



Analisis Performa Waktu Eksekusi dan Penggunaan Memori Algoritma Linear Search dan Binary Search pada Sistem Pencarian Buku Perpustakaan Digital

Husnul Amisyah, Agung Kharisma Hidayah*

Teknik, Teknik Informatika, Universitas Muhammadiyah Bengkulu, Bengkulu
Jl. Bali, Kampung Bali, Kec. Teluk Segara, Kota Bengkulu, Bengkulu, Indonesia

Email: ¹amisyahhusnul@gmail.com, ^{2,*}kharisma@umb.ac.id

Email Penulis Korespondensi: kharisma@umb.ac.id

Submitted: 25/05/2026; Accepted: 23/07/2026; Published: 24/07/2026

Abstrak—Perkembangan perpustakaan digital mendorong kebutuhan akan sistem pencarian buku yang mampu memberikan hasil secara cepat dan efisien seiring meningkatnya jumlah koleksi data. Meskipun algoritma Linear Search dan Binary Search telah banyak digunakan dalam proses pencarian data, implementasi dan evaluasi performanya pada sistem perpustakaan digital berbasis web masih memerlukan analisis yang mempertimbangkan kondisi implementasi nyata. Penelitian ini bertujuan menganalisis dan membandingkan performa kedua algoritma tersebut berdasarkan waktu eksekusi dan penggunaan memori pada sistem pencarian buku perpustakaan digital. Penelitian menggunakan metode eksperimen dengan mengimplementasikan algoritma Linear Search dan Binary Search menggunakan bahasa pemrograman PHP dan basis data MySQL. Pengujian dilakukan pada dataset berukuran 10, 50, 100, 500, dan 1000 data buku dalam lingkungan pengujian yang sama. Hasil penelitian menunjukkan bahwa Binary Search menghasilkan waktu eksekusi yang lebih rendah pada sebagian besar variasi dataset, namun pada pengujian menggunakan 1000 data waktu eksekusinya sedikit lebih tinggi dibandingkan Linear Search. Rata-rata penggunaan memori Linear Search sebesar 0,000048 MB, sedangkan Binary Search sebesar 0,000043 MB, meskipun pada dataset 500 dan 1000 data penggunaan memori Binary Search lebih tinggi. Temuan ini menunjukkan bahwa performa algoritma tidak hanya dipengaruhi oleh kompleksitas teoritis, tetapi juga oleh implementasi sistem, karakteristik data, dan lingkungan pengujian. Oleh karena itu, pemilihan algoritma pencarian perlu mempertimbangkan kebutuhan sistem dan kondisi implementasi agar diperoleh kinerja yang optimal.

Kata Kunci: Linear Search; Binary Search; Perpustakaan Digital; Waktu Eksekusi; Penggunaan Memori.

Abstract—The growth of digital libraries has increased the need for book search systems capable of providing fast and efficient retrieval as the amount of stored data continues to grow. Although Linear Search and Binary Search algorithms have been widely applied in search processes, their implementation and performance evaluation in web-based digital library systems still require analysis under practical implementation conditions. This study aims to analyze and compare the performance of both algorithms based on execution time and memory usage in a digital library book search system. An experimental method was employed by implementing Linear Search and Binary Search using PHP and MySQL. Performance testing was conducted on datasets containing 10, 50, 100, 500, and 1000 book records under the same testing environment. The results indicate that Binary Search achieved lower execution times for most dataset sizes; however, its execution time was slightly higher than Linear Search when tested with 1000 records. The average memory usage of Linear Search was 0.000048 MB, while Binary Search averaged 0.000043 MB, although Binary Search consumed more memory on the 500- and 1000-record datasets. These findings demonstrate that algorithm performance is influenced not only by theoretical complexity but also by system implementation, data characteristics, and the testing environment. Therefore, selecting a search algorithm should consider the implementation context and system requirements to achieve optimal performance.

Keywords: Linear Search; Binary Search; Digital Library; Execution Time; Memory Usage.

1. PENDAHULUAN

Seiring meningkatnya jumlah koleksi data pada sistem perpustakaan digital, kebutuhan terhadap mekanisme pencarian yang cepat dan efisien juga semakin tinggi. Fitur pencarian buku menjadi salah satu komponen utama karena berperan dalam membantu pengguna menemukan informasi berdasarkan judul, pengarang, maupun kata kunci tertentu. Performa fitur pencarian tidak hanya dipengaruhi oleh jumlah data yang diproses, tetapi juga oleh algoritma pencarian yang digunakan. Oleh karena itu, pemilihan algoritma pencarian menjadi aspek penting dalam pengembangan sistem perpustakaan digital agar mampu memberikan layanan pencarian yang efisien pada berbagai ukuran dataset [1].

Algoritma pencarian merupakan metode yang digunakan untuk menemukan data tertentu dari sekumpulan data yang tersedia. Berbagai algoritma pencarian telah dikembangkan untuk meningkatkan efisiensi proses pencarian sesuai dengan karakteristik data dan kebutuhan sistem. Salah satu penelitian mengembangkan algoritma pencarian cepat untuk basis data fuzzy berskala besar [2]. Penelitian lain melakukan studi komparatif guna membantu pemilihan algoritma pencarian yang tepat berdasarkan karakteristik data [3]. Penelitian lainnya mengusulkan modifikasi terhadap algoritma Binary Search untuk meningkatkan performanya [4]. Penelitian berikutnya mengembangkan penerapan Binary Search pada struktur data graf [5]. Selain itu, terdapat pula penelitian yang mengkaji strategi pencarian yang robust dan adaptif terhadap berbagai kondisi data [6]. Di antara berbagai algoritma tersebut, Linear Search dan Binary Search merupakan dua metode pencarian yang paling banyak digunakan. Linear Search melakukan pencarian secara berurutan mulai dari elemen pertama hingga data

ditemukan sehingga dapat diterapkan pada data yang belum terurut. Namun, waktu pencariannya meningkat secara linear seiring bertambahnya jumlah data [7]. Sebaliknya, Binary Search bekerja dengan membagi ruang pencarian menjadi dua bagian pada setiap iterasi sehingga hanya dapat diterapkan pada data yang telah terurut. Pendekatan ini menghasilkan kompleksitas waktu yang lebih rendah sehingga lebih efisien untuk pencarian pada dataset berukuran besar [8].

Berbagai penelitian menunjukkan bahwa Binary Search umumnya memberikan performa pencarian yang lebih baik dibandingkan Sequential Search pada berbagai sistem informasi. Implementasi algoritma tersebut telah diterapkan pada sistem e-library [9], aplikasi e-order [10], aplikasi surat perjalanan dinas [11], sistem pencarian e-arsip berbasis web, sistem pencarian data jemaat gereja [12], hingga aplikasi kamus bahasa digital [13]. Selain itu, Mohammed et al [14] mengembangkan algoritma Interpolated Binary Search yang mampu meningkatkan efisiensi proses pencarian pada dataset terurut dengan jumlah data yang besar. Hasil-hasil penelitian tersebut menunjukkan bahwa pemilihan algoritma pencarian memiliki peran penting dalam meningkatkan performa sistem informasi.

Meskipun demikian, penelitian-penelitian terdahulu umumnya masih berfokus pada perbandingan waktu eksekusi algoritma dan dilakukan pada objek maupun lingkungan implementasi yang berbeda, misalnya perbandingan Linear Search dan Binary Search pada struktur data larik statis, larik dinamis, dan linked list [15]. Perbedaan karakteristik data, implementasi sistem, serta metode pengujian menyebabkan hasil penelitian sulit dibandingkan secara langsung. Di sisi lain, evaluasi terhadap penggunaan memori sebagai salah satu parameter performa algoritma masih relatif terbatas. Selain itu, belum banyak penelitian yang menguji kedua algoritma menggunakan variasi ukuran dataset yang sama dengan skenario best case, average case, dan worst case dalam satu lingkungan implementasi yang identik sehingga karakteristik masing-masing algoritma dapat diamati secara lebih objektif [16].

Berdasarkan research gap tersebut, penelitian ini menawarkan evaluasi empiris terhadap algoritma Linear Search dan Binary Search pada sistem pencarian buku perpustakaan digital berbasis web yang dikembangkan sebagai media pengujian. Berbeda dengan penelitian sebelumnya, kedua algoritma diimplementasikan pada sistem, lingkungan perangkat lunak, dan dataset yang sama sehingga perbandingan performa dilakukan pada kondisi pengujian yang identik. Evaluasi dilakukan menggunakan lima variasi ukuran dataset, yaitu 10, 50, 100, 500, dan 1000 data buku, dengan tiga skenario pencarian (best case, average case, dan worst case). Selain membandingkan waktu eksekusi, penelitian ini juga mengevaluasi penggunaan memori sehingga karakteristik kedua algoritma dapat dianalisis secara lebih komprehensif. Pendekatan tersebut diharapkan mampu memberikan gambaran yang lebih objektif mengenai efisiensi masing-masing algoritma serta memberikan kontribusi terhadap penelitian mengenai evaluasi performa algoritma pencarian pada sistem perpustakaan digital.

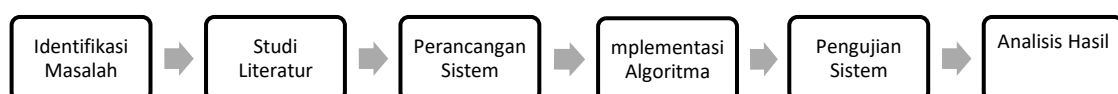
Penelitian ini bertujuan untuk menganalisis dan membandingkan performa algoritma Linear Search dan Binary Search pada sistem pencarian buku perpustakaan digital berbasis web. Analisis dilakukan berdasarkan parameter waktu eksekusi dan penggunaan memori menggunakan dataset sebanyak 10, 50, 100, 500, dan 1000 data buku dengan skenario best case, average case, dan worst case. Hasil penelitian diharapkan dapat menjadi referensi bagi pengembang sistem informasi dalam menentukan algoritma pencarian yang sesuai dengan karakteristik data yang digunakan serta memberikan kontribusi terhadap pengembangan penelitian mengenai evaluasi performa algoritma pencarian pada sistem perpustakaan digital.

2. METODOLOGI PENELITIAN

Penelitian ini menggunakan metode eksperimen (experimental research) untuk menganalisis dan membandingkan performa algoritma Linear Search dan Binary Search pada sistem pencarian buku perpustakaan digital berbasis web. Metode eksperimen dipilih karena memungkinkan kedua algoritma diuji pada lingkungan implementasi yang sama sehingga hasil pengukuran dapat dibandingkan secara objektif. Parameter yang dianalisis meliputi waktu eksekusi (execution time), jumlah iterasi, dan penggunaan memori (memory usage). Pengujian dilakukan menggunakan beberapa variasi ukuran dataset sehingga dapat diketahui perubahan performa algoritma ketika jumlah data meningkat.

2.1 Tahapan Penelitian

Penelitian dilakukan secara bertahap agar proses implementasi dan pengujian berjalan secara sistematis. Tahapan penelitian dimulai dari identifikasi masalah, studi literatur, perancangan sistem, implementasi algoritma, pengujian performa, hingga analisis hasil penelitian. Alur penelitian ditunjukkan pada Gambar 1.



Gambar 1. Tahapan Penelitian

Berdasarkan Gambar 1, penelitian dilakukan secara bertahap mulai dari identifikasi masalah, studi literatur, perancangan sistem, implementasi algoritma, pengujian sistem, hingga analisis hasil sehingga seluruh proses penelitian dapat dilakukan secara sistematis.

- a. Tahap pertama dilakukan dengan mengidentifikasi permasalahan pada sistem pencarian buku di perpustakaan digital. Permasalahan utama yang ditemukan adalah meningkatnya kebutuhan terhadap proses pencarian data yang cepat dan efisien seiring bertambahnya jumlah koleksi buku. Oleh karena itu, diperlukan evaluasi terhadap algoritma pencarian yang mampu memberikan performa optimal pada berbagai ukuran dataset.
- b. Tahap ini dilakukan dengan mengumpulkan berbagai referensi yang berkaitan dengan algoritma Linear Search, Binary Search, sistem perpustakaan digital, serta penelitian terdahulu yang relevan. Studi literatur digunakan sebagai dasar dalam penyusunan metodologi penelitian serta sebagai acuan dalam menganalisis hasil pengujian.
- c. Pada tahap ini dilakukan perancangan sistem pencarian buku berbasis web menggunakan bahasa pemrograman PHP dan basis data MySQL. Sistem dirancang agar kedua algoritma dapat diimplementasikan pada lingkungan yang sama sehingga hasil pengujian dapat dibandingkan secara objektif.
- d. Tahap implementasi dilakukan dengan menerapkan algoritma Linear Search dan Binary Search pada sistem pencarian buku. Kedua algoritma digunakan untuk mencari data berdasarkan judul buku dengan parameter pengujian yang sama.
- e. Setelah implementasi selesai, dilakukan pengujian menggunakan beberapa ukuran dataset dan tiga skenario pencarian, yaitu best case, average case, dan worst case. Pengujian dilakukan secara berulang agar hasil yang diperoleh lebih konsisten.
- f. Tahap terakhir adalah menganalisis seluruh hasil pengujian berdasarkan waktu eksekusi, jumlah iterasi, dan penggunaan memori. Selanjutnya hasil tersebut dibandingkan dengan kompleksitas teoritis masing-masing algoritma sehingga dapat diketahui karakteristik performa kedua algoritma.

2.2 Implementasi Algoritma

Pada penelitian ini diterapkan dua algoritma pencarian, yaitu Linear Search dan Binary Search. Kedua algoritma diimplementasikan menggunakan bahasa pemrograman PHP dengan basis data MySQL pada sistem pencarian buku perpustakaan digital.

2.2.1 Linear Search

Linear Search merupakan algoritma pencarian yang bekerja dengan membandingkan data secara berurutan mulai dari elemen pertama hingga data yang dicari ditemukan atau seluruh data selesai diperiksa. Algoritma ini tidak memerlukan data dalam kondisi terurut sehingga lebih sederhana untuk diimplementasikan. Namun, semakin banyak jumlah data maka semakin besar pula jumlah proses perbandingan yang dilakukan. Kompleksitas waktu algoritma Linear Search dinyatakan sebagai:

$$T(n) = O(n) \quad (1)$$

dengan $T(n)$ menyatakan waktu pencarian dan n menyatakan jumlah data yang diproses. Berdasarkan persamaan tersebut, waktu pencarian Linear Search akan meningkat secara linear seiring bertambahnya jumlah data. Oleh karena itu, semakin besar nilai n , semakin banyak proses perbandingan yang harus dilakukan sehingga waktu pencarian juga akan semakin meningkat.

2.2.2 Binary Search

Binary Search merupakan algoritma pencarian yang bekerja dengan membagi ruang pencarian menjadi dua bagian pada setiap iterasi sehingga jumlah data yang diperiksa terus berkurang hingga data ditemukan. Konsep dasar algoritma ini telah dijelaskan secara umum dalam beberapa kajian [17]. Salah satu penelitian menerapkan algoritma ini secara khusus pada pencarian data bilangan asli (natural number) [18]. Penelitian lain membandingkan varian Binary Search standar dan uniform dalam konteks pengindeksan statis berbasis pembelajaran [19]. Algoritma ini hanya dapat diterapkan apabila data telah diurutkan terlebih dahulu. Kompleksitas waktu algoritma Binary Search dinyatakan sebagai:

$$T(n) = O(\log_2 n) \quad (2)$$

dengan $T(n)$ menyatakan waktu pencarian dan n menyatakan jumlah data yang diproses. Persamaan tersebut menunjukkan bahwa penambahan waktu pencarian pada Binary Search berlangsung secara logaritmik karena ruang pencarian dibagi menjadi dua bagian pada setiap iterasi. Akibatnya, peningkatan jumlah data tidak menyebabkan kenaikan waktu pencarian secara signifikan dibandingkan Linear Search. Karena ruang pencarian terus dibagi menjadi dua bagian, jumlah iterasi yang diperlukan relatif lebih sedikit dibandingkan Linear Search [20]. Penelitian lain mengusulkan pendekatan Bayesian dalam Binary Search untuk meningkatkan efisiensi pencarian [21]. Penelitian lainnya mengkaji batas teoretis kompleksitas pada struktur Binary Search Tree [22].

2.3 Dataset dan Skenario Pengujian

Pengujian dilakukan menggunakan lima variasi jumlah dataset untuk mengetahui perubahan performa kedua algoritma ketika jumlah data meningkat.

Tabel 1. Dataset Pengujian

Jumlah Dataset	Skenario Pengujian
10 Data	Best Case, Average Case, Worst Case
50 Data	Best Case, Average Case, Worst Case
100 Data	Best Case, Average Case, Worst Case
500 Data	Best Case, Average Case, Worst Case
1000 Data	Best Case, Average Case, Worst Case

Tabel 1 Menunjukkan Setiap skenario dilakukan sebanyak 10 kali pengujian kemudian dihitung nilai rata-ratanya untuk mengurangi variasi hasil pengukuran. Adapun skenario pengujian terdiri atas:

- Best Case, yaitu kondisi ketika data berada pada posisi yang paling menguntungkan bagi algoritma.
- Average Case, yaitu kondisi ketika data berada pada posisi tengah sehingga merepresentasikan pencarian normal.
- Worst Case, yaitu kondisi ketika data berada pada posisi terakhir atau tidak ditemukan sehingga algoritma melakukan proses pencarian maksimum.

2.4 Lingkungan Pengujian

Seluruh pengujian dilakukan menggunakan perangkat keras dan perangkat lunak yang sama agar hasil eksperimen dapat direplikasi.

Tabel 2. Spesifikasi Lingkungan Pengujian

Komponen	Spesifikasi
Processor	Intel Core i3-1215U
RAM	4 GB
Sistem Operasi	Windows 11 64-bit
Web Server	Apache 2.4 (XAMPP)
PHP	Versi 8.x
Database	MySQL/MariaDB
Browser	Google Chrome

Berdasarkan Tabel 2, seluruh pengujian dilaksanakan menggunakan spesifikasi perangkat keras dan perangkat lunak yang sama. Lingkungan pengujian yang identik bertujuan untuk meminimalkan pengaruh faktor eksternal sehingga hasil pengukuran waktu eksekusi dan penggunaan memori dapat dibandingkan secara objektif. Pengujian dilakukan pada kondisi komputer idle dengan aplikasi lain ditutup untuk meminimalkan pengaruh proses lain terhadap waktu eksekusi dan penggunaan memori. Dengan demikian, seluruh pengukuran dilakukan pada kondisi lingkungan yang sama sehingga hasil eksperimen lebih konsisten dan dapat direplikasi.

2.5 Parameter Pengujian

Evaluasi performa algoritma pada penelitian ini dilakukan menggunakan dua parameter utama, yaitu waktu eksekusi (execution time) dan penggunaan memori (memory usage). Kedua parameter tersebut dipilih karena mampu menggambarkan tingkat efisiensi algoritma dalam melakukan proses pencarian data, baik dari aspek kecepatan pemrosesan maupun penggunaan sumber daya sistem selama proses pencarian berlangsung. Pengukuran kedua parameter dilakukan pada setiap variasi ukuran dataset dengan kondisi lingkungan pengujian yang sama sehingga hasil yang diperoleh dapat dibandingkan secara objektif.

a. Waktu Eksekusi (Execution Time)

Waktu eksekusi merupakan lama waktu yang dibutuhkan algoritma untuk menemukan data yang dicari sejak proses pencarian dimulai hingga selesai. Pengukuran dilakukan menggunakan fungsi `microtime(true)` pada PHP sehingga diperoleh nilai waktu eksekusi dalam satuan detik. Parameter ini digunakan untuk mengetahui tingkat kecepatan masing-masing algoritma dalam menyelesaikan proses pencarian.

b. Penggunaan Memori (Memory Usage)

Penggunaan memori menunjukkan jumlah memori yang digunakan selama proses pencarian berlangsung. Pengukuran dilakukan menggunakan fungsi `memory_get_usage` pada PHP untuk mengetahui efisiensi penggunaan sumber daya sistem oleh masing-masing algoritma. Semakin kecil penggunaan memori yang dibutuhkan, maka semakin efisien algoritma dalam memanfaatkan sumber daya yang tersedia.

3. HASIL DAN PEMBAHASAN

Penelitian ini bertujuan membandingkan performa algoritma Linear Search dan Binary Search pada sistem pencarian buku perpustakaan digital berbasis web. Pengujian dilakukan menggunakan lima variasi dataset, yaitu 10, 50, 100, 500, dan 1000 data buku dengan tiga skenario pengujian, yaitu best case, average case, dan worst case. Evaluasi dilakukan berdasarkan dua parameter utama, yaitu waktu eksekusi (execution time) dan penggunaan

memori (memory usage). Seluruh pengujian dilaksanakan pada lingkungan perangkat keras dan perangkat lunak yang sama agar hasil yang diperoleh dapat dibandingkan secara objektif.

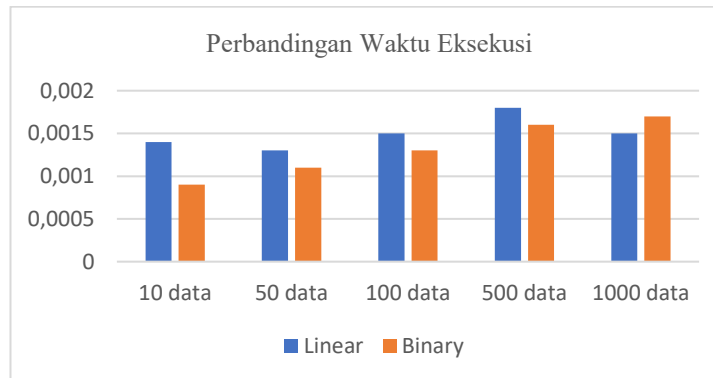
3.1 Hasil Pengujian Waktu Eksekusi

Pengujian waktu eksekusi dilakukan untuk mengetahui lama waktu yang dibutuhkan oleh algoritma Linear Search dan Binary Search dalam menemukan data buku pada berbagai ukuran dataset. Waktu eksekusi diukur dalam satuan detik menggunakan fungsi microtime pada PHP. Semakin kecil nilai waktu eksekusi yang diperoleh, maka semakin baik performa algoritma dalam melakukan proses pencarian. Hasil nilai rata-rata dari seluruh skenario pengujian yang telah dilakukan di sajikan dalam Tabel 3.

Tabel 3. Perbandingan Rata-Rata Waktu Pencarian (Detik)

Data Buku	Linear Search	Binary Search
10	0,0014	0,0009
50	0,0013	0,0011
100	0,0015	0,0013
500	0,0018	0,0016
1000	0,0015	0,0017
Rata-rata	0,0015	0,0013

Berdasarkan Tabel 3, waktu eksekusi kedua algoritma menunjukkan kecenderungan meningkat seiring bertambahnya jumlah dataset yang digunakan. Namun demikian, peningkatan waktu eksekusi pada Binary Search relatif lebih kecil dibandingkan Linear Search. Hal tersebut menunjukkan bahwa mekanisme pembagian ruang pencarian pada Binary Search mampu mengurangi jumlah data yang harus diperiksa sehingga proses pencarian menjadi lebih efisien. Pada dataset berukuran kecil, perbedaan waktu eksekusi kedua algoritma masih relatif kecil karena jumlah data yang diproses belum terlalu banyak. Akan tetapi, ketika jumlah dataset bertambah hingga 500 dan 1000 data, Binary Search menunjukkan waktu eksekusi yang lebih stabil dibandingkan Linear Search. Hasil ini sesuai dengan teori kompleksitas algoritma, di mana Linear Search memiliki kompleksitas waktu $O(n)$ sehingga waktu pencarian meningkat seiring bertambahnya jumlah data, sedangkan Binary Search memiliki kompleksitas waktu $O(\log n)$ yang memungkinkan proses pencarian berlangsung lebih cepat pada data yang telah terurut. Selain itu, hasil penelitian ini juga sejalan dengan penelitian terdahulu yang menyatakan bahwa Binary Search lebih efisien dibandingkan Linear Search pada proses pencarian data yang telah diurutkan [19], [23]. Dengan demikian, algoritma Binary Search lebih sesuai diterapkan pada sistem perpustakaan digital yang memiliki koleksi data dalam jumlah besar dan telah tersusun secara terstruktur.



Gambar 2. Grafik Perbandingan Waktu Eksekusi

Berdasarkan Gambar 2, terlihat bahwa waktu eksekusi algoritma Linear Search dan Binary Search mengalami perubahan pada setiap variasi jumlah dataset yang digunakan. Pada dataset 10, 50, 100, dan 500 data, algoritma Binary Search menghasilkan waktu eksekusi yang lebih rendah dibandingkan Linear Search. Hasil tersebut menunjukkan bahwa mekanisme pembagian ruang pencarian pada Binary Search mampu mengurangi jumlah data yang harus diperiksa sehingga proses pencarian menjadi lebih efisien ketika data telah berada dalam kondisi terurut. Sebaliknya, Linear Search melakukan pencarian secara berurutan mulai dari elemen pertama hingga data yang dicari ditemukan, sehingga waktu pencarian cenderung meningkat seiring bertambahnya jumlah data. Namun, pada pengujian menggunakan dataset sebanyak 1000 data, algoritma Binary Search menunjukkan waktu eksekusi yang sedikit lebih tinggi dibandingkan Linear Search. Perbedaan tersebut mengindikasikan bahwa performa aktual suatu algoritma tidak hanya dipengaruhi oleh kompleksitas teoritis, tetapi juga oleh faktor implementasi dan lingkungan eksekusi, seperti overhead pemanggilan fungsi, pengelolaan memori, kondisi server lokal, serta mekanisme eksekusi PHP dan MySQL yang digunakan selama proses pengujian. Selain itu, ukuran dataset yang masih relatif kecil, yaitu maksimum 1000 data, menyebabkan selisih waktu eksekusi berada pada skala milidetik sehingga variasi kecil pada lingkungan pengujian dapat memengaruhi hasil yang diperoleh.

Secara keseluruhan, hasil pengujian menunjukkan bahwa Binary Search memiliki kecenderungan memberikan waktu eksekusi yang lebih baik pada sebagian besar variasi dataset. Temuan ini sejalan dengan teori kompleksitas algoritma, di mana Linear Search memiliki kompleksitas waktu $O(n)$, sedangkan Binary Search memiliki kompleksitas waktu $O(\log n)$ pada data yang telah diurutkan. Meskipun demikian, penelitian ini juga menunjukkan bahwa kompleksitas teoritis tidak selalu tercermin secara identik pada implementasi nyata, karena hasil pengujian dipengaruhi oleh berbagai faktor teknis selama proses eksekusi. Oleh karena itu, pemilihan algoritma pencarian sebaiknya tidak hanya mempertimbangkan teori kompleksitas, tetapi juga karakteristik data, kebutuhan sistem, serta lingkungan implementasi yang digunakan. Penelitian selanjutnya disarankan menggunakan dataset yang lebih besar dan melakukan pengujian berulang pada lingkungan yang lebih terkontrol agar hasil evaluasi performa menjadi lebih akurat dan representatif.

3.2 Analisis Sumber Daya Memori (Memory Profiling)

Selain waktu eksekusi, penelitian ini juga mengevaluasi penggunaan memori sebagai salah satu parameter untuk mengetahui efisiensi algoritma dalam memanfaatkan sumber daya sistem selama proses pencarian. Penggunaan memori diukur menggunakan fungsi `memory_get_usage` pada PHP dan dinyatakan dalam satuan megabyte (MB). Hasil rata-rata penggunaan memori algoritma Linear Search dan Binary Search pada setiap variasi dataset disajikan pada Tabel 4.

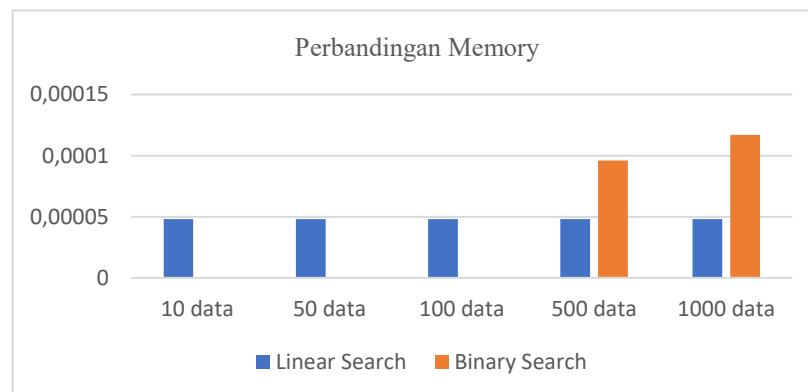
Tabel 4. Perbandingan Rata-Rata Penggunaan Memori (MB)

Data	Linear Search	Binary Search
10	0,000048	0
50	0,000048	0
100	0,000048	0
500	0,000048	0,000096
1000	0,000048	0,000117
Rata-rata	0,000048	0,000043

Berdasarkan Tabel 4, algoritma Linear Search menunjukkan penggunaan memori yang relatif konstan sebesar 0,000048 MB pada seluruh variasi dataset. Sebaliknya, Binary Search menunjukkan penggunaan memori sebesar 0 MB pada dataset 10, 50, dan 100 data, kemudian meningkat menjadi 0,000096 MB pada dataset 500 data dan 0,000117 MB pada dataset 1000 data. Secara keseluruhan, rata-rata penggunaan memori Binary Search sebesar 0,000043 MB, sedikit lebih rendah dibandingkan Linear Search yang memiliki rata-rata 0,000048 MB.

Nilai penggunaan memori sebesar 0 MB pada beberapa dataset tidak menunjukkan bahwa algoritma tidak menggunakan memori sama sekali. Nilai tersebut mengindikasikan bahwa perubahan penggunaan memori berada di bawah batas deteksi fungsi `memory_get_usage()` sehingga tidak dapat direkam oleh instrumen pengukuran. Oleh karena itu, hasil tersebut merupakan keterbatasan metode pengukuran dan bukan karakteristik algoritma. Hal ini juga menjadi salah satu keterbatasan penelitian yang perlu diperhatikan dalam menginterpretasikan hasil pengujian.

Berdasarkan Gambar 3, menunjukkan perbandingan penggunaan memori algoritma Linear Search dan Binary Search pada berbagai ukuran dataset. Berdasarkan grafik tersebut, penggunaan memori Linear Search terlihat stabil pada seluruh variasi dataset dengan nilai sekitar 0,000048 MB. Sementara itu, Binary Search tidak menunjukkan perubahan penggunaan memori yang terdeteksi pada dataset 10, 50, dan 100 data, kemudian mengalami peningkatan pada dataset 500 dan 1000 data. Peningkatan penggunaan memori pada Binary Search menunjukkan bahwa kebutuhan memori dapat bertambah ketika jumlah data yang diproses semakin besar. Meskipun demikian, besarnya peningkatan tersebut masih berada pada skala yang sangat kecil sehingga belum memberikan pengaruh yang signifikan terhadap kinerja sistem secara keseluruhan. Hasil ini menunjukkan bahwa kedua algoritma memiliki kebutuhan memori yang relatif efisien untuk pengujian hingga 1000 data buku.



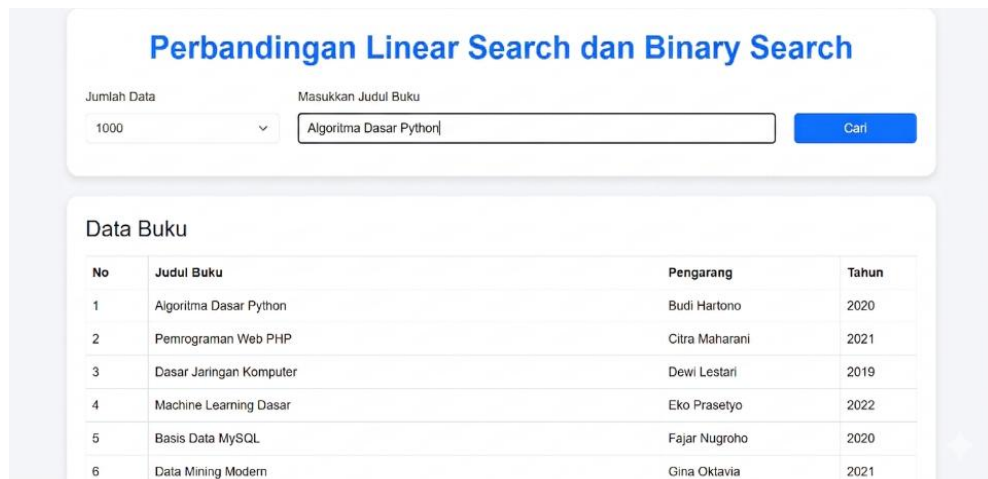
Gambar 3. Grafik Perbandingan Memory

Jika dibandingkan berdasarkan rata-rata keseluruhan, Binary Search memiliki penggunaan memori yang sedikit lebih rendah dibandingkan Linear Search. Akan tetapi, pada dataset berukuran 500 dan 1000 data, Binary Search justru menggunakan memori lebih besar. Hal ini menunjukkan bahwa penggunaan memori tidak hanya dipengaruhi oleh karakteristik algoritma, tetapi juga oleh implementasi program, mekanisme alokasi memori PHP, serta lingkungan pengujian yang digunakan. Oleh karena itu, pada penelitian ini penggunaan memori belum dapat dijadikan indikator utama untuk menyatakan bahwa salah satu algoritma lebih unggul dibandingkan algoritma lainnya. Secara keseluruhan, hasil penelitian menunjukkan bahwa perbedaan penggunaan memori antara Linear Search dan Binary Search relatif kecil dibandingkan perbedaan waktu eksekusi. Dengan demikian, parameter waktu eksekusi menjadi indikator yang lebih representatif dalam mengevaluasi performa kedua algoritma pada sistem pencarian buku perpustakaan digital. Untuk penelitian selanjutnya, pengujian dapat dilakukan menggunakan dataset yang lebih besar serta perangkat profiler dengan tingkat akurasi yang lebih tinggi agar perubahan penggunaan memori dapat diamati secara lebih rinci.

3.3 Implementasi Sistem Pencarian pada Perpustakaan Digital

3.3.1 Antarmuka Sistem Pencarian Buku

Implementasi sistem dilakukan untuk menerapkan algoritma Linear Search dan Binary Search pada sistem pencarian buku perpustakaan digital berbasis web. Sistem dikembangkan menggunakan bahasa pemrograman PHP dengan basis data MySQL sebagai media penyimpanan data buku. Antarmuka dirancang sederhana, responsif, dan mudah digunakan sehingga pengguna dapat melakukan proses pencarian buku serta mendukung pelaksanaan pengujian performa algoritma secara langsung. Selain berfungsi sebagai media pencarian informasi buku, sistem juga digunakan sebagai sarana untuk mengukur waktu eksekusi dan penggunaan memori pada setiap proses pencarian. Sebagaimana ditunjukkan pada Gambar 4.



No	Judul Buku	Pengarang	Tahun
1	Algoritma Dasar Python	Budi Hartono	2020
2	Pemrograman Web PHP	Citra Maharani	2021
3	Dasar Jaringan Komputer	Dewi Lestari	2019
4	Machine Learning Dasar	Eko Prasetyo	2022
5	Basis Data MySQL	Fajar Nugroho	2020
6	Data Mining Modern	Gina Oktavia	2021

Gambar 4. Tampilan Halaman Pencarian Buku

Berdasarkan Gambar 4, halaman utama sistem menampilkan beberapa komponen utama, yaitu pilihan jumlah dataset, kolom input kata kunci berdasarkan judul buku, tombol Cari, serta tabel yang berisi daftar koleksi buku. Pengguna dapat menentukan jumlah dataset yang akan digunakan dalam pengujian, kemudian memasukkan judul buku yang ingin dicari sebelum menjalankan proses pencarian. Tabel data buku menampilkan informasi berupa nomor, judul buku, nama pengarang, dan tahun terbit sehingga pengguna dapat melihat data yang tersedia pada sistem. Antarmuka tersebut dirancang agar proses pengujian kedua algoritma dilakukan pada kondisi yang sama, sehingga hasil yang diperoleh dapat dibandingkan secara objektif. Setiap proses pencarian menggunakan dataset yang dipilih oleh pengguna dan kata kunci yang sama untuk kedua algoritma, sehingga perbedaan hasil pengujian lebih mencerminkan karakteristik algoritma dibandingkan perbedaan kondisi pengujian. Dengan demikian, implementasi sistem tidak hanya berfungsi sebagai media pencarian buku, tetapi juga sebagai sarana untuk mengevaluasi performa algoritma berdasarkan parameter waktu eksekusi dan penggunaan memori.

3.3.2 Hasil Pencarian Buku

Setelah pengguna memasukkan kata kunci judul buku dan menekan tombol Cari, sistem akan memproses data menggunakan algoritma Linear Search dan Binary Search secara terpisah. Hasil pencarian kemudian ditampilkan dalam bentuk informasi buku yang berhasil ditemukan, meliputi judul buku, nama pengarang, dan tahun terbit. Selain menampilkan informasi buku, sistem juga menyajikan waktu eksekusi yang dibutuhkan oleh masing-masing algoritma sehingga pengguna dapat membandingkan performa pencarian secara langsung pada dataset yang sama. Tampilan hasil pencarian buku pada sistem perpustakaan digital ditunjukkan pada Gambar 5.

Hasil Linear Search	Hasil Binary Search
Judul Buku : Algoritma Dasar Python	Judul Buku : Algoritma Dasar Python
Pengarang : Andi Setiawan	Pengarang : Andi Setiawan
Tahun : 2020	Tahun : 2020
0.0016 detik	0.0009 detik

Gambar 5. Hasil Pencarian Buku

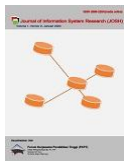
Gambar 5, sistem berhasil menemukan data buku sesuai dengan kata kunci "Algoritma Dasar Python" yang dimasukkan oleh pengguna. Hasil pencarian ditampilkan dalam dua panel terpisah, yaitu hasil pencarian menggunakan algoritma Linear Search dan Binary Search. Kedua algoritma menghasilkan data buku yang sama, yaitu judul buku Algoritma Dasar Python, pengarang Andi Setiawan, dan tahun terbit 2020, sehingga menunjukkan bahwa kedua algoritma memiliki tingkat akurasi yang sama dalam menemukan data yang sesuai dengan kata kunci pencarian. Perbedaan yang terlihat pada implementasi ini terdapat pada waktu eksekusi masing-masing algoritma. Pada contoh pengujian yang ditampilkan, Linear Search memerlukan waktu eksekusi sebesar 0,0016 detik, sedangkan Binary Search memerlukan waktu 0,0009 detik untuk menemukan data yang sama. Perbedaan tersebut menunjukkan bahwa pada skenario pencarian ini, Binary Search mampu menyelesaikan proses pencarian lebih cepat dibandingkan Linear Search. Hal ini disebabkan karena Binary Search melakukan pencarian dengan mempersempit ruang pencarian secara bertahap pada data yang telah diurutkan, sedangkan Linear Search melakukan pemeriksaan data secara berurutan mulai dari elemen pertama hingga data ditemukan. Meskipun demikian, hasil yang ditampilkan pada Gambar 5 merupakan salah satu contoh implementasi sistem pada proses pencarian buku. Evaluasi performa algoritma secara menyeluruh tetap mengacu pada hasil pengujian waktu eksekusi dan penggunaan memori yang telah dibahas pada subbab sebelumnya. Oleh karena itu, kesimpulan mengenai performa kedua algoritma tidak hanya didasarkan pada satu contoh pencarian, tetapi berdasarkan keseluruhan hasil eksperimen pada berbagai variasi ukuran dataset.

3.4 Pembahasan

Hasil penelitian menunjukkan bahwa algoritma Binary Search secara umum memiliki performa waktu eksekusi yang lebih baik dibandingkan Linear Search pada sebagian besar variasi dataset yang digunakan. Temuan ini sejalan dengan penelitian [9] yang menerapkan Binary Search pada sistem e-library serta penelitian [11] yang menyatakan bahwa Binary Search lebih efisien dibandingkan Sequential Search pada proses pencarian data karena mampu mengurangi jumlah data yang diperiksa melalui pembagian ruang pencarian secara berulang. Hasil penelitian ini juga mendukung penelitian [14] yang menjelaskan bahwa Binary Search memberikan efisiensi pencarian yang lebih baik pada dataset yang telah terurut. Meskipun demikian, pada penelitian ini ditemukan bahwa waktu eksekusi Binary Search sedikit lebih tinggi dibandingkan Linear Search pada dataset berjumlah 1000 data. Perbedaan tersebut menunjukkan bahwa performa algoritma pada implementasi nyata tidak hanya dipengaruhi oleh kompleksitas teoritis, tetapi juga oleh faktor implementasi program, mekanisme eksekusi PHP dan MySQL, kondisi perangkat keras, serta lingkungan pengujian yang digunakan. Selain membandingkan waktu eksekusi, penelitian ini juga mengevaluasi penggunaan memori sebagai parameter tambahan. Hasil pengujian menunjukkan bahwa perbedaan penggunaan memori antara kedua algoritma relatif kecil sehingga penggunaan memori belum dapat dijadikan indikator utama dalam menentukan algoritma yang lebih unggul pada sistem pencarian buku perpustakaan digital. Dengan demikian, penelitian ini memperkuat hasil penelitian terdahulu bahwa Binary Search lebih sesuai diterapkan pada data yang telah terurut dan berukuran besar, sekaligus memberikan kontribusi berupa evaluasi penggunaan memori dan pengujian kedua algoritma pada lingkungan implementasi yang sama dengan variasi ukuran dataset yang identik sehingga perbandingan performa dapat dilakukan secara lebih objektif. Temuan itu juga konsisten dengan penelitian [15] yang menunjukkan bahwa Binary Search memiliki efisiensi waktu yang lebih baik dibandingkan Linear Search pada data yang telah diurutkan.

4. KESIMPULAN

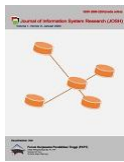
Berdasarkan hasil penelitian dan pengujian yang telah dilakukan pada sistem pencarian buku di perpustakaan digital menggunakan algoritma Linear Search dan Binary Search, dapat disimpulkan bahwa sistem yang dibangun telah berhasil mengimplementasikan kedua algoritma tersebut untuk melakukan proses pencarian data buku berdasarkan kata kunci judul. Sistem juga mampu melakukan pengujian performa algoritma berdasarkan waktu eksekusi pencarian dan penggunaan memori pada berbagai jumlah dataset. Berdasarkan hasil pengujian menggunakan variasi jumlah data mulai dari 10 hingga 1000 data buku, algoritma Linear Search menunjukkan



bahwa waktu pencarian meningkat seiring dengan bertambahnya jumlah data yang digunakan. Hal ini terjadi karena Linear Search melakukan proses pemeriksaan data secara berurutan dari elemen pertama hingga data ditemukan atau seluruh data selesai diperiksa. Pada kondisi terburuk (worst case), ketika data berada di posisi terakhir atau tidak ditemukan, algoritma ini membutuhkan proses pencarian yang lebih lama dengan kompleksitas waktu $O(n)$. Sementara itu, algoritma Binary Search memiliki performa pencarian yang lebih baik dibandingkan Linear Search, terutama pada jumlah data yang besar. Hal tersebut dikarenakan Binary Search melakukan proses pencarian dengan membagi data menjadi dua bagian secara berulang sehingga jumlah perbandingan data menjadi lebih sedikit. Berdasarkan hasil pengujian, Binary Search menghasilkan waktu pencarian yang lebih rendah pada sebagian besar skenario pengujian dengan kompleksitas waktu $O(\log n)$. Namun, algoritma ini memiliki syarat bahwa data harus sudah dalam kondisi terurut sebelum proses pencarian dilakukan. Hasil pengujian penggunaan memori menunjukkan bahwa kedua algoritma tidak memiliki perbedaan penggunaan memori yang signifikan karena keduanya menggunakan sumber data yang sama dan tidak membutuhkan struktur data tambahan yang besar. Perbedaan utama dari kedua algoritma lebih terlihat pada efisiensi waktu eksekusi dalam proses pencarian data. Berdasarkan hasil penelitian tersebut, dapat disimpulkan bahwa Binary Search lebih efektif diterapkan pada sistem pencarian buku dengan jumlah data yang besar dan data yang telah terurut, karena mampu menghasilkan waktu pencarian yang lebih cepat dibandingkan Linear Search. Sedangkan Linear Search masih dapat digunakan pada sistem dengan jumlah data yang relatif kecil atau kondisi data yang belum terurut karena memiliki implementasi yang lebih sederhana. Dengan demikian, sistem pencarian buku digital yang dikembangkan telah berhasil menjadi media pengujian untuk menganalisis perbandingan performa algoritma Linear Search dan Binary Search secara nyata. Hasil penelitian ini menunjukkan bahwa pemilihan algoritma pencarian yang tepat dapat meningkatkan efisiensi proses pencarian data pada sistem perpustakaan digital.

REFERENCES

- [1] Y. Reswan, Y. Darnita, A. R. W. M. Mahfuzhi, and E. Elviani, "Implementation Of Sequential Searching Algorithm In Car Sales E-Commerce Web," *J. Komputer, Inf. dan Teknol.*, vol. 4, no. 1, pp. 1–14, 2024, doi: 10.53697/jkomitek.v4i1.1807.
- [2] F. Hao, J. Daugman, and P. Zieliński, "A fast search algorithm for a large fuzzy database," *IEEE Trans. Inf. Forensics Secur.*, vol. 3, no. 2, pp. 203–212, 2008, doi: 10.1109/TIFS.2008.920726.
- [3] S. Xavier P, J. V A, and B. M, "How to Select a Searching Algorithm-A Comparative Study," *Int. J. Comput. Trends Technol.*, vol. 48, no. 3, pp. 108–111, 2017, doi: 10.14445/22312803/ijett-v48p122.
- [4] A. R. Chadha, R. Misal, and T. Mokashi, "Modified Binary Search Algorithm," *Int. J. Appl. Inf. Syst.*, vol. 7, no. 2, pp. 2089–2100, 2014.
- [5] A. Deligkas, G. B. Mertzios, and P. G. Spirakis, "Binary Search in Graphs Revisited," *Algorithmica*, vol. 81, no. 5, pp. 1757–1780, 2019, doi: 10.1007/s00453-018-0501-y.
- [6] Y. Disser and S. Kratsch, "Robust and adaptive search," *Leibniz Int. Proc. Informatics, LIPIcs*, vol. 66, no. 26, pp. 1–14, 2017, doi: 10.4230/LIPIcs.STACS.2017.26.
- [7] W. S. Wahyuni, S. Andryana, and B. Rahman, "Penggunaan Algoritma Sequential Searching Pada Aplikasi Perpustakaan Berbasis Web," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 7, no. 2, pp. 294–302, 2022, doi: 10.29100/jipi.v7i2.2646.
- [8] S. Salakos and N. Ploskas, "Analysis and Comparison of Binary and Interpolation Search Algorithms in a B-tree," *ACM Int. Conf. Proceeding Ser.*, pp. 74–78, 2021, doi: 10.1145/3503823.3503837.
- [9] T. K. Wulandari, "Penerapan Metode Binary Search dan Hamming Distance untuk E-library SMAN 2 Katingan Hilir," *KONSTELASI Konvergensi Teknol. dan Sist. Inf.*, vol. 2, no. 1, pp. 33–42, 2022, doi: 10.24002/konstelasi.v2i1.5623.
- [10] R. Toyib, Y. Darnita, and A. R. S. Deva, "Penerapan Algoritma Binary Search Pada Aplikasi E-Order," *J. Media Infotama*, vol. 17, no. 1, pp. 30–37, 2021, doi: 10.37676/jmi.v17i1.1314.
- [11] D. Markuci and C. Prianto, "Analisis Perbandingan Penggunaan Algoritma Sequential Search Dan Binary Search Pada Aplikasi Surat Perjalanan Dinas," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 6, no. 1, pp. 110–119, 2022, doi: 10.36040/jati.v6i1.4569.
- [12] B. J. D. Sitompul, A. Yusupa, and N. J. Tuturoong, "Implementasi Algoritma Binary Search Pada Pencarian Data Jemaat Gereja Hkbp Manado," *J. Inform. Polinema*, vol. 9, no. 1, pp. 17–24, 2022, doi: 10.33795/jip.v9i1.1123.
- [13] Agung Kharisma, "Aplikasi Kamus Bahasa Indonesia - Rejang Menggunakan Metode Sequential Searching Berbasis Android," *J. Process.*, vol. 17, no. 1, pp. 58–65, 2022, doi: 10.33998/processor.2022.17.1.1171.
- [14] A. S. Mohammed, Ş. E. Amrahov, and F. V. Çelebi, "Interpolated binary search: An efficient hybrid search algorithm on ordered datasets," *Eng. Sci. Technol. an Int. J.*, vol. 24, no. 5, pp. 1072–1079, 2021, doi: 10.1016/j.jestech.2021.02.009.
- [15] V. P. Parmar and C. Kumbharana, "Comparing Linear Search and Binary Search Algorithms to Search an Element from a Linear List Implemented through Static Array, Dynamic Array and Linked List," *Int. J. Comput. Appl.*, vol. 121, no. 3, pp. 13–17, 2015, doi: 10.5120/21519-4495.
- [16] W. Istiono, "Speed Analysis of Binary Search and Interpolation Search for Searching Identification Numbers on National Identity Cards," *Asian J. Res. Comput. Sci.*, vol. 15, no. 4, pp. 34–41, 2023, doi: 10.9734/ajrcos/2023/v15i4328.
- [17] A. Lin, "Binary search algorithm," *WikiJournal Sci.*, vol. 2, no. 1, pp. 1–13, 2019, doi: 10.15347/wjs/2019.005.
- [18] S. Orhani, "Binary search algorithm for natural number targets," *J. Basic Sci. Technol. J. homepage*, vol. 11, no. 3, pp. 88–92, 2022.
- [19] D. Amato, G. Lo Bosco, and R. Giancarlo, "Standard versus uniform binary search and their variants in learned static indexing: The case of the searching on sorted data benchmarking software platform," *Softw. - Pract. Exp.*, vol. 53, no. 2, pp. 318–346, 2023, doi: 10.1002/spe.3150.
- [20] Kunal, Tushar, and G. Chakraborty, "Decomposed Algorithm for Reducing Time Complexity in Binary Search," *Int. J.*



Eng. Res. Technol., vol. 10, no. 04, pp. 117–121, 2021.

- [21] V. Singh et al., “Bayesian Binary Search,” *Algorithms*, vol. 18, no. 8, pp. 1–27, 2025, doi: 10.3390/a18080452.
- [22] P. Chalermsook, J. Chuzhoy, and T. Saranurak, “Pinning Down the Strong Wilber-1 Bound for Binary Search Trees,” *Theory Comput.*, vol. 19, no. 8, pp. 1–71, 2023, doi: 10.4086/toc.2023.v019a008.
- [23] M. N. Saeed et al., “Empirical Analysis of Quaternary and Binary Search,” *J. Comput. Biomed. Informatics*, vol. 2024, no. Special Issue (ICASET), pp. 1–6, 2024.