***APLIKASI PENCARIAN CEPAT KATA KUNCI PADA DOKUMEN DIGITAL BERBASIS STRING MATCHING***

**Muhammad Rizal Haris\*1, Agung Adi Rangga2**

1,2Department of Informatics, Universitas Muhammadiyah Makassar, Makassar, 90221, Indonesia

Email: 1105841103223@student@unismuh.ac.id, 2105841102123@student@unismuh.ac.id

\*Penulis Korespondensi

(Naskah masuk: dd mmm yyyy, diterima untuk diterbitkan: dd mmm yyyy)

**Abstrak**

*Pertumbuhan pesat dokumen digital berbasis teks, khususnya berkas PDF, telah meningkatkan permintaan akan mekanisme pencarian kata kunci yang efisien dan akurat yang mampu beroperasi pada konten teks berskala besar. Studi ini memaparkan desain dan implementasi aplikasi pencarian kata kunci PDF yang didasarkan pada algoritma pencocokan string eksak klasik, yaitu Naive, Knuth–Morris–Pratt (KMP), dan Boyer–Moore. Sistem yang diusulkan melakukan ekstraksi teks, prapemrosesan, dan pencocokan kata kunci dalam kerangka aplikasi terpadu, memungkinkan perbandingan kinerja yang transparan dan adil antar algoritma. Evaluasi eksperimental dilakukan dengan menjalankan ulang pada dokumen PDF berbasis teks nyata, dengan waktu eksekusi diukur sebagai metrik kinerja utama. Hasil menunjukkan perbedaan kinerja yang jelas antar algoritma: Boyer–Moore secara konsisten mencapai waktu eksekusi tercepat, KMP menunjukkan kinerja stabil dan dapat diprediksi, sedangkan Naive menunjukkan ketidakefisienan yang signifikan saat diterapkan pada teks yang lebih panjang. Meskipun ada perbedaan ini, semua algoritma menghasilkan hasil pencarian kata kunci yang identik dan akurat. Temuan ini menegaskan bahwa algoritma pencocokan string klasik tetap efektif dan relevan untuk aplikasi pengambilan dokumen saat dipilih dengan tepat. Selain itu, studi ini menetapkan dasar empiris yang dapat mendukung perluasan di masa depan, termasuk pencarian multi-kata kunci, pencocokan string aproximatif, pengambilan berbasis indeks, dan teknik percepatan komputasi untuk koleksi dokumen berskala besar.*

**Kata Kunci**: *Pencocokan String Tepat, Algoritma Naif, Knuth–Morris–Pratt, Boyer–Moore, Pencarian Dokumen Digital, Pengambilan Teks PDF*

***KEYWORD QUICK SEARCH APPLICATION ON STRING-BASED DIGITAL DOCUMENTS***

**Abstract**

*The rapid growth of text-based digital documents, particularly PDF files, has increased the demand for efficient and accurate keyword search mechanisms capable of operating on large-scale text content. This study describes the design and implementation of PDF keyword search applications based on classic exact-string matching algorithms, namely Naive, Knuth–Morris–Pratt (KMP), and Boyer–Moore. The proposed system performs text extraction, preprocessing, and keyword matching within a unified application framework, allowing for transparent and fair performance comparisons between algorithms. Experimental evaluations were conducted by rerunning on real text-based PDF documents, with execution time measured as a key performance metric. The results show clear performance differences between the algorithms: Boyer–Moore consistently achieves the fastest execution times, KMP shows stable and predictable performance, while Naive shows significant inefficiencies when applied to longer text. Despite these differences, all algorithms produce identical and accurate keyword search results. These findings confirm that classic string-matching algorithms remain effective and relevant for document retrieval applications when appropriately selected. In addition, the study establishes an empirical basis that could support future expansion, including multi-keyword searches, approximate string matching, index-based retrieval, and computational acceleration techniques for large-scale document collections.*

**Keywords**: *Exact String Matching, Naive Algorithm, Knuth–Morris–Pratt, Boyer–Moore, Digital Document Search, PDF Text Retrieval*

# PENDAHULUAN

Meningkatnya penggunaan dokumen digital berbasis teks—termasuk berkas PDF, laporan institusional, dan arsip akademik—mendorong kebutuhan akan sistem temu-kembali informasi yang mampu melakukan pencarian secara cepat dan akurat terhadap konten tekstual, terutama ketika jumlah dokumen terus berkembang dan metode pencarian manual tidak lagi efisien untuk mendukung aktivitas akademik maupun operasional (Flynn & Olukoya, 2025), (Rocco et al., 2026). Pada sistem pencarian berbasis teks, algoritma string matching memegang peranan utama dalam memastikan apakah suatu pattern atau kata kunci benar-benar muncul pada text yang diperoleh dari hasil ekstraksi dokumen, sehingga keandalan hasil pencarian sangat dipengaruhi oleh ketepatan mekanisme pencocokan yang diterapkan. Di samping ketepatan hasil, aspek efisiensi komputasi—khususnya yang berkaitan dengan waktu eksekusi dan intensitas perbandingan karakter—menjadi pertimbangan penting karena aplikasi pencarian dokumen dituntut untuk memberikan respons secara interaktif dan real-time; dengan demikian, pengujian efisiensi algoritma pencocokan string merupakan elemen kunci dalam pengembangan sistem pencarian kata kunci pada dokumen digital berbasis teks (Parvez et al., 2022).

Meskipun jumlah dokumen digital berbasis teks terus meningkat, proses pencarian informasi di dalamnya masih banyak bergantung pada pencarian manual atau mekanisme dasar yang kurang efisien ketika diterapkan pada teks berukuran besar, sehingga menurunkan responsivitas sistem dan membebani pengguna. Pendekatan pencocokan karakter secara berurutan memiliki kompleksitas waktu yang tinggi dan kurang sesuai untuk kebutuhan pencarian yang bersifat interaktif (Dondi et al., 2022). Selain itu, walaupun algoritma exact string matching klasik seperti Naive, Knuth–Morris–Pratt, dan Boyer–Moore telah dikaji secara luas dari sisi teoretis, implementasi dan evaluasi komparatif algoritma-algoritma tersebut dalam aplikasi pencarian dokumen digital yang transparan dan terukur masih relatif terbatas, khususnya dalam konteks biaya komputasi dan kinerja nyata sistem (Karcioglu & Bulut, 2021).

Walaupun algoritma exact string matching seperti Naive, Knuth–Morris–Pratt, dan Boyer–Moore telah lama menjadi fondasi dalam kajian desain dan analisis algoritma, sebagian besar penelitian yang ada masih berfokus pada aspek teoretis, seperti analisis kompleksitas, optimasi struktur data, atau pengembangan variasi algoritmik lanjutan. Studi-studi tersebut umumnya menekankan peningkatan performa melalui pendekatan algoritmik atau automata, dengan keterbatasan pada implementasi sistem pencarian dokumen yang bersifat praktis dan mudah direplikasi (Faro & Scafiti, 2023), (Faro & Scafiti, 2022). Selain itu, evaluasi eksperimental sering disajikan dalam bentuk metrik numerik semata, tanpa disertai visualisasi proses dan keluaran sistem yang dapat membantu pemahaman perilaku algoritma secara aplikatif. Kondisi ini menunjukkan adanya celah penelitian pada studi implementatif yang menyajikan perbandingan transparan algoritma string matching klasik dalam satu kerangka aplikasi pencarian dokumen digital.

Penelitian ini bertujuan untuk merancang dan mengimplementasikan sebuah aplikasi pencarian kata kunci pada dokumen digital berbasis teks dengan memanfaatkan algoritma exact string matching sebagai mekanisme utama pencocokan pola. Sistem yang dikembangkan mengintegrasikan tiga algoritma klasik, yaitu Naive, Knuth–Morris–Pratt, dan Boyer–Moore, ke dalam satu kerangka aplikasi yang sama sehingga memungkinkan perbandingan kinerja yang objektif dan konsisten. Fokus utama penelitian ini adalah mengevaluasi performa masing-masing algoritma berdasarkan waktu eksekusi dan hasil pencarian pada dokumen digital nyata, guna memperoleh pemahaman empiris mengenai efektivitas dan efisiensi algoritma string matching dalam konteks aplikasi pencarian dokumen.

Keunikan penelitian ini terletak pada penerapan langsung algoritma exact string matching klasik ke dalam sebuah aplikasi pencarian dokumen PDF berbasis teks, sehingga perilaku dan kinerja algoritma yang umumnya dibahas secara konseptual dapat diamati secara nyata dalam lingkungan sistem yang operasional. Berbeda dari penelitian yang menitikberatkan pada pengembangan algoritma lanjutan atau penerapan pada domain khusus, studi ini memfokuskan evaluasi kinerja pada data digital aktual berupa dokumen yang diunggah oleh pengguna, sehingga hasil pengujian merepresentasikan kondisi penggunaan yang realistis. Kontribusi penelitian ini diperkuat melalui penyajian keluaran sistem secara eksplisit, mencakup hasil pencarian, waktu eksekusi, serta visualisasi perbandingan algoritma, yang tidak hanya mendukung analisis performa tetapi juga meningkatkan transparansi dan pemahaman implementasi algoritma pencocokan string dalam aplikasi pencarian dokumen digital.

# TINJAUAN LITERATUR

**2.1** **Landasan Konseptual dan Teoritis**

Dalam permasalahan pencarian teks, string didefinisikan sebagai urutan simbol yang dibentuk dari suatu alfabet tertentu, di mana text merepresentasikan string berdimensi besar yang menjadi ruang pencarian, sedangkan pattern merupakan string yang lebih pendek dan berfungsi sebagai objek pencocokan. Permasalahan exact string matching berfokus pada identifikasi seluruh posisi kemunculan pattern di dalam text dengan tingkat kecocokan yang sepenuhnya identik, tanpa mengizinkan substitusi, penyisipan, maupun penghapusan karakter, sehingga setiap kecocokan ditentukan melalui kesetaraan simbol secara berurutan (Dondi et al., 2022).

Dari perspektif teoretis, permasalahan ini memiliki peran sentral dalam desain dan analisis algoritma karena kinerja pencarian sangat dipengaruhi oleh kompleksitas waktu dan ruang yang dihasilkan selama proses pencocokan. Kompleksitas waktu berkaitan dengan jumlah operasi perbandingan karakter terhadap panjang text dan pattern, sedangkan kompleksitas ruang mencerminkan kebutuhan memori tambahan untuk menyimpan informasi pendukung selama pencarian. Variasi karakteristik kompleksitas inilah yang mendasari pengembangan dan evaluasi berbagai algoritma exact string matching, serta menjadikannya fondasi penting dalam kajian algoritma pencarian teks dan sistem temu-kembali informasi (S. Kim et al., 2024).

**2.2 Landasan Metodologis**

Dari sudut pandang metodologis, Algoritma Naive String Matching merupakan pendekatan paling dasar dalam pencarian string yang bekerja dengan membandingkan karakter pattern terhadap text secara berurutan pada setiap kemungkinan posisi pergeseran. Proses pencocokan dilakukan dari kiri ke kanan hingga seluruh karakter cocok atau ditemukan ketidakcocokan, sehingga pada kasus terburuk algoritma ini dapat melakukan perbandingan karakter dalam jumlah yang besar ketika diterapkan pada teks berukuran panjang (Karcioglu & Bulut, 2021). Algoritma Knuth–Morris–Pratt (KMP) dikembangkan untuk mengatasi inefisiensi pendekatan Naive dengan memanfaatkan informasi kecocokan parsial melalui prefix function atau Longest Proper Prefix which is also Suffix (LPS). Dengan mekanisme ini, pergeseran pattern dapat dilakukan tanpa mengulang perbandingan karakter yang telah diketahui cocok, sehingga proses pencarian menjadi lebih efisien terutama pada teks dengan pola berulang (Nakashima et al., 2022), (S.-H. Kim & Cho, 2021). Berbeda dari pendekatan sebelumnya, algoritma Boyer–Moore menerapkan strategi pencocokan dari arah kanan ke kiri dan memanfaatkan heuristik bad character untuk menentukan besar pergeseran pola berdasarkan karakter yang tidak sesuai. Strategi ini memungkinkan pergeseran pola yang lebih jauh dalam satu langkah, sehingga dalam praktik sering menghasilkan performa yang lebih baik pada dokumen berukuran besar. Pendekatan Boyer–Moore juga menjadi dasar bagi berbagai pengembangan metode pencocokan string yang berorientasi pada optimasi efisiensi pencarian (Faro & Scafiti, 2023), (Faro et al., 2023).

**2.3 Sintesis Karya Terkait**

Berbagai penelitian menunjukkan bahwa algoritma pencocokan string telah diterapkan secara luas dalam sistem pencarian teks dan analisis pola pada data digital, terutama untuk mendukung kebutuhan temu-kembali informasi yang menuntut kecepatan dan ketepatan hasil. Sejumlah studi membedakan pendekatan pencarian berbasis brute force dengan metode yang telah dioptimalkan, di mana upaya peningkatan kinerja umumnya difokuskan pada pengurangan jumlah perbandingan karakter serta pemanfaatan informasi pencocokan sebelumnya untuk meningkatkan efisiensi, khususnya pada teks berukuran besar (Pikies & Ali, 2021), (Khan et al., 2023). Namun demikian, sebagian besar karya tersebut lebih menekankan pengembangan teknik lanjutan atau solusi spesifik domain, sehingga penerapan algoritma exact string matching klasik dalam konteks sistem pencarian dokumen umum masih relatif terbatas.

Di sisi lain, penelitian yang mengarah pada pendekatan advanced seperti semantic text matching dan model berbasis pembelajaran mendalam menunjukkan peningkatan akurasi pada skenario tertentu, tetapi sering kali memerlukan sumber daya komputasi yang lebih besar serta tidak secara langsung merepresentasikan perilaku algoritma pencocokan string klasik (Wang et al., 2024). Oleh karena itu, penelitian ini diposisikan sebagai studi implementatif komparatif yang secara eksplisit mengintegrasikan dan mengevaluasi algoritma Naive, Knuth–Morris–Pratt, dan Boyer–Moore dalam satu kerangka aplikasi pencarian dokumen digital, dengan penekanan pada transparansi proses, keluaran sistem, serta pengukuran performa yang dapat diamati secara empiris.

# METODOLOGI

**3.1 Kerangka Penelitian**

Kerangka kerja sistem yang diusulkan dalam penelitian ini dirangkum secara menyeluruh pada Gambar 1, yang merepresentasikan alur pemrosesan pencarian kata kunci pada dokumen digital berbasis PDF dengan memanfaatkan algoritma exact string matching. Alur tersebut menggambarkan proses end-to-end yang dimulai dari unggahan dokumen digital, dilanjutkan dengan ekstraksi dan pra-pemrosesan teks, hingga tahap konfigurasi kata kunci serta pemilihan algoritma pencarian yang digunakan dalam sistem. Penyusunan workflow ini bertujuan untuk memastikan bahwa setiap tahapan pemrosesan berjalan secara terstruktur dan konsisten, sehingga proses pencarian dapat dilakukan secara sistematis (Parvez et al., 2022).



Gambar 1. Alur Kerja Sistem Aplikasi Pencarian Kata Kunci PDF yang Diusulkan Berdasarkan Pencocokan String yang Tepat

Sebagaimana ditunjukkan pada Gambar 1, integrasi antara pemrosesan dokumen, mekanisme pencocokan string, dan pengukuran kinerja algoritma memungkinkan dilakukannya analisis komparatif yang transparan terhadap algoritma Naive, Knuth–Morris–Pratt, dan Boyer–Moore. Struktur alur kerja yang terdefinisi dengan jelas ini memastikan bahwa setiap tahap menghasilkan keluaran yang dapat diamati secara langsung, baik berupa hasil pencarian kata kunci maupun informasi efisiensi waktu eksekusi, sehingga secara langsung mendukung tujuan penelitian dalam mengevaluasi performa algoritma string matching pada aplikasi pencarian dokumen digital.

## 3.2 Deskripsi Kumpulan Data

Dataset yang digunakan dalam penelitian ini berupa dokumen digital berformat PDF berbasis teks (text-based PDF), sehingga konten tekstual dapat diekstraksi secara langsung tanpa memerlukan proses pengenalan karakter optik. Dataset bersifat dinamis karena dokumen diperoleh melalui mekanisme unggahan pengguna selama sistem beroperasi, yang memungkinkan variasi konten dan ukuran dokumen serta merepresentasikan kondisi penggunaan nyata pada aplikasi pencarian dokumen digital. Seluruh teks hasil ekstraksi diperlakukan sebagai text yang menjadi ruang pencarian bagi algoritma exact string matching, sedangkan kata kunci yang dimasukkan oleh pengguna berfungsi sebagai pattern dalam proses pencocokan, sejalan dengan praktik pemrosesan teks dan analisis pola pada sistem berbasis dokumen digital yang menangani data tidak terstruktur (Flynn & Olukoya, 2025).

Tabel 1, Tabel 2, dan Tabel 3 menyajikan hasil pengukuran waktu eksekusi algoritma Naive, Knuth–Morris–Pratt (KMP), dan Boyer–Moore pada proses pencarian kata kunci dalam dokumen PDF berbasis teks. Pengujian dilakukan melalui sepuluh kali eksekusi berulang (multiple runs) pada dataset yang sama untuk setiap algoritma, sehingga variasi waktu eksekusi yang diamati merefleksikan karakteristik performa intrinsik masing-masing metode pencocokan string. Penyajian waktu eksekusi per run ini bertujuan untuk memberikan gambaran kuantitatif yang rinci mengenai stabilitas dan efisiensi algoritma, sekaligus menjadi dasar bagi analisis komparatif performa yang dibahas pada subbagian berikutnya.

Tabel 1. Waktu Eksekusi Algoritma Naive per Eksekusi

|  |  |
| --- | --- |
| Run Pengujian | Waktu Pengujian (ms) |
| Percobaan 1 | 301.56 |
| Percobaan 2 | 342.04 |
| Percobaan 3 | 349.10 |
| ... | ... |
| Percobaan 10 | 352.54 |

Tabel 2. Waktu Eksekusi Algoritma Knuth–Morris–Pratt (KMP) per Eksekusi

|  |  |
| --- | --- |
| Run Pengujian | Waktu Pengujian (ms) |
| Percobaan 1 | 281.25 |
| Percobaan 2 | 285.84 |
| Percobaan 3 | 218.76 |
| ... | ... |
| Percobaan 10 | 224.11 |

Tabel 3. Waktu Eksekusi Algoritma Boyer–Moore per Eksekusi

|  |  |
| --- | --- |
| Run Pengujian | Waktu Pengujian (ms) |
| Percobaan 1 | 79.65 |
| Percobaan 2 | 93.49 |
| Percobaan 3 | 109.06 |
| ... | ... |
| Percobaan 10 | 160.61 |

## 3.3 Prosedur Pra-pemrosesan

Tahap pra-pemrosesan bertujuan untuk memastikan bahwa data teks berada dalam kondisi yang konsisten sebelum diproses oleh algoritma exact string matching. Proses ini diawali dengan ekstraksi konten tekstual dari dokumen PDF berbasis teks, di mana seluruh teks pada setiap halaman dokumen dikonversi dan digabungkan menjadi satu representasi text yang utuh. Selanjutnya, dilakukan normalisasi huruf dengan menerapkan pendekatan case-insensitive agar variasi penggunaan huruf kapital dan huruf kecil tidak memengaruhi proses pencocokan kata kunci. Pada tahap akhir, dilakukan pembersihan karakter non-teks, termasuk simbol yang tidak relevan, karakter khusus, serta spasi berlebih, sehingga algoritma pencarian string bekerja pada data teks yang bersih dan terstandarisasi, serta meminimalkan potensi kesalahan pencocokan akibat noise pada dokumen digital (Pikies & Ali, 2021).

## 3.4 Deskripsi Model

Model pencarian yang dikembangkan dalam penelitian ini mengadopsi tiga algoritma exact string matching klasik, yaitu Naive, Knuth–Morris–Pratt (KMP), dan Boyer–Moore, yang masing-masing merepresentasikan strategi pencocokan dan tingkat efisiensi komputasi yang berbeda. Dalam kerangka sistem, algoritma Naive digunakan sebagai pendekatan dasar yang melakukan pencocokan karakter secara berurutan pada setiap kemungkinan posisi pencarian, sementara KMP dan Boyer–Moore berperan sebagai algoritma pembanding yang menerapkan mekanisme optimisasi pencarian melalui pemanfaatan informasi kecocokan parsial serta strategi pergeseran pola yang lebih efisien (Karcioglu & Bulut, 2021). Ketiga algoritma tersebut diimplementasikan sebagai modul pencarian yang dapat dipilih secara fleksibel oleh pengguna, sehingga memungkinkan evaluasi dan perbandingan perilaku serta kinerja algoritma dalam lingkungan aplikasi yang sama. Uraian teknis mengenai detail implementasi, alur eksekusi, dan pengukuran performa disajikan secara terpisah pada bagian Results untuk menekankan analisis empiris dibandingkan aspek konseptual model (Faro & Scafiti, 2023).

## 3.5 Ruang Lingkup Algoritmik dan Kendala Desain

Ruang lingkup algoritmik pada penelitian ini secara sengaja dibatasi pada penerapan algoritma exact string matching klasik, yaitu Naive, Knuth–Morris–Pratt, dan Boyer–Moore, tanpa melibatkan pendekatan hybrid, teknik optimisasi lanjutan, maupun integrasi dengan model berbasis pembelajaran mesin. Batasan desain ini ditetapkan untuk menjaga kemurnian evaluasi karakteristik algoritma, sehingga perbedaan kinerja yang diamati sepenuhnya merefleksikan perilaku intrinsik masing-masing metode pencocokan. Dengan menghindari penambahan mekanisme optimisasi eksternal, analisis yang dilakukan menjadi lebih transparan dan terkontrol, khususnya dalam menilai efisiensi waktu eksekusi dan hasil pencarian, serta membedakan penelitian ini dari studi yang berfokus pada pengembangan metode pencocokan lanjutan atau automata kompleks (Faro & Scafiti, 2022).

## 3.6 Metrik Evaluasi

Evaluasi kinerja sistem pencarian dilakukan menggunakan metrik kuantitatif yang menitikberatkan pada aspek efisiensi komputasi dan efektivitas hasil pencocokan. Metrik utama yang digunakan adalah waktu eksekusi, yang diukur dalam milidetik untuk merepresentasikan durasi proses pencarian sejak algoritma dijalankan hingga hasil dikembalikan kepada pengguna, sehingga mencerminkan tingkat responsivitas sistem dalam skenario penggunaan interaktif. Selain itu, jumlah kemunculan kata kunci dicatat untuk menggambarkan keberhasilan pencocokan pattern di dalam text, sementara jumlah dokumen yang terdeteksi digunakan untuk menunjukkan cakupan hasil pencarian pada dokumen yang diproses. Kombinasi metrik tersebut memungkinkan perbandingan kinerja yang jelas dan objektif antara algoritma Naive, Knuth–Morris–Pratt, dan Boyer–Moore, sejalan dengan praktik evaluasi performa algoritma pencocokan string pada sistem nyata (Chhabra et al., 2025).

# HASIL DAN DISKUSI

## 4.1 Hasil Kinerja Kuantitatif

Bagian ini menyajikan hasil evaluasi kuantitatif terhadap performa algoritma Naive, Knuth–Morris–Pratt (KMP), dan Boyer–Moore dalam proses pencarian kata kunci pada dokumen PDF berbasis teks. Pengujian difokuskan pada pengukuran waktu eksekusi algoritma melalui sejumlah eksekusi berulang pada dataset yang sama, sehingga perbandingan performa dilakukan secara adil dan konsisten. Selain itu, hasil pencarian yang dihasilkan oleh masing-masing algoritma juga dianalisis untuk memastikan konsistensi keluaran sistem.

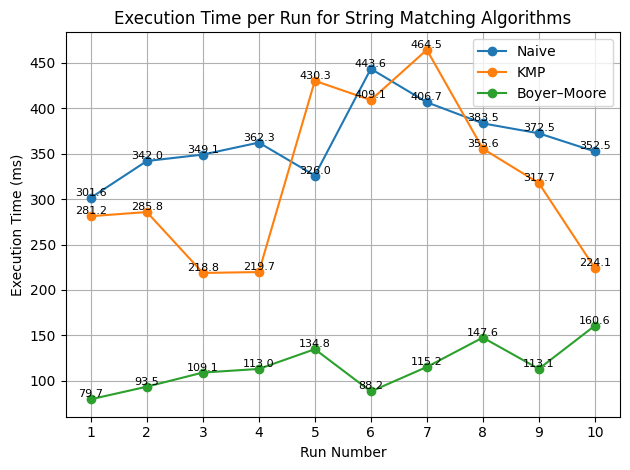
## 4.1.1 Waktu Eksekusi Naive, KMP, dan Boyer–Moore

Table 4 menyajikan ringkasan waktu eksekusi algoritma Naive, KMP, dan Boyer–Moore dalam satuan milliseconds, yang diperoleh dari sepuluh kali eksekusi pencarian kata kunci pada dokumen PDF yang sama. Nilai yang ditampilkan mencakup rata-rata waktu eksekusi, simpangan baku, serta jumlah pengujian, sehingga memberikan gambaran kuantitatif mengenai efisiensi dan stabilitas masing-masing algoritma.

Tabel 4. Waktu Eksekusi (ms) Algoritma Pencocokan String

|  |  |  |  |
| --- | --- | --- | --- |
| **Algoritma** | **Waktu Eksekusi Rata-rata (ms)** | **Standar Deviasi (ms)** | **Jumlah Perulangan** |
| Naive | 363.99 | 40.53 | 10 |
| KMP | 320.70 | 90.94 | 10 |
| Boyer–Moore | 115.47 | 25.83 | 10 |

Berdasarkan Tabel 4, terlihat adanya perbedaan karakteristik performa antar algoritma. Algoritma Naive menunjukkan waktu eksekusi rata-rata tertinggi, sedangkan Boyer–Moore menghasilkan waktu eksekusi paling rendah dan relatif konsisten. Sementara itu, KMP berada di antara keduanya dengan variasi waktu eksekusi yang lebih besar, mencerminkan sensitivitas algoritma terhadap pola dan distribusi teks. Untuk memberikan gambaran yang lebih rinci mengenai variasi waktu eksekusi pada setiap pengujian, Gambar 2 menyajikan visualisasi perubahan waktu eksekusi algoritma Naive, KMP, dan Boyer–Moore pada setiap run pengujian. Visualisasi ini memungkinkan analisis langsung terhadap fluktuasi performa algoritma selama proses pencarian berlangsung.



Gambar 2. Waktu eksekusi per eksekusi algoritme pencocokan string Naive, KMP, dan Boyer–Moore di sepuluh eksekusi berulang.

Secara keseluruhan, hasil kuantitatif pada Tabel 1 dan Gambar 2 menunjukkan perbedaan karakteristik kinerja yang konsisten antar algoritma string matching. Algoritma Boyer–Moore menghasilkan waktu eksekusi terendah dan variasi yang relatif stabil, terutama pada teks berukuran panjang, yang mencerminkan efektivitas strategi pergeseran pola dalam mengurangi jumlah perbandingan karakter. Temuan ini selaras dengan prinsip optimisasi pencarian berbasis struktur dan automata yang menekankan efisiensi pencocokan pola pada teks panjang (Faro & Scafiti, 2023).

Sebaliknya, algoritma Knuth–Morris–Pratt (KMP) memperlihatkan performa yang lebih stabil dibandingkan Naive, meskipun tidak selalu lebih cepat daripada Boyer–Moore. Stabilitas ini menunjukkan keunggulan KMP dalam menjaga konsistensi pencocokan melalui pemanfaatan informasi prefix, khususnya pada pola dengan struktur berulang. Algoritma Naive, di sisi lain, menunjukkan waktu eksekusi tertinggi dan fluktuasi performa yang lebih besar, yang mengindikasikan keterbatasan pendekatan pencocokan karakter secara berurutan tanpa mekanisme optimisasi.

Meskipun hasil penelitian ini menegaskan relevansi algoritma string matching klasik dalam konteks aplikasi pencarian dokumen digital, sejumlah studi menunjukkan bahwa peningkatan kinerja yang lebih signifikan dapat dicapai melalui optimisasi lanjutan, termasuk pemanfaatan akselerasi berbasis perangkat keras dan instruksi paralel tingkat rendah. Pendekatan tersebut terbukti mampu meningkatkan throughput pencarian secara substansial, namun juga menambah kompleksitas sistem dan kebutuhan sumber daya komputasi tambahan (Chhabra et al., 2025).

## 4.1.2 Hasil Pencarian Kata Kunci

Tabel 5 menyajikan hasil pencarian kata kunci yang diukur berdasarkan jumlah kemunculan kata kunci (Kemunculan kata kunci) dan jumlah dokumen yang terdeteksi oleh masing-masing algoritma string matching. Evaluasi ini bertujuan untuk memastikan bahwa perbedaan performa waktu eksekusi yang ditunjukkan pada subbagian sebelumnya tidak memengaruhi konsistensi hasil pencarian.

Tabel 5. Hasil Pencarian Kata Kunci

|  |  |  |
| --- | --- | --- |
| **Algoritma** | **Kemunculan Kata Kunci** | **Dokumen yang Terdeteksi** |
| Naive | 8 | 1 |
| KMP | 8 | 1 |
| Boyer–Moore | 8 | 1 |

Berdasarkan Tabel 5, seluruh algoritma-Naive, KMP, dan Boyer–Moore-menghasilkan jumlah kemunculan kata kunci dan jumlah dokumen terdeteksi yang identik. Hal ini menunjukkan bahwa ketiga algoritma memberikan hasil pencarian yang konsisten dan akurat pada dokumen PDF yang diuji, meskipun memiliki karakteristik waktu eksekusi yang berbeda.

## 4.2 Perbandingan dengan Literatur Sebelumnya

Temuan empiris dalam penelitian ini konsisten dengan teori kompleksitas waktu yang telah banyak dibahas dalam literatur algoritma string matching. Algoritma Boyer–Moore secara teoretis dikenal mampu mencapai efisiensi yang lebih baik pada praktik nyata, terutama pada teks panjang, sedangkan KMP dirancang untuk menghindari perbandingan ulang karakter melalui pemanfaatan struktur pola secara sistematis. Sebaliknya, algoritma Naive memiliki kompleksitas waktu yang lebih tinggi karena melakukan pencocokan ulang pada setiap kemungkinan posisi teks (Dondi et al., 2022), (Karcioglu & Bulut, 2021).

Implementasi pada dokumen PDF berbasis teks dalam penelitian ini memberikan validasi empiris terhadap karakteristik teoretis tersebut, sekaligus menunjukkan bahwa perbedaan performa algoritma klasik tetap dapat diamati secara nyata dalam konteks aplikasi pencarian dokumen. Dengan demikian, penelitian ini memperkuat temuan sebelumnya yang menekankan keunggulan algoritma dengan mekanisme optimisasi pencarian, serta menegaskan relevansi algoritma string matching klasik dalam sistem pencarian modern ketika diimplementasikan secara tepat (Faro & Scafiti, 2023), (Faro et al., 2023).

Selain pendekatan klasik, sejumlah penelitian terkini juga mengusulkan strategi pencocokan pola yang lebih kompleks, seperti order-preserving pattern matching dan simulasi automata dengan struktur data khusus, yang ditujukan untuk menangani variasi permasalahan pencarian pada skala dan karakteristik data tertentu. Meskipun pendekatan tersebut berpotensi meningkatkan performa pada konteks spesifik, kompleksitas desain dan kebutuhan komputasi tambahan yang menyertainya membedakan fokusnya dari penelitian ini, yang secara sengaja menitikberatkan pada evaluasi algoritma string matching klasik dalam lingkungan aplikasi yang sederhana, transparan, dan mudah direplikasi (Y. Kim et al., 2023), (Vaiwsri et al., 2024).

## 4.3 Implikasi Praktis dan Teoritis

Dari perspektif praktis, hasil penelitian ini menunjukkan bahwa aplikasi pencarian kata kunci berbasis algoritma string matching klasik dapat dimanfaatkan sebagai alat bantu pencarian dokumen digital yang efisien, transparan, dan mudah dipahami oleh pengguna. Penyajian informasi waktu eksekusi dan hasil pencarian secara langsung memungkinkan pengguna untuk mengamati perbedaan performa antar algoritma dan memilih pendekatan yang paling sesuai dengan kebutuhan pencarian dalam konteks operasional (Parvez et al., 2022).

Dari sisi teoretis dan edukatif, penelitian ini berkontribusi dalam menjembatani kesenjangan antara konsep desain dan analisis algoritma dengan implementasi nyata pada sistem berbasis aplikasi. Dengan mengintegrasikan algoritma Naive, KMP, dan Boyer–Moore dalam satu kerangka sistem yang dapat diamati secara langsung, studi ini memperkuat pemahaman konseptual mengenai perilaku algoritma string matching serta relevansinya dalam pemrosesan teks dan sistem pencarian informasi modern (Rocco et al., 2026).

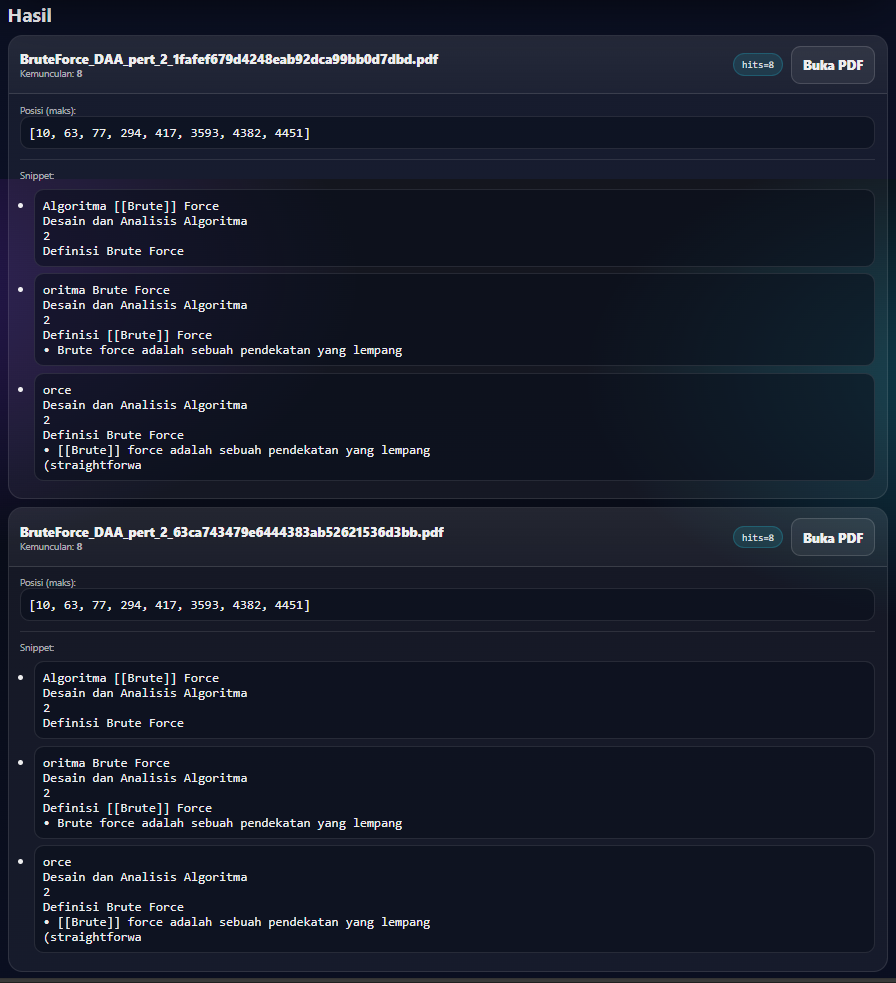
Selain itu, temuan dalam penelitian ini juga memiliki implikasi terhadap pengembangan sistem pencarian yang lebih efisien di masa mendatang. Sejumlah studi menunjukkan bahwa peningkatan throughput dan efisiensi pencarian dapat dicapai melalui optimisasi tingkat lanjut, seperti pemanfaatan struktur indeks khusus maupun akselerasi berbasis perangkat keras. Meskipun pendekatan tersebut berada di luar cakupan penelitian ini, hasil yang diperoleh dapat digunakan sebagai baseline empiris untuk membandingkan algoritma klasik dengan metode optimisasi lanjutan pada penelitian selanjutnya (Zhao et al., 2024).

## 4.4 Visualisasi dan Analisis Interpretatif

Gambar 3 menyajikan visualisasi antarmuka aplikasi pencarian kata kunci berbasis string matching yang dikembangkan dalam penelitian ini. Visualisasi ditampilkan dalam bentuk tangkapan layar aplikasi yang digabungkan, sehingga memberikan representasi menyeluruh terhadap proses pencarian kata kunci dan penyajian hasil yang dihasilkan oleh sistem.

****

**(a)**

****

**(b)**

Gambar 3. Antarmuka pengguna dan visualisasi output dari aplikasi pencarian kata kunci PDF yang diusulkan: (a) antarmuka utama yang menunjukkan unggahan PDF, input kata kunci, pemilihan algoritma, dan perbandingan kinerja; (b) hasil pencarian kata kunci yang menampilkan dokumen yang terdeteksi, kemunculan kata kunci, dan cuplikan teks.

Pada Gambar 3(a) ditunjukkan antarmuka utama aplikasi, yang mencakup fitur unggah dokumen PDF, input kata kunci, serta pemilihan algoritma pencarian (Naive, KMP, atau Boyer–Moore). Selain itu, bagian ini juga menampilkan ringkasan status pencarian dan grafik perbandingan performa waktu eksekusi, yang memungkinkan pengguna untuk mengamati perbedaan performa algoritma secara visual.

Sementara itu, Gambar 3(b) memperlihatkan hasil pencarian kata kunci dalam bentuk daftar dokumen yang terdeteksi, lengkap dengan jumlah kemunculan kata kunci, posisi kemunculan dalam dokumen, serta cuplikan teks (snippet) yang relevan. Penyajian ini memberikan transparansi terhadap hasil pencocokan string yang dilakukan oleh algoritma, sekaligus memudahkan pengguna dalam memverifikasi konteks kemunculan kata kunci pada dokumen yang diproses.

Secara keseluruhan, visualisasi pada Gambar 3 menunjukkan bahwa sistem mampu mengintegrasikan proses pencarian kata kunci dan penyajian performa algoritma dalam satu antarmuka yang konsisten. Visualisasi ini berfungsi sebagai pendukung hasil kuantitatif yang telah disajikan sebelumnya, sekaligus menegaskan keterpaduan antara implementasi algoritma string matching dan keluaran sistem pencarian dokumen digital.

# KESIMPULAN

Penelitian ini mengimplementasikan dan mengevaluasi algoritma string matching klasik—Naive, Knuth–Morris–Pratt (KMP), dan Boyer–Moore—dalam sebuah aplikasi pencarian kata kunci pada dokumen PDF berbasis teks. Hasil pengujian menunjukkan perbedaan performa yang konsisten, di mana Boyer–Moore memberikan efisiensi waktu eksekusi tertinggi, KMP menunjukkan kestabilan performa pada berbagai kondisi, dan Naive memiliki keterbatasan ketika diterapkan pada teks berukuran besar. Temuan ini menegaskan bahwa algoritma string matching klasik tetap relevan dan efektif untuk sistem pencarian dokumen digital apabila dipilih sesuai dengan karakteristik penggunaan. Ke depan, pengembangan sistem dapat diarahkan pada dukungan pencarian multi-kata kunci, penerapan approximate string matching, serta integrasi mekanisme index-based search dan akselerasi komputasi untuk meningkatkan efisiensi pada koleksi dokumen berskala besar, sebagaimana ditunjukkan dalam studi terkait akselerasi pencocokan string berbasis optimisasi komputasi dan perangkat keras (Marchisio et al., 2023). Selain itu, eksplorasi pendekatan approximate pattern matching berbasis struktur data lanjutan, seperti Cartesian tree, juga berpotensi memperluas fleksibilitas sistem pencarian pada skenario data yang lebih kompleks (S. Kim & Han, 2025).

# DAFTAR PUSTAKA

Chhabra, T., Ghuman, S. S., & Tarhio, J. (2025). String searching with mismatches using AVX2 and AVX-512 instructions. *Information Processing Letters*, *189*, 106557. https://doi.org/https://doi.org/10.1016/j.ipl.2025.106557

Dondi, R., Mauri, G., & Zoppis, I. (2022). On the complexity of approximately matching a string to a directed graph. *Information and Computation*, *288*, 104748. https://doi.org/https://doi.org/10.1016/j.ic.2021.104748

Faro, S., Marino, F. P., & Pavone, A. (2023). Improved characters distance sampling for online and offline text searching. *Theoretical Computer Science*, *946*, 113684. https://doi.org/https://doi.org/10.1016/j.tcs.2022.12.034

Faro, S., & Scafiti, S. (2022). A weak approach to suffix automata simulation for exact and approximate string matching. *Theoretical Computer Science*, *933*, 88–103. https://doi.org/https://doi.org/10.1016/j.tcs.2022.08.028

Faro, S., & Scafiti, S. (2023). Compact suffix automata representations for searching long patterns. *Theoretical Computer Science*, *940*, 254–268. https://doi.org/https://doi.org/10.1016/j.tcs.2022.11.005

Flynn, R., & Olukoya, O. (2025). Using approximate matching and machine learning to uncover malicious activity in logs. *Computers & Security*, *151*, 104312. https://doi.org/https://doi.org/10.1016/j.cose.2025.104312

Karcioglu, A. A., & Bulut, H. (2021). Improving hash-q exact string matching algorithm with perfect hashing for DNA sequences. *Computers in Biology and Medicine*, *131*, 104292. https://doi.org/https://doi.org/10.1016/j.compbiomed.2021.104292

Khan, M. G., Halim, Z., & Baig, A. R. (2023). An efficient approach for faster matching of approximate patterns in graphs. *Knowledge-Based Systems*, *276*, 110770. https://doi.org/https://doi.org/10.1016/j.knosys.2023.110770

Kim, S.-H., & Cho, H.-G. (2021). Simpler FM-index for parameterized string matching. *Information Processing Letters*, *165*, 106026. https://doi.org/https://doi.org/10.1016/j.ipl.2020.106026

Kim, S., & Han, Y.-S. (2025). Approximate Cartesian tree pattern matching. *Theoretical Computer Science*, *1056*, 115506. https://doi.org/https://doi.org/10.1016/j.tcs.2025.115506

Kim, S., Ko, S.-K., & Han, Y.-S. (2024). Simon’s congruence pattern matching. *Theoretical Computer Science*, *994*, 114478. https://doi.org/https://doi.org/10.1016/j.tcs.2024.114478

Kim, Y., Kang, M., Na, J. C., & Sim, J. S. (2023). Order-preserving pattern matching with scaling. *Information Processing Letters*, *180*, 106333. https://doi.org/https://doi.org/10.1016/j.ipl.2022.106333

Marchisio, A., Teodonio, F., Rizzi, A., & Shafique, M. (2023). ISMatch: A real-time hardware accelerator for inexact string matching of DNA sequences on FPGA. *Microprocessors and Microsystems*, *97*, 104763. https://doi.org/https://doi.org/10.1016/j.micpro.2023.104763

Nakashima, K., Fujisato, N., Hendrian, D., Nakashima, Y., Yoshinaka, R., Inenaga, S., Bannai, H., Shinohara, A., & Takeda, M. (2022). Parameterized DAWGs: Efficient constructions and bidirectional pattern searches. *Theoretical Computer Science*, *933*, 21–42. https://doi.org/https://doi.org/10.1016/j.tcs.2022.09.008

Parvez, M. R., Hu, W., & Chen, T. (2022). Real-time pattern matching and ranking for early prediction of industrial alarm floods. *Control Engineering Practice*, *120*, 105004. https://doi.org/https://doi.org/10.1016/j.conengprac.2021.105004

Pikies, M., & Ali, J. (2021). Analysis and safety engineering of fuzzy string matching algorithms. *ISA Transactions*, *113*, 1–8. https://doi.org/https://doi.org/10.1016/j.isatra.2020.10.014

Rocco, L. Di, Ferraro Petrillo, U., Giancarlo, R., & Cattaneo, G. (2026). Distributed compressive genomics: Fundamental pattern matching primitives via spark. *Future Generation Computer Systems*, *176*, 108169. https://doi.org/https://doi.org/10.1016/j.future.2025.108169

Vaiwsri, S., Ranbaduge, T., & Christen, P. (2024). Encryption-based sub-string matching for privacy-preserving record linkage. *Journal of Information Security and Applications*, *81*, 103712. https://doi.org/https://doi.org/10.1016/j.jisa.2024.103712

Wang, Y., Zhang, B., Liu, W., Cai, J., & Zhang, H. (2024). STMAP: A novel semantic text matching model augmented with embedding perturbations. *Information Processing & Management*, *61*(1), 103576. https://doi.org/https://doi.org/10.1016/j.ipm.2023.103576

Zhao, L., Yan, Z., Hu, K., Feng, S., Wang, J., Cao, X., & Lin, T. (2024). A memory access number constraint-based string prediction technique for high throughput SCC implemented in AVS3. *Journal of Visual Communication and Image Representation*, *105*, 104338. https://doi.org/https://doi.org/10.1016/j.jvcir.2024.104338