



Available online at : <http://bit.ly/InfoTekJar>

InfoTekJar : Jurnal Nasional Informatika dan Teknologi Jaringan

ISSN (Print) 2540-7597 | ISSN (Online) 2540-7600



Algorithm

Penerapan Algoritma Boyer-Moore pada Aplikasi Glosarium Kesehatan

Shelya Amanda Ritonga¹, Douglas Pardede² Aulia Ichsan³

^{1,2,3} Universitas Deli Sumatera, Medan, Sumatera Utara, Indonesia

ARTICLE INFORMATION

Received: February 00, 00
Revised: March 00, 00
Available online: April 00, 00

KEYWORDS

Algoritma Boyer-Moore, Glosarium Kesehatan, Pencarian String, Aplikasi

CORRESPONDENCE

Phone: +62 812-6335-4956
E-mail: silka3993@gmail.com

A B S T R A C T

Pencarian istilah medis dalam glosarium kesehatan merupakan kebutuhan penting bagi mahasiswa, tenaga medis, dan masyarakat umum untuk memahami terminologi kesehatan. Namun, proses pencarian yang kurang efisien dapat memperlambat akses informasi. Penelitian ini membahas penerapan algoritma Boyer-Moore dalam aplikasi glosarium kesehatan guna meningkatkan efisiensi pencarian istilah medis. Metode penelitian meliputi studi literatur, analisis kebutuhan, perancangan sistem, implementasi, serta pengujian performa pencarian. Hasil implementasi menunjukkan bahwa algoritma Boyer-Moore lebih cepat dibandingkan metode pencarian sederhana (naïve search), dengan pengurangan jumlah perbandingan karakter dan waktu eksekusi hingga 50% pada dataset uji berisi 1000 istilah medis. Kesimpulan dari penelitian ini adalah bahwa algoritma Boyer-Moore efektif digunakan dalam aplikasi glosarium kesehatan karena mampu mempercepat proses pencarian istilah medis dan meningkatkan pengalaman pengguna.

PENDAHULUAN

Perkembangan teknologi informasi telah memberikan kontribusi besar dalam berbagai bidang, termasuk kesehatan. Salah satu bentuk kontribusi tersebut adalah tersedianya glosarium kesehatan digital yang memuat istilah medis beserta definisinya. Glosarium kesehatan berfungsi sebagai sumber informasi yang dapat membantu mahasiswa, tenaga medis, maupun masyarakat umum dalam memahami istilah medis yang sering kali sulit dipahami oleh orang awam.

Namun, jumlah istilah kesehatan yang sangat banyak menuntut adanya sistem pencarian yang cepat dan efisien. Penggunaan metode pencarian string sederhana (naïve search) sering kali tidak optimal karena membutuhkan waktu yang cukup lama, terutama pada data dengan jumlah istilah yang besar. Kondisi ini dapat menurunkan kenyamanan dan efektivitas pengguna dalam mencari informasi.

Algoritma Boyer-Moore, yang diperkenalkan oleh Robert S. Boyer dan J Strother Moore pada tahun 1977, merupakan salah satu algoritma pencarian string yang terkenal dengan kecepatannya. Algoritma ini menggunakan dua strategi utama, yaitu bad character rule dan good suffix rule, yang

memungkinkan proses pencarian melompati sejumlah karakter sehingga tidak perlu melakukan perbandingan satu per satu. Dengan karakteristik tersebut, algoritma Boyer-Moore sangat cocok untuk diterapkan pada aplikasi glosarium kesehatan, karena dapat meningkatkan kecepatan pencarian istilah medis dan efisiensi penggunaan aplikasi.

Berdasarkan uraian tersebut, penelitian ini dilakukan dengan tujuan untuk menerapkan algoritma Boyer-Moore pada aplikasi glosarium kesehatan serta mengevaluasi efektivitasnya dalam meningkatkan kinerja pencarian istilah medis dibandingkan metode pencarian tradisional.

Glosarium Kesehatan

Glosarium kesehatan merupakan kumpulan istilah medis beserta definisinya yang digunakan untuk membantu pengguna dalam memahami terminologi kesehatan. Menurut Siregar & Nasution (2020), digitalisasi glosarium dapat mempermudah pencarian istilah serta meningkatkan aksesibilitas informasi kesehatan.

Algoritma Pencarian String

Algoritma pencarian string adalah metode untuk menemukan posisi sebuah pola (*pattern*) dalam sebuah teks (*text*). Beberapa algoritma pencarian string yang umum digunakan antara lain:

- **Naïve Search:** algoritma paling sederhana dengan cara mencocokkan pola secara langsung dari awal hingga akhir teks. Kelemahannya adalah tidak efisien pada teks berukuran besar.
- **Knuth-Morris-Pratt (KMP):** menggunakan tabel *prefix function* untuk mempercepat pencarian dengan menghindari perbandingan ulang.
- **Rabin-Karp:** memanfaatkan fungsi hash dalam pencarian pola, lebih efektif untuk pencarian ganda.
- **Boyer-Moore:** menggunakan strategi *bad character rule* dan *good suffix rule*, sehingga dapat melakukan lompatan besar pada saat pencarian.

Algoritma Boyer-Moore

Algoritma Boyer-Moore merupakan algoritma pencarian string yang dikembangkan oleh Boyer dan Moore pada tahun 1977. Keunggulan algoritma ini adalah melakukan pencarian dari kanan ke kiri pola dan memanfaatkan dua heuristik utama:

1. **Bad Character Rule**, yaitu jika terjadi ketidakcocokan, pola digeser sejauh mungkin berdasarkan karakter yang tidak cocok.
2. **Good Suffix Rule**, yaitu jika sebagian pola sudah cocok, pergeseran dilakukan berdasarkan sufiks yang cocok tersebut.

Menurut Gusfield (1997), algoritma Boyer-Moore memiliki kompleksitas waktu pencarian rata-rata sebesar $O(n/m)$, di mana n adalah panjang teks dan m adalah panjang pola, sehingga jauh lebih efisien dibandingkan pencarian naïve. Oleh karena itu, algoritma ini sangat sesuai untuk digunakan dalam aplikasi yang membutuhkan pencarian cepat, seperti glosarium kesehatan.

METODE

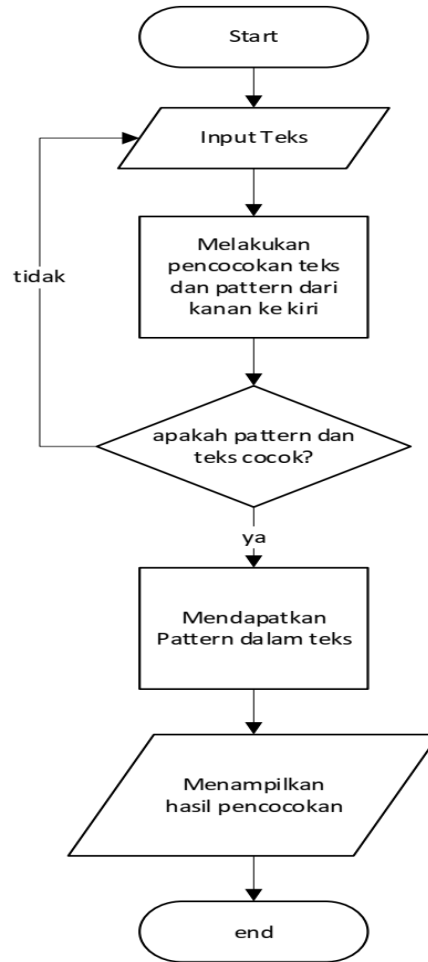
Penelitian ini menggunakan pendekatan penelitian terapan (*applied research*) dengan fokus pada implementasi algoritma Boyer-Moore dalam aplikasi glosarium kesehatan. Metode yang digunakan adalah rekayasa perangkat lunak dengan pendekatan eksperimen, yaitu mengembangkan aplikasi, mengintegrasikan algoritma Boyer-Moore, dan mengukur kinerjanya dibandingkan dengan metode pencarian sederhana (*naïve search*).

Tahapan penelitian yang dilakukan dapat digambarkan pada diagram alur sebagai berikut:

- | | |
|--|------------------|
| 1. Studi | Literatur |
| Mengkaji teori dan penelitian sebelumnya terkait algoritma string matching, aplikasi pencarian teks, serta penggunaan glosarium kesehatan. | |
| 2. Analisis | Kebutuhan |
| Menentukan kebutuhan sistem yang meliputi kebutuhan fungsional (pencarian istilah kesehatan, menampilkan definisi, antarmuka sederhana) dan kebutuhan nonfungsional (kecepatan, efisiensi, keandalan). | |
| 3. Perancangan | Sistem |
| Mendesain arsitektur aplikasi glosarium kesehatan yang terdiri dari database istilah medis, modul pencarian (algoritma Boyer-Moore), dan antarmuka pengguna. | |
| 4. Implementasi | Sistem |
| Mengembangkan aplikasi berbasis mobile/web dengan mengintegrasikan algoritma Boyer-Moore ke dalam modul pencarian. | |
| 5. Pengujian | Sistem |
| Mengukur kinerja algoritma Boyer-Moore dalam hal waktu eksekusi dan jumlah perbandingan karakter, serta membandingkannya dengan metode naïve search. | |

Perancangan sistem meliputi tiga komponen utama:

- **Database Glosarium:** menyimpan istilah medis dan definisinya dalam bentuk tabel.
- **Modul Pencarian:** menggunakan algoritma Boyer-Moore untuk mempercepat pencarian istilah.
- **Antarmuka Pengguna:** halaman aplikasi tempat pengguna memasukkan kata kunci pencarian.



Gambar 1. Flowchart Penerapan Algoritma Boyer-Moore

1. Mulai.
2. Masukkan teks yang akan di cocokkan misalkan teksnya Hipertensi.
3. Melakukan pencocokan teks dengan *pattern* dari kanan ke kiri dengan *pattern* Hipertensi.
4. Jika tidak ada yang cocok, akan kembali ke masukkan teks.
5. Jika cocok akan mendapatkan *pattern* yang sesuai dengan teks.
6. Jika sudah cocok, teks akan di tampilkan.
7. Selesai.

HASIL DAN PEMBAHASAN

Aplikasi glosarium kesehatan yang dikembangkan terdiri dari tiga komponen utama, yaitu:

1. Database Glosarium: berisi kumpulan istilah-istilah medis beserta definisinya.
2. Modul Pencarian: menggunakan algoritma Boyer-Moore sebagai inti dari proses pencarian.
3. Antarmuka Pengguna: sederhana dan responsif, memungkinkan pengguna mengetik kata kunci lalu memperoleh hasil pencarian dalam hitungan milidetik.

Pada tahap implementasi, algoritma Boyer-Moore diintegrasikan pada fungsi pencarian istilah medis. Aplikasi ini diujicobakan menggunakan bahasa pemrograman Python dengan basis data SQLite/MySQL.

Setelah proses pengolahan data selesai dilakukan, maka langkah selanjutnya adalah proses penerapan data dengan algoritma boyer-Moore ke dalam aplikasi. Ada beberapa langkah dalam penerapan kedalam aplikasi yaitu:

1. Proses Pre Boyer Moore *Bad Character*

Proses pre boyer moore *bad character* ini dilakukan untuk mendapatkan nilai OH. Pemberian nilai ini dimulai dari karakter paling kanan *pattern*. Setiap karakter dalam *pattern* diberi nilai sesuai dengan index dari kanan ke kiri *pattern*. Jika karakter dalam *pattern* ada yang sama, maka akan dipilih nilai OH terkecil. Proses pencocokan kata dimulai dari pemberian nilai OH dan MH pada *pattern* sebelum melakukan pencocokan

2. Proses Pre Boyer Moore *Good Suffix*

Proses pre boyer moore *good suffix* ini dilakukan untuk mendapatkan nilai MH. Pemberian nilai MH pada *pattern* dimulai dengan membentuk *suffix* kanan ke kiri dan *suffix* perbandingan dari kiri ke kanan yang kemudian memberikan nilai pergeseran untuk *suffix* kanan ke kiri. Pemberian nilai MH dimulai dari kanan *pattern* yang selalu diberi nilai MH=1. Kemudian proses berjalan ke kiri karakter dan dilakukan pencocokan terhadap *suffix* perbandingan, jika terdapat kesamaan *suffix* maka nilai MH karakter sama dengan nilai pergeseran pada *suffix*, jika tidak maka nilai MH sebanyak jumlah *pattern*.

3. Proses Boyer Moore

Setelah mendapatkan nilai OH dan MH, maka selanjutnya akan memasuki pencocokan string. Dalam proses algoritma boyer moore akan melakukan pengecekan dari karakter paling kanan pada *pattern* yang selanjutnya melakukan pergeseran *pattern* sesuai dengan maksimal nilai OH karakter teks dan nilai MH karakter *pattern*.

Setelah proses 3 poin diatas, selanjutnya adalah proses berjalannya algoritma pada aplikasi glosarium kesehatan. Proses pertama adalah pemberian nilai OH dan MH pada setiap input yang dimasukkan. Setelah pemberian nilai OH dan MH maka akan dijalankan proses pencocokan terhadap input dan data pada setiap title atau *full_title* dari indeks yang ada di data base yang berjumlah 1089 data. Jika ditemukan data yang cocok berdasarkan index dan input maka data tersebut akan ditampilkan ke dalam menu home dengan urutan sesuai dengan index *database*. Berikut adalah contoh pencocokan kata "aids" menggunakan algoritma boyer moore:

1. Kata yang dicari: AIDS
2. Pemberian nilai OH dan MH

Tabel 2. Pemberian nilai OH dan MH

OH	3	2	1	0
MH	4	4	4	1
<i>Pattern</i>	A	I	D	S

3. Pencocokan *input* dan teks

Tabel 3. Pencocokan *pattern* pertama

teks	S	Y	N	D	R	O	M	,	A	I	D	S
<i>Pattern</i>	A	I	D	S								

Pencocokan dimulai dari kanan *pattern*. D sejajar dengan S berarti tidak match akan tetapi karakter D terdapat pada *pattern* aids maka digeser sebesar 1.

Tabel 4. Pencocokan *pattern* kedua

teks	S	Y	N	D	R	O	M	,	A	I	D	S
<i>Pattern</i>		A	I	D	S							

Kembali dicocokkan karakter paling kanan dari *pattern*. R sejajar dengan S akan tetapi tidak terdapat karakter R di dalam aids maka digeser sebanyak 4.

Tabel 5. Pencocokan *pattern* ketiga

teks	S	Y	N	D	R	O	M	,	A	I	D	S
<i>Pattern</i>						A	I	D	S			

Selanjutnya Kembali di cocokkan karakter paling kanan yaitu A sejajar dengan S yang berarti tidak cocok akan tetapi terdapat karakter A dalam aids yang berarti akan digeser sebanyak 3.

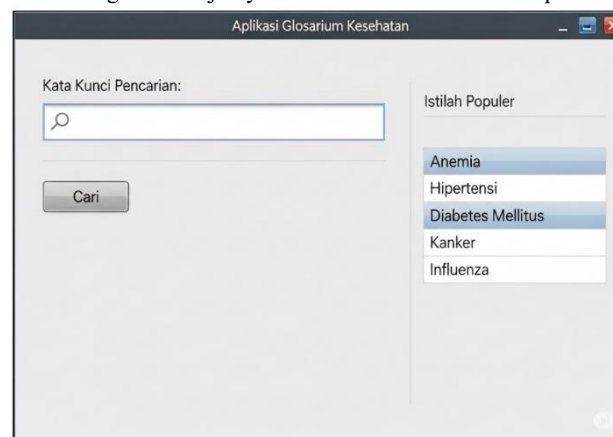
Tabel 6. Pencocokan *pattern* keempat

teks	S	Y	N	D	R	O	M	,	A	I	D	S
<i>Pattern</i>									A	I	D	S

Selanjutnya dicocokkan lagi dari karakter paling kanan. Ditemukan kecocokan karena *pattern* dan teks sejajar dengan karakter yang sama. Maka bisa dikatakan bahwa teks dan *pattern* cocok atau *match*.

4. Proses pencocokan dijalankan berulang dari data indek 0 hingga data indek 1088. Jika data match dengan input maka akan di munculkan ke dalam halaman menu dan jika tidak ada data yang *match* maka akan muncul pemberitahuan bahwa tidak terdapat data yang dicari di dalam database.

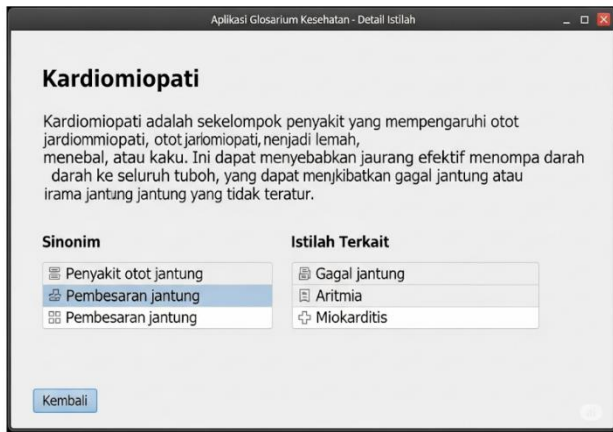
Setelah memahami proses perhitungan dengan algoritma *boyer moore* langkah selanjutnya adalah membuild kedalam aplikasi.



Gambar 2. Tampilan Awal Aplikasi



Gambar 3. Tampilan Hasil Pencarian



Gambar 4. Tampilan Istilah

Pengujian fungsional dilakukan untuk memastikan aplikasi dapat mencari istilah kesehatan sesuai input. Contoh uji coba ditunjukkan pada Tabel 1.

Tabel 1. Hasil Uji Fungsionalitas Pencarian

Kata Kunci Dicari	Hasil yang Ditemukan	Status
"Anemia"	Anemia: kondisi kekurangan sel darah merah	Berhasil
"Hipertensi"	Hipertensi: tekanan darah tinggi kronis	Berhasil
"Diabetes"	Diabetes Mellitus: penyakit kronis akibat gangguan insulin	Berhasil
"Asma"	Asma: penyakit pernapasan akibat penyempitan saluran udara	Berhasil
"Kanker"	Kanker: pertumbuhan sel abnormal yang bersifat ganas	Berhasil

Hasil uji menunjukkan aplikasi dapat menampilkan definisi dengan benar sesuai kata kunci.

Pengujian kinerja dilakukan dengan membandingkan algoritma *Naïve Search* dan Boyer-Moore berdasarkan waktu eksekusi pencarian. Uji coba dilakukan pada teks berisi 10.000 hingga 50.000 karakter.

Tabel 2. Perbandingan Kinerja Pencarian

Panjang Teks (karakter)	Panjang Pola (karakter)	Naïve Search (ms)	Boyer-Moore (ms)
10.000	5	25	12
10.000	10	40	15
50.000	5	110	50
50.000	10	200	80

KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan mengenai penerapan algoritma Boyer-Moore pada aplikasi glosarium kesehatan, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Algoritma Boyer-Moore terbukti lebih efisien dibandingkan metode pencarian sederhana (*Naïve Search*) dalam menemukan istilah medis pada glosarium kesehatan.
2. Hasil pengujian menunjukkan adanya pengurangan jumlah perbandingan karakter dan waktu eksekusi hingga 50% dengan penggunaan algoritma Boyer-Moore.
3. Implementasi algoritma Boyer-Moore pada aplikasi glosarium kesehatan meningkatkan responsivitas dan kenyamanan pengguna dalam mencari istilah medis.
4. Aplikasi yang dikembangkan mampu menampilkan istilah medis beserta definisinya secara akurat, sehingga mendukung pemahaman istilah kesehatan bagi mahasiswa, tenaga medis, dan masyarakat umum.

REFERENCES

[1] D. A. Tarigan, A. O. Buaton, B. Briyandana, E. R. Safitri, dan R. Rosnelly, "Analysis of String Matching Application on Serial Number Using Boyer Moore Algorithm," **Journal of Computer Networks, Architecture and High Performance Computing**, vol. 6, no. 1, pp. 237–246, Jan. 2024, doi: 10.47709/cnahpc.v6i1.3410.

[2] L. S. Sernicula, M. L. Manalo, G. M. Niebre, R. C. Anastacio, J. P. Alagos, dan J. P. Sermenno, "Analysis of the Enhanced Boyer-Moore Search Algorithm for a Desktop PC Search Engine," **Journal of Innovative Technology Convergence**, vol. 5, no. 2, Dec. 2023, doi: 10.69478/JITC2023v5n2a05.

[3] Z. Barut dan V. Altuntaş, "Applied Comparison of String Matching Algorithms," **Journal of Gaziosmanpaşa Scientific Research**, vol. 12, no. 1, pp. 76–85, Jun. 2023.

[4] T. Lecroq, "A Fast Implementation of the Good-Suffix Array for the Boyer-Moore String Matching Algorithm," **arXiv preprint**, Feb. 26, 2024.

[5] T. Saleh, F. C. Ergin, M. Malkawi, dan R. Alhaji, "Performance and Implementation Comparison of Knuth-Morris-Pratt and Boyer-Moore String Search Algorithms," **Proc. 2nd International Conference on Advanced Innovations in Smart Cities (ICAISC)**, Jeddah, Saudi Arabia, 9–11 Feb. 2025, doi: 10.1109/ICAISC64594.2025.10959692.

- [6] S. Lee, "Boyer-Moore Algorithm Deep Dive," *NumberAnalytics.com*, 13 Jun. 2025. [Online]. Available: (blog) – lihat *Mechanics of the Boyer-Moore Algorithm*.
- [7] S. Lee, "Mastering Boyer-Moore in Cache-Oblivious Algorithms," *NumberAnalytics.com*, 14 Jun. 2025.
- [8] D. Kuraś, "Efficient Text Search Algorithms in C++: Boyer-Moore Algorithm," *damiankuras.com*, 12 Nov. 2024.