen pada RNA sama dengan DNA, hanya pada RNA basa timin diganti menjadi urasil.

RNA terdiri dari 3 jenis, yaitu :

- RNA-kurir (messenger-RNA), yang disintesis dengan RNA polymerase I

- RNA-ribosom (ribosomal-RNA), yang disintesis dengan RNA polimerasi II
- RNA-transfer (transfer-RNA), yang disintesis dengan RNA polymerase III.

Peran penting RNA terletak pada fungsinya sebagai perantara antara DNA dan protein dalam proses ekspresi genetik karena ini berlaku untuk semua organisme hidup. Dalam peran ini, RNA diproduksi sebagai salinan kode urutan basa nitrogen DNA dalam proses transkripsi. Kode urutan basa ini tersusun dalam bentuk triplet, tiga urutan basa N, yang dikenal dengan nama kodon. Setiap kodon berelasi dengan satu asam amino dan monomer yang menyusun protein.



Gambar : Perbandingan Struktur RNA dengan DNA

Sumber : <http://idpengertian.com>

C. Proses Sintesis dari DNA ke RNA

DNA yang terletak di dalam nukleus merupakan suatu cetakan kode genetik yang menghasilkan informasi genetik. Kode genetik disusun oleh urutan basa nitrogen (A, T, G, dan C). Dalam sintesis protein, kode-kode genetik dalam DNA disalin menjadi mRNA. Proses ini disebut transkripsi.

Proses ini diawali dengan melekatnya RNA polimerase pada molekul DNA sehingga sebagian rantai double helix DNA membuka. Akibatnya, salah satu rantai DNA yang membuka tersebut mencetak RNA. Rantai DNA yang mengandung kode-kode genetik (kodon) dan dapat mencetak mRNA disebut rantai sense. Rantai DNA yang tidak mencetak mRNA disebut rantai antisense. Misalnya urutan basa N pada rantai DNA terdiri atas TAC, GCT, CGA, dan CTA maka urutan basa N pada rantai mRNA yaitu AUG, CGA, GCU, dan GAU.

Setelah disalin, mRNA keluar dari nukleus menuju sitoplasma. mRNA tidak dapat mengenali suatu asam amino secara langsung. Oleh karena itu, diperlukan tRNA untuk dapat membaca kode-kode yang dibawa mRNA. Di dalam sitoplasma banyak terdapat tRNA, asam amino dan enzim amino asil sintetase. Asam amino tersebut diaktifkan

menggunakan ATP (Adenosin Trifosfat) dan enzim amino asil sintetase sehingga dihasilkan Amino asil Adenosin monofosfat (AA-AMP) dan fosfat organik.

Selanjutnya Aminoasil Adenosin monofosfat diikat oleh t-RNA dan dibawa ke ribosom. Setiap tRNA memiliki tiga basa N dan asam amino, tiga basa N tRNA akan berpasangan dengan tiga basa N mRNA yang sesuai. mRNA merupakan susunan kodon yang panjang. Setiap tRNA akan menerjemahkan tiga basa.

Setelah tRNA pertama melepaskan diri, datang tRNA selanjutnya, begitu terus-menerus sampai kodon pada mRNA habis. Asam amino yang terbentuk selama penerjemahan oleh tRNA akan membentuk suatu ikatan. Bagian basa N pada tRNA yang menerjemahkan kode yang dibawa mRNA disebut *antikodon*.

Sementara itu, tiga bagian basa N pada mRNA tersebut di atas yang memiliki kode untuk menspesifikasikan asam amino disebut *kodon*. Proses penerjemahan kode yang dibawa mRNA oleh tRNA disebut translasi. Asam amino-asam amino akan berjajar membentuk urutan sesuai dengan kode yang dibawa mRNA sehingga terbentuklah protein..

Tabel berikut berisi hubungan antara kodon tiga huruf dengan asam amino yang ditambahkan pada molekul protein yang disusun oleh sebuah RNA. Tabel ini menggunakan basa nukleotida RNA, yaitu U, C, A, dan G, dan bukan basa DNA T, C, A, dan G. Kodon mulai dan berhenti mengendalikan mulai dan berakhirnya transkripsi RNA.

| | | Anden base | | | | |
|-------------|---|--|-----------------------------------|--|--|------------------|
| | | U | C | A | G | |
| Förste base | U | UUU Phenyl-alanin UUC UUA Leucin UUG | UCU Serin UCC UCA UCG | UAU Tyrosin UAC UAA Stop kodon UAG | UGU Cystein UGC UGA Stop kodon UGG Tryptophan | U C A G |
| | C | CUU Leucin CUC CUA CUG | CCU Prolin CCC CCA CCG | CAU Histidin CAC CAA Glutamin CAG | CGU Arginin CGC CGA CGG | U C A G |
| | A | AUU Isoleucin AUC AUA AUG Methionin Start kodon | ACU Threonin ACC ACA ACG | AAU Aspargin AAC AAA Lysin AAG | AGU Serin AGC AGA Arginin AGG | U C A G |
| | G | GUU Valin GUC GUA GUG | GCU Alanin GCC GCA GCG | GAU Aspargin syre GAC GAA Glutamin syre GAG | GGU Glysin GGC GGA GGG | U C A G |

Gambar : Tabel asam amino

Sumber : <https://fajarcahyani46.wordpress.com>

D. Algoritma Brute Force untuk Pencarian Asam Amino

Algoritma Brute Force adalah algoritma penyelesaian masalah tanpa strategi. Inti dari algoritma ini adalah yang penting menyelesaikan persoalan tanpa memperhatikan masalah memori dan waktu. Langkah penyelesaian :

1. Pencocokan dimulai karakter pertama
2. Jika pencocokan tidak berhasil, maka geser

keyword 1 karakter ke kanan, lalu cocokkan kembali karakter keyword dengan string satu per satu.

Namun, pada pencarian asam amino hasil yang diterima adalah hasil yang jika indeks karakter awalnya adalah kelipatan 3, karena pada gugus asam amino, setiap asam amino di kodekan dengan 3 basa N, sehingga jika indeks karakter awal bukan kelipatan 3, maka hasilnya tidak diterima .

E. Algoritma KMP untuk Pencarian Asam Amino

Algoritma KMP adalah salah satu algoritma pencarian pattern dari kiri ke kanan, namun algoritma ini melangkahi huruf lebih cerdas daripada algoritma brute force . Algoritma ini ditemukan oleh Donald Knuth, Vaughan Pratt dan James H. Morris pada tahun 1974 . Pada algoritma ini, pengecekan pattern menggunakan “boundary value” yang dimiliki oleh pattern yang ingin dicari . Oleh karena itu, sebelum melakukan proses *matching*, nilai boundary untuk pattern harus dicari terlebih dahulu . Langkah pencarian nilai boundary :

1. Inisiasi nilai boundary value untuk panjang string 1 sama dengan 0
2. Lalu untuk substring dengan panjang 2 sampai n lakukan :
 - a. Cari jumlah karakter sama terbanyak dari substring tersebut dari suffix dan prefixnya . Nilai boundary tidak boleh maksimal adalah panjang string – 1 .
Contoh :
abaab → nilai boundary=2 yaitu “ab”
ab → nilai boundary 0
abaaba → nilai boundary=3, yaitu “aba”

Setelah nilai boundary untuk tiap kata ditemukan, lalu lakukan proses String Matching. Proses melakukan string matching adalah sebagai berikut :

1. Mulai langkah pencocokan dari karakter pertama
2. Cari berapakah jumlah karakter dari keyword yang cocok
3. Jika tidak seluruh karakter keyword cocok, maka cari nilai boundary untuk x karakter yang cocok . Misalkan nilai boundary untuk x karakter adalah y .
4. Lalu geser keyword sejauh “ $x-y$ ”
5. Setelah itu pencarian dilakukan mulai dari karakter karakter keyword ke $y+1$.

III. HASIL PENCARIAN GUGUS PROTEIN DALAM RANTAI DNA DENGAN 3 ALGORITMA PENCARIAN

Pada percobaan ini, langkah langkah pengujian adalah sebagai berikut :

- Rantai DNA akan di-generate secara random mulai dari panjangnya dan protein penyusunnya
- Lalu protein yang akan dicari juga di-generate secara random mulai dari panjangnya dan gugus proteinnya
- Setelah itu, rantai DNA dan gugus protein akan diproses dengan Algoritma Brute Force, KMP, dan Boyer-Moore
- Dalam setiap algoritma, rantai DNA akan terlebih dahulu diubah menjadi rantai rNA, lalu gugus protein juga akan diterjemahkan menjadi kode genetik, karena kode genetic yang menjadi input masih dalam berbentuk nama asam amino, bukan dalam kode genetic
- Setelah proses translasi selesai, lalu akan dijalankan String Matching tiap algoritma dengan inputnya adalah rantai rNA dan gugus protein yang sudah dalam bentuk kode genetic.

Berikut hasil beberapa percobaan :

```
C:\Users\Tama Dananik\Documents\NetBeansProjects\StringMatching\dist>java -jar StringMatching.jar
%% Rantai DNA : GACCAAAGTCAGGTACAGATAGTACAAATACCCCGAGGAGGUTACTACTCGGGAGGUC
ATCAATGAAACGACCTGCTACCAATGAGGTATGTGAGGUGGGAGTGGGAGGUGTTCAGCGCTCAGGAGGUCAC
C %%
%% Rantai mRNA : GACCAAAGUCGAGUACAGAUAGGUAACAUAACCCCGAGGAGUACUACUACCGGAGGU
CAUCAUUGAAACGACGUCUACCAUAGGUGUUGGAGGUGGGAGUGCGGAGGUGUUCACCGUCAGGAGGUCACU
CC %%

==== Algoritma BruteForce ====
@ Gugus Protein : Glisin1 Glisin4
@ Gugus Protein (Kode Genetik) : GGAGGU
@ Panjang mRNA : 145
@ Panjang Gugus Protein : 6
@ Total perbandingan : 221
@ Waktu : 436.921 milisecond
@ Total Pattern yang sama : 5! Asam Amino yang sama : 2
Indeks : 39*, 57*, 100, 115, 133.
*Asam amino ditemukan jika indeks awal pattern yang sama merupakan kelipatan 3

==== Algoritma KMP ====
@ Gugus Protein : Glisin1 Glisin4
@ Gugus Protein (Kode Genetik) : GGAGGU
@ Panjang mRNA : 145
@ Panjang Gugus Protein : 6
@ Total perbandingan : 110
@ Waktu : 111.43 milisecond
@ Total Pattern yang sama : 5! Asam Amino yang sama : 2
Indeks : 39*, 57*, 100, 115, 133.
*Asam amino ditemukan jika indeks awal pattern yang sama merupakan kelipatan 3

==== Algoritma Boyer-Moore ====
@ Gugus Protein : Glisin1 Glisin4
@ Gugus Protein (Kode Genetik) : GGAGGU
@ Panjang mRNA : 145
@ Panjang Gugus Protein : 6
@ Total perbandingan : 105
@ Waktu : 233.055 milisecond
@ Total Pattern yang sama : 5! Asam Amino yang sama : 2
Indeks : 39*, 57*, 100, 115, 133.
*Asam amino ditemukan jika indeks awal pattern yang sama merupakan kelipatan 3
```

```
C:\Users\Tama Dananik\Documents\NetBeansProjects\StringMatching>cd dist
C:\Users\Tama Dananik\Documents\NetBeansProjects\StringMatching\dist>java -jar StringMatching.jar
%% Rantai DNA : GATGAAGUUCGAGTCGGGTGGTGCATATACAAGUUGAAGUUAACUUTTTTCCTGGACACA
CGAAACCGAGGATAGATGTCGGGGATCTGGTGAAGUUGGTGAGUUTAGACGGAATCAAGTTACCGATGTCGGC
GTGAGUUTCTGAGATCCCGGAGUUAACCCCAACGACGCTCGATGAGUUTGCGGAGUUTGTATTGCGTTG
AAGUUGTGTGACGAAATGACACATGAGUUCGTCCTGTAAGAGAGUACCTACCTCCCGTTAATGTGATA %%
%% Rantai mRNA : GAUCAAUUUCCAGUCGGUUCGUGAUAAGAUUAAGAUUAAGAUUUUUUUGGUCGACAC
ACGAAACCGAGUAGUUCGCGGAGUCUGGUUGAAGUUGGUAAGUUAAGCGAAUCAAAGUUAACGAGUGCC
GGUAAUUUUUUGAAGUCCCGGAGUUAACCCCAACGACGUCGUUAAGAUUAUUGCCGAGUUUUUUAUUGCCU
GAGUUGGUCUACGAAAUUGAAGAUUAAGUUCGUCUUAAGAUUAACCUACCUCCCGUUAUUGCAUA %%

==== Algoritma BruteForce ====
@ Gugus Protein : Glumatat2 Valin2
@ Gugus Protein (Kode Genetik) : GAAGUU
@ Panjang mRNA : 299
@ Panjang Gugus Protein : 6
@ Total perbandingan : 442
@ Waktu : 651.717 milisecond
@ Total Pattern yang sama : 13! Asam Amino yang sama : 6
Indeks : 3*, 31, 37, 43, 99*, 108*, 146, 165*, 193, 204*, 223, 252*, 269,
*Asam amino ditemukan jika indeks awal pattern yang sama merupakan kelipatan 3

==== Algoritma KMP ====
@ Gugus Protein : Glumatat2 Valin2
@ Gugus Protein (Kode Genetik) : GAAGUU
@ Panjang mRNA : 299
@ Panjang Gugus Protein : 6
@ Total perbandingan : 226
@ Waktu : 216.995 milisecond
@ Total Pattern yang sama : 13! Asam Amino yang sama : 6
Indeks : 3*, 31, 37, 43, 99*, 108*, 146, 165*, 193, 204*, 223, 252*, 269,
*Asam amino ditemukan jika indeks awal pattern yang sama merupakan kelipatan 3

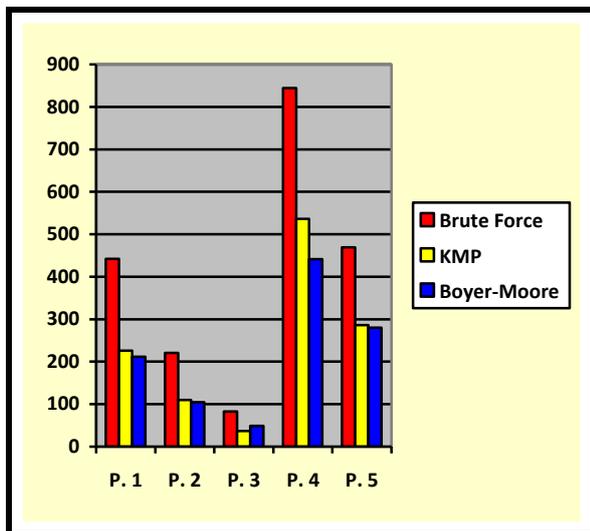
==== Algoritma Boyer-Moore ====
@ Gugus Protein : Glumatat2 Valin2
@ Gugus Protein (Kode Genetik) : GAAGUU
@ Panjang mRNA : 299
@ Panjang Gugus Protein : 6
@ Total perbandingan : 212
@ Waktu : 389.271 milisecond
@ Total Pattern yang sama : 13! Asam Amino yang sama : 6
Indeks : 3*, 31, 37, 43, 99*, 108*, 146, 165*, 193, 204*, 223, 252*, 269,
*Asam amino ditemukan jika indeks awal pattern yang sama merupakan kelipatan 3
```

```
C:\Users\Tama Dananik\Documents\NetBeansProjects\StringMatching\dist>java -jar StringMatching.jar
%% Rantai DNA : TCTGGAGTGGATCTATTACCTAAATTACGGCGTAGGACCGATGGATCCTCGGGATCTT
%% Rantai mRNA : UCUCGGAGUGGAUCUUAUUCACCUAAAUUACGGCGUAGGACCGAUGGAUCUUGGGAUCUU

==== Algoritma BruteForce ====
@ Gugus Protein : Glisin1
@ Gugus Protein (Kode Genetik) : GGA
@ Panjang mRNA : 62
@ Panjang Gugus Protein : 3
@ Total perbandingan : 83
@ Waktu : 208.197 milisecond
@ Total Pattern yang sama : 5! Asam Amino yang sama : 3
Indeks : 4, 9*, 37, 45*, 54*,
*Asam amino ditemukan jika indeks awal pattern yang sama merupakan kelipatan 3

==== Algoritma KMP ====
@ Gugus Protein : Glisin1
@ Gugus Protein (Kode Genetik) : GGA
@ Panjang mRNA : 62
@ Panjang Gugus Protein : 3
@ Total perbandingan : 37
@ Waktu : 79.907 milisecond
@ Total Pattern yang sama : 5! Asam Amino yang sama : 3
Indeks : 4, 9*, 37, 45*, 54*,
*Asam amino ditemukan jika indeks awal pattern yang sama merupakan kelipatan 3

==== Algoritma Boyer-Moore ====
@ Gugus Protein : Glisin1
@ Gugus Protein (Kode Genetik) : GGA
@ Panjang mRNA : 62
@ Panjang Gugus Protein : 3
@ Total perbandingan : 49
@ Waktu : 361.413 milisecond
@ Total Pattern yang sama : 5! Asam Amino yang sama : 3
Indeks : 4, 9*, 37, 45*, 54*,
*Asam amino ditemukan jika indeks awal pattern yang sama merupakan kelipatan 3
```

Grafik : Jumlah perbandingan tiap percobaan

V. CONCLUSION

Dari hasil percobaan pengujian “Pencarian Gugus Protein dalam Rantai DNA” dapat disimpulkan :

- Algoritma Brute Force tidak cocok digunakan dalam kasus ini, karena dalam setiap percobaan, algoritma ini memerlukan waktu yang lebih besar dan jumlah perbandingan yang besar
- Algoritma KMP dapat digunakan untuk kasus ini jika mempertimbangkan kebutuhan waktu yang singkat, karena dalam beberapa percobaan algoritma KMP memerlukan waktu yang paling sedikit
- Algoritma Boyer-Moore dapat digunakan untuk kasus ini jika mempertimbangkan jumlah perbandingan karakter, karena algoritma ini memerlukan jumlah perbandingan karakter yang sedikit
- Jika String yang akan dicari pattern-nya cenderung pendek, maka berdasarkan hasil percobaan ini, algoritma KMP lebih cocok digunakan, karena algoritma ini memakan waktu yang paling sedikit dan jumlah perbandingan yang paling sedikit

VI. UCAPAN TERIMA KASIH

Penulis ingin mengucapkan puji syukur kepada Allah SWT karena telah memberikan kemampuan dan waktu untuk menyelesaikan makalah ini dengan baik dan tepat waktu . Penulis juga ingin mengucapkan terima kasih kepada Bapak Rinaldi dan Ibu Ulfa yang telah mengajarkan dasar dasar teori yang penulis butuhkan untuk makalah ini . Penulis juga ingin mengucapkan terima kasih kepada kedua dosen yang telah membimbing dan menemani kami selama satu semester ini dan mengajarkan kami semua pelajaran Strategi

Algoritma yang sangat berguna kedepannya dan telah membentangkan pengetahuan yang berharga bagi penulis .

REFERENCES

- [1] <http://www.materisma.com/2014/10/penjelasan-dna-rna-sintesis-protein-kode-genetika.html> (Akses tanggal 1 Mei 2015)
- [2] <http://www.biologi-sel.com/2012/06/sintesis-protein-dan-kode-genetik.html> (Akses tanggal 1 Mei 2015)
- [3] <http://www.informasi-pendidikan.com/2013/01/perbedaan-dna-dan-rna.html> (Akses tanggal 1 Mei 2015)
- [4] http://ilmu-pengetahuan-unik.blogspot.com/2013/02/asam-amino_7883.html (Akses tanggal 1 Mei 2015)
- [5] <https://www.fmi.uni-sofia.bg/> (Akses tanggal 2 Mei 2015)
- [6] <http://stackoverflow.com> (Akses tanggal 1 – 2 Mei 2015)
- [7] Munir,Rinaldi,*Strategi Algoritma*, Informatika Bandung, 2009
- [8] Slide Perkuliahan Strategi Algoritma Semester 2 2014/2015

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, 2 Mei 2015

Pratama Nugraha Damanik
13513001