Informatics

ARMARI FAMILY FOUR

Artikel

Sistem Pencarian Data Karyawan Menggunakan Algoritma Boyer Moore

Nasib Marbun

Teknologi Rekayasa Komputer Grafis, Politeknik Cendana; Medan; Indonesia; marbunnasib93@gmail.com

Abstrak: Pencarian data dalam sistem informasi kepegawaian menjadi elemen penting untuk menunjang efisiensi manajemen sumber daya manusia. Salah satu permasalahan yang sering dihadapi adalah lambatnya proses pencarian data karyawan, terutama dalam database berukuran besar. Artikel ini membahas implementasi algoritma Boyer Moore dalam pengembangan sistem pencarian data karyawan berbasis teks. Algoritma ini dikenal dengan efisiensinya dalam pencocokan string karena melakukan pergeseran lebih besar dibanding metode konvensional. Hasil analisis menunjukkan bahwa algoritma Boyer Moore terbukti mampu melakukan pencarian string dengan lebih cepat dan efisien dibandingkan metode pencocokan karakter satu per satu, khususnya pada kasus di mana pattern memiliki struktur karakter yang unik dan tidak sering berulang dalam teks.

Kata Kunci: Sistem Informasi Karyawan, Algoritma Boyer Moore, Pencarian Teks, Pencocokan String

1. Pendahuluan

Seiring dengan pesatnya pertumbuhan data karyawan dalam sistem perusahaan, kebutuhan akan metode pencarian informasi yang cepat, akurat, dan efisien menjadi semakin mendesak. Dalam lingkungan kerja yang semakin digital dan dinamis, data karyawan dapat mencakup berbagai informasi penting seperti identitas, riwayat pekerjaan, kualifikasi, performa, serta keterlibatan dalam proyek tertentu. Pertambahan jumlah data dari waktu ke waktu menyebabkan kompleksitas dan volume data menjadi semakin besar, sehingga metode pencarian konvensional tidak lagi memadai untuk menangani kebutuhan pencarian dalam skala besar.

Metode pencarian tradisional seperti pencarian linear atau naïf cenderung melakukan pemeriksaan satu per satu terhadap setiap elemen dalam kumpulan data. Meskipun sederhana, pendekatan ini memiliki kelemahan dalam hal efisiensi, terutama ketika jumlah data yang harus dicari sangat besar. Hal ini dapat menyebabkan proses pencarian menjadi lambat dan tidak efektif, yang pada akhirnya dapat menghambat pengambilan keputusan dan operasional perusahaan secara keseluruhan.

Untuk menjawab tantangan tersebut, diperlukan penerapan algoritma *string matching* yang lebih canggih dan efisien. Salah satu pendekatan yang diusulkan adalah penggunaan algoritma *Boyer Moore* sebagai solusi optimal dalam sistem pencarian data karyawan. Algoritma ini pertama kali dikembangkan oleh *Robert S. Boyer* dan *J Strother Moore* pada tahun 1977 [1], [2]. Hingga kini, algoritma *Boyer Moore* dikenal luas sebagai salah satu algoritma pencocokan string tercepat dan paling efisien yang pernah dikembangkan [3], [4].

Keunggulan utama dari algoritma *Boyer Moore* terletak pada kemampuannya untuk melewatkan sebagian besar karakter dalam teks saat proses pencarian berlangsung. Alih-alih membandingkan setiap karakter satu per satu seperti pada pencarian linear, algoritma ini menggunakan dua aturan utama, yaitu *bad character rule* dan *good suffix rule*, untuk menentukan sejauh mana pola pencarian dapat digeser setelah terjadi ketidakcocokan. Dengan strategi ini, *Boyer Moore* mampu mempercepat proses pencarian secara signifikan, terutama ketika digunakan pada teks yang panjang atau ketika pola pencarian memiliki panjang tertentu [5], [6].

Riwayat Artikel:
Diajukan : 02-06-2025
Direvisi : 11-06-2025
Diterima : 19-07-2025
Diterbitkan : 04-08-2025

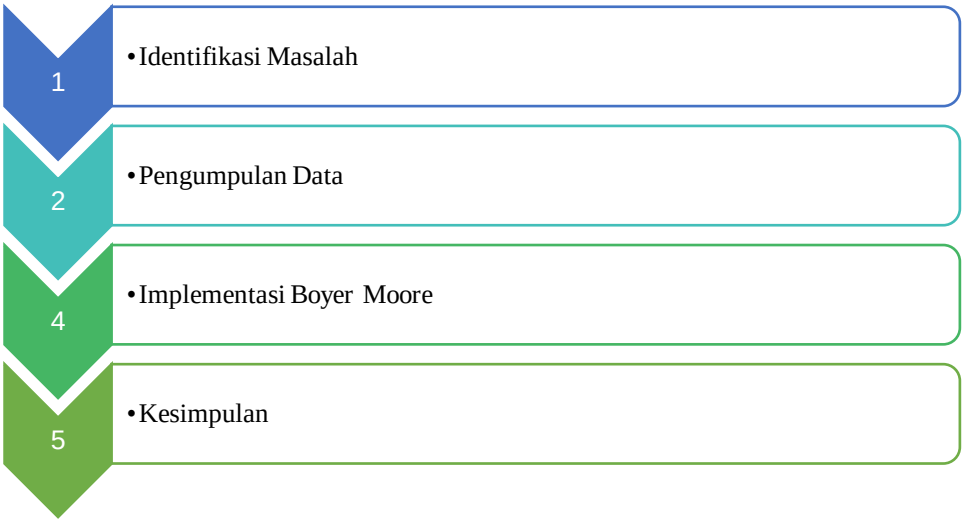
Hak Cipta: © 2025 oleh penulis.

Artikel ini adalah artikel akses terbuka yang didistribusikan di bawah syarat dan ketentuan Creative Commons Attribution-ShareAlike 4.0 [creativecommons.org/licenses/by-sa/4.0/]

2. Metodologi

2.1 Tahapan Penelitian

Tahapan penelitian ini dapat ditunjukkan pada Gambar 1 di bawah ini:



Gambar 1. Tahapan Penelitian

2.2 String Matching

String matching melibatkan pencarian satu atau beberapa contoh urutan karakter tertentu, yang dikenal sebagai pola, dalam urutan yang lebih besar yang disebut teks. Teknik ini diterapkan secara luas di berbagai bidang seperti mesin pencari, pengenalan pola, dan pemrosesan teks. Teknik ini memainkan peran penting dalam memungkinkan komputer mengidentifikasi pola, mengekstrak informasi yang relevan, dan memfasilitasi tugas-tugas seperti pencarian data, penyaringan, dan analisis teks. Baik dalam pencarian kata kunci sederhana maupun aplikasi pembelajaran mesin yang kompleks, algoritme *string matching* membantu memastikan pemrosesan data tekstual yang akurat dan efisien. Pentingnya algoritma ini mencakup berbagai bidang, mulai dari ilmu komputer hingga pemrosesan bahasa alami dan bioinformatika [7], [8], [9], [10], [11].

2.3 Boyer Moore

Boyer Moore merupakan salah satu algoritma *string matching* yang diperkenalkan oleh dua ilmuwan, yaitu *Robert S. Boyer* dan *J. Strother Moore* (1997). Algoritma ini memiliki keunggulan dalam hal efisiensi pencarian, karena memulai proses pencocokan dari kanan pola (*pattern*), meskipun penjajaran awal pola tetap dilakukan dari kiri teks. Dalam mekanismenya, algoritma *Boyer Moore* memanfaatkan dua strategi pergeseran utama untuk mempercepat proses pencocokan, yaitu *good suffix shift* dan *bad character shift*.

1. *Good suffix shift* diterapkan ketika tidak ditemukan kecocokan pada karakter yang sedang dibandingkan, namun bagian dari pola yang sejajar dengan teks terdapat kembali di bagian lain dari pola. Pergeseran dilakukan berdasarkan *Occurrence Heuristic*.
2. *Bad character shift* digunakan ketika karakter dalam teks yang sedang dibandingkan tidak ada dalam pola. Pergeseran ditentukan berdasarkan *Mismatch Heuristic* atau panjang keseluruhan pola.

Dengan kedua strategi ini, algoritma *Boyer Moore* mampu menghindari pencocokan karakter yang tidak perlu, sehingga mempercepat proses pencarian *string* dalam teks [3], [12], [13], [14].

Secara keseluruhanm proses pencocokan *string* pada algoritma *boyer moore* dapat dijelaskan pada Pseudocode berikut ini:

BoyerMoore(text, pattern):
  m ← *length(pattern)*
  n ← *length(text)*

```
// Step 1: Preprocessing
badChar ← BuildBadCharacterTable(pattern)
goodSuffix ← BuildGoodSuffixTable(pattern)

// Step 2: Searching
s ← 0  // shift of the pattern with respect to text
while s ≤ (n - m):
   j ← m - 1

   // Move from right to left in pattern
   while j ≥ 0 and pattern[j] = text[s + j]:
      j ← j - 1

   if j < 0:
      print("Pattern found at index", s)
      s ← s + goodSuffix[0]  // Shift pattern by goodSuffix rule
   else:
      shiftBadChar ← j - badChar[text[s + j]]
      shiftGoodSuffix ← goodSuffix[j]
      s ← s + max(1, max(shiftBadChar, shiftGoodSuffix))
```

3. Hasil dan Pembahasan

Pada penelitian ini, contoh studi kasus pencarian data karyawan menggunakan *Boyer Moore* dilakukan pada *text* "KARMEN MARBUN" yang dicocokkan dengan *string* "MARBUN". Adapun hasil analisis penerapan *Boyer Moore* dalam studi kasus tersebut dalam terlihat pada uraian berikut:

Tabel 1. *Occurence Heuristic (OH)* dan *Mismatch Heuristic (MH)*

Indeks	**0**	**1**	**2**	**3**	**4**	**5**
Pattern	M	A	R	B	U	N
OH	5	4	3	2	1	0
MH	6	6	6	6	6	1

1. Tahap Pencocokan *String* ke-1
   Pada tahapan pencocokan *string* ke-1, string N pada *pattern* dicocokan dengan *string* N pada Text yang sejajar dengan indeks 5. Pada pencocokan *string* pertama ini ditemukan kecocokan antara string yang sejajar dengan indeks 5 tersebut, sehingga dilakukan pencocokan selanjutnya antara *string* U dengan *string*. Namun pada pencocokan tersebut terjadi ketidakcocokan dan *string* E tidak ada di dalam *pattern*, sehingga dilakukan pergeseran *pattern* ke arah kanan sebanyak 6 langkah (sesuai dengan jumlah *string pattern*).

Tabel 2. Tahap Pencocokan String ke-1

Indeks	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Text	K	A	R	M	E	N		M	A	R	B	U	N
Pattern	M	A	R	B	U	N							

2. Tahap Pencocokan *String* ke-2
   Pada tahapan pencocokan *string* ke-2, string N pada *pattern* dicocokan dengan *string* U pada Text yang sejajar dengan indeks 11. Pada pencocokan *string* kedua ini ditemukan ketidakcocokan antara string yang sejajar dengan indeks 11 tersebut tetapi *string* U ada di dalam *pattern*, sehingga dilakukan pergeseran *pattern* ke arah kanan sebanyak 1 langkah (sesuai dengan jumlah OH *string U*).

Tabel 3. Tahap Pencocokan String ke-1

Indeks	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Text	K	A	R	M	E	N		M	A	R	B	U	N
Pattern							M	A	R	B	U	N	

3. Tahap Pencocokan *String* ke-3
   Pada tahapan pencocokan *string* ke-3, string N pada *pattern* dicocokan dengan *string* N pada Text yang sejajar dengan indeks 12. Pada pencocokan *string* kedua ini ditemukan kecocokan antara string yang sejajar dengan indeks 12 tersebut, sehingga dilakukan pencocokan selanjutnya antara *string* U dengan *string* U yang sejajar dengan indeks 11 hingga *string* M dengan *string* M yang sejajar dengan indeks 7, dan pada seluruh pencocokan *string* yang dilakukan terjadi kecocokan. Dikarenakan seluruh *string pattern* telah memiliki kecocokan dengan string *text*, maka proses pencocokan *string* diberhentikan.

Tabel 4. Tahap Pencocokan String ke-1

Indeks	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Text	K	A	R	M	E	N		M	A	R	B	U	N
Pattern								M	A	R	B	U	N

Berdasarkan uraian di atas, dengan kemampuan pergeseran yang adaptif berdasarkan karakter mismatch, algoritma Boyer Moore dapat diterapkan secara luas dalam sistem pencarian berbasis teks untuk meningkatkan efisiensi pencarian dan mengurangi waktu pemrosesan. Dalam konteks manajemen data karyawan, algoritma ini mendukung pencarian data yang cepat dan akurat, terlebih jika dataset berukuran besar.

4. Kesimpulan

Berdasarkan hasil studi kasus pencarian data karyawan menggunakan algoritma Boyer Moore pada teks “KARMEN MARBUN” dengan string pencarian “MARBUN”, dapat disimpulkan bahwa algoritma ini efektif dalam melakukan pencocokan string dengan pendekatan berbasis heuristik, yaitu Occurrence Heuristic (OH) dan Mismatch Heuristic (MH).

Dalam proses pencocokan, terjadi tiga tahap pergeseran sebelum ditemukan kecocokan penuh antara pattern dan bagian dari teks. Pada tahap awal, ketidaksesuaian karakter menyebabkan pergeseran berdasarkan nilai heuristik, yang menunjukkan efisiensi dalam mengurangi jumlah perbandingan karakter. Pada tahap ketiga, seluruh karakter pattern berhasil dicocokkan secara tepat dengan bagian teks, yang menandakan keberhasilan algoritma dalam menemukan pola yang dicari.

Secara keseluruhan, algoritma Boyer Moore terbukti mampu melakukan pencarian string dengan lebih cepat dan efisien dibandingkan metode pencocokan karakter satu per satu, khususnya pada kasus di mana pattern memiliki struktur karakter yang unik dan tidak sering berulang dalam teks.

Daftar Pustaka

[1] P. Aigbe and E. Nwelih, “Analysis and Performance Evaluation of Selected Pattern Matching Algorithms,” *NIPES - J. Sci. Technol. Res.*, vol. 3, no. 2, pp. 64–70, 2021, doi: 10.37933/nipes/3.2.2021.7.

[2] A. Azhar, N. Marbun, S. Aripin, and E. Buulolo, “Implementasi Algoritma Horspool Pada Aplikasi Istilah Fashion,” *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 3, no. 1, pp. 549–551, 2019, doi: 10.30865/komik.v3i1.1641.

[3] T. Raita, “Tuning the boyer-moore-horspool string searching algorithm,” *Softw. Pract. Exp.*, vol. 22, no. 10, pp. 879–884, Oct. 1992, doi: 10.1002/spe.4380221006.

[4] C. B. Setiawan, “Penerapan dan Perbandingan Algoritma String Matching pada Aplikasi UUD 1945 dan UU di Indonesia,” *Jsika*, vol. 4, pp. 1–7, 2019.

[5] L. Sernicula, M. Manalo, G. A. Niebre, R. Anastacio, J. Alagos, and J. Sermeno, “Analysis of the Enhanced Boyer-Moore Search Algorithm for a Desktop PC Search Engine,” *J. Innov. Technol. Converg.*, vol. 5, no. 2, pp. 45–56, Dec. 2023, doi: 10.69478/JITC2023v5n2a05.

[6] “Penerapan Web Semantik Berdasarkan Ontologi Pada Pencarian Judul Skripsi Dengan Algoritma Boyer-Moore,” *J. Ilm. Komputasi*, vol. 23, no. 2, Jun. 2024, doi: 10.32409/jikstik.23.2.3587.

[7] N. Marbun, I. J. Sinaga, A. Azhar, A. Manik, and S. B. F. Ginting, “Penerapan Algoritma Levenshtein dalam Pencarian Arti Istilah Penelitian,” *Pros. SINTAKS 2019*, vol. 1, no. 1, pp. 51–55, 2019.

[8] S. Kayan and L. Gaol, “Implementasi Knuth-Morris-Pratt (KMP) Untuk Pencarian Tempat Wisata,” vol. 1, no. 2, pp. 57–62, 2024.

[9] D. Cantone, S. Faro, and A. Pavone, "Approximate String Matching with Non-Overlapping Adjacent Unbalanced Translocations," *Mathematics*, vol. 13, no. 13, pp. 1–28, 2025, doi: 10.3390/math13132103.

[10] M. S. Kasttet, A. Lyhyaoui, D. Zbakh, A. Aramja, and A. Kachkari, "Toward Effective Aircraft Call Sign Detection Using Fuzzy String-Matching between ASR and ADS-B Data," *Aerospace*, vol. 11, no. 1, 2024, doi: 10.3390/aerospace11010032.

[11] M. Çelebi and U. Yavanoğlu, "Accelerating Pattern Matching Using a Novel Multi-Pattern-Matching Algorithm on GPU," *Appl. Sci.*, vol. 13, no. 14, 2023, doi: 10.3390/app13148104.

[12] A. K. Hidayah, U. Juhardi, R. Toyib, and N. A. Wijaya, "Designing an Android-Based Bisindo Dictionary Application Using the Boyer Moore Method," *J. Komputer, Inf. dan Teknol.*, vol. 2, no. 2, pp. 553–560, 2022, doi: 10.53697/jkomitek.v2i2.970.

[13] R. Kristianto Hondro, "Implementasi Algoritma Boyer Moore Pada Website Pencarian Jasa Servis Drone," *FIMERKOM J. Inf. Syst. Technol.*, vol. 1, no. 1, pp. 1–5, 2024.

[14] F. Fauzi Alvianda and Y. Sumaryana, "PerbandinganAlgoritma Brute Force Dengan Boyer-Moore Pada Aplikasi Pencarian Kerja Berbasis Web," *Inf.* , vol. 87, no. 1, pp. 87–99, 2023.