JURIKOM (Jurnal Riset Komputer), Vol. 13 No. 5, Oktober 2026
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v15i5.9973
Hal 1530-1540
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

AlgoMetrics: Platform Web Evaluasi Kinerja Algoritma Linear Search dan Binary Search pada Sistem Informasi Akademik

Rohana Labora Munthe*, Amsal Tampubolon, Ronald Hutasoit, Febryan Panjaitan, Lamhot Sitorus

Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Katolik Santo Thomas, Medan, Indonesia
Email: [1*]rohanamunthe29@gmail.com, [2]amsaltampubolon24@gmail.com, [3]ronald.htst@gmail.com,
[4]panjaitanfebryan2@gmail.com, [5]lamhot68@yahoo.com
Email Penulis Korespondensi: rohanamunthe29@gmail.com*
Submitted **07-07-2026**; Accepted **29-10-2026**; Published **31-10-2026**

Abstrak

Sistem Informasi Akademik (SIAKAD) menangani pengelolaan data mahasiswa dalam jumlah besar sehingga diperlukan mekanisme pencarian yang efektif. Penelitian ini bertujuan mengembangkan AlgoMetrics, yaitu platform berbasis web untuk menganalisis dan membandingkan kinerja algoritma *Linear Search* dan *Binary Search* dalam pencarian data mahasiswa. Penelitian menggunakan metode eksperimen dengan dataset berformat JSON yang terdiri atas 1.000, 5.000, dan 10.000 entri. Pengujian dilakukan dalam tiga skenario, yaitu pencarian ketika target berada pada posisi pertama, posisi terakhir, dan ketika data tidak ditemukan. Kinerja kedua algoritma dievaluasi berdasarkan jumlah operasi perbandingan dan waktu eksekusi. Hasil pengujian pada dataset berisi 10.000 entri dengan target di posisi akhir menunjukkan bahwa *Linear Search* melakukan 10.000 perbandingan dengan waktu eksekusi 207,30 µs, sedangkan *Binary Search* hanya memerlukan 14 perbandingan dengan waktu eksekusi 1,00 µs. Ketika data tidak ditemukan, *Linear Search* tetap melakukan 10.000 perbandingan, sementara *Binary Search* membutuhkan 13–14 perbandingan dalam tiga kali pengujian. Secara keseluruhan, *Binary Search* memerlukan lebih sedikit perbandingan dan mencatat waktu eksekusi yang lebih cepat pada skenario yang diuji. Namun, penerapan algoritma ini mensyaratkan dataset telah diurutkan berdasarkan Nomor Induk Mahasiswa (NIM). Dengan demikian, AlgoMetrics dapat dimanfaatkan sebagai sarana eksperimen untuk mengevaluasi dan membandingkan kinerja kedua algoritma pencarian.

Kata Kunci: AlgoMetrics; Evaluasi Kinerja; *Linear Search*; *Binary Search*; Sistem Informasi Akademik.

Abstract

The Academic Information System (SIAKAD) handles the management of large amounts of student data so that an effective search mechanism is needed. This research aims to develop AlgoMetrics, a web-based platform to analyze and compare the performance of *Linear Search* and *Binary Search* algorithms in searching for student data. The research uses an experimental method with a JSON-formatted dataset consisting of 1,000, 5,000, and 10,000 entries. The test was carried out in three scenarios, namely search when the target is in the first position, the last position, and when the data is not found. The performance of the two algorithms is evaluated based on the number of comparison operations and the *execution time*. The test results on the dataset containing 10,000 entries with the target in the final position show that *Linear Search* performs 10,000 comparisons with an *execution time* of 207.30 µs, while Binary Search only requires 14 comparisons with an *execution time* of 1.00 µs. When no data is found, *Linear Search* still performs 10,000 comparisons, while requires 13–14 comparisons in three tests. Overall, *Binary Search* requires fewer comparisons and records faster *execution times* in the tested scenarios. However, the application of this algorithm requires that the datasets have been sorted based on the Student Identification Number (NIM). Thus, AlgoMetrics can be used as an experimental tool to evaluate and compare the performance of the two search algorithms.

Keywords: AlgoMetrics; Performance Evaluation; *Linear Search*; *Binary Search*; Academic Information Systems.

1. PENDAHULUAN

Perkembangan teknologi informasi mendorong institusi pendidikan untuk menerapkan Sistem Informasi Akademik (SIAKAD) sebagai sarana pengelolaan data mahasiswa, dosen, mata kuliah, jadwal perkuliahan, dan nilai. Melalui sistem ini, data dapat diorganisasikan secara terstruktur sehingga memudahkan pengguna dalam mengakses informasi sesuai dengan kebutuhannya. Meskipun demikian, efektivitas sistem informasi akademik tidak hanya bergantung pada kemampuannya dalam menyimpan dan menyajikan informasi, tetapi juga pada efisiensi pengolahan serta pencarian data. Seiring bertambahnya volume data akademik, mekanisme pencarian perlu dirancang secara optimal agar informasi yang diperlukan dapat ditemukan dengan cepat dan akurat [1], [2].

Pencarian data menjadi salah satu tahapan penting dalam proses pengolahan data karena dapat memengaruhi waktu respons aplikasi. Algoritma yang sering digunakan untuk proses tersebut meliputi *Linear Search* dan *Binary Search*. *Linear Search* melakukan pencarian dengan memeriksa setiap elemen secara berurutan hingga data yang diinginkan ditemukan atau seluruh elemen telah diperiksa. Algoritma ini dapat digunakan pada dataset yang tidak

This Journal is licensed under a Creative Commons Attribution 4.0 International License

JURIKOM (Jurnal Riset Komputer), Vol. 13 No. 5, Oktober 2026
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v15i5.9973
Hal 1530-1540
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

terurut, tetapi jumlah pemeriksaan yang diperlukan berpotensi bertambah seiring dengan meningkatnya ukuran dataset. Berbeda dengan itu, *Binary Search* bekerja dengan mempersempit cakupan pencarian menjadi dua bagian pada setiap tahap. Algoritma ini memiliki kompleksitas waktu O(log n), dengan ketentuan bahwa dataset telah diurutkan berdasarkan atribut yang digunakan dalam pencarian. Sementara itu, *Linear Search* memiliki kompleksitas waktu O(n) dalam kasus terburuk. Namun, evaluasi terhadap kinerja aktual kedua algoritma tersebut perlu memperhatikan beberapa aspek, meliputi ukuran dataset, posisi target pencarian, kondisi keterurutan data, serta metode yang digunakan untuk mengukur waktu eksekusi [2], [3].

Berbagai penelitian sebelumnya telah mengkaji perbandingan kinerja algoritma pencarian berdasarkan beragam kondisi pengujian. Surachman et al. [2] menganalisis kinerja *Linear Search* dan *Binary Search* melalui variasi jumlah elemen dengan memperhatikan waktu eksekusi dan kompleksitas algoritma. Zamroni [4] juga mengkaji efisiensi kedua algoritma tersebut. Farizi [5] meneliti proses pencarian informasi mahasiswa berdasarkan kompleksitas teoretis dan hasil pengujian empiris. Selanjutnya, Leana et al. [6] membandingkan waktu eksekusi dan jumlah perbandingan dalam pencarian nama siswa menggunakan struktur data array. Mevia et al. [7] membahas implementasi algoritma pencarian untuk mengolah data nilai mahasiswa berbasis website, sedangkan Amisyah dan Hidayah [8] meneliti waktu eksekusi serta penggunaan memori dalam pencarian novel pada perpustakaan digital. Selain itu, Rahman et al. [9] mengevaluasi kinerja *Linear Search* dan *Binary Search* berdasarkan ukuran data serta kondisi keterurutannya. Secara keseluruhan, penelitian-penelitian tersebut menunjukkan bahwa evaluasi kinerja algoritma pencarian dapat dilakukan dengan mempertimbangkan karakteristik data, metrik pengukuran, dan konteks penerapannya.

Takawale et al. [10] mengkaji perbandingan algoritma *Linear Search* dan *Binary Search* dengan membahas mekanisme kerja serta analisis kompleksitas waktu masing-masing algoritma. Kajian ini memberikan dasar teoretis untuk memahami perbedaan karakteristik dan tingkat efisiensi kedua algoritma pencarian tersebut. Hidayat et al. [11] melakukan evaluasi terhadap kinerja algoritma *Linear Search*, *Binary Search*, dan SQL Query pada sistem informasi pengelolaan data siswa berbasis web dengan menggunakan 1.000 data. Pengujian dilakukan dengan mengukur waktu eksekusi pencarian berdasarkan Nomor Induk Siswa (NIS) . Lutfina et al. [12] menganalisis perbandingan penerapan metode rekursif dan iteratif pada algoritma *Linear Search* berdasarkan waktu akses dan penggunaan memori. Istiono [13] mengevaluasi *Binary Search* dan Interpolation Search dalam pencarian nomor identitas menggunakan variasi ukuran data, dengan jumlah iterasi dan waktu eksekusi sebagai metrik pengukuran. Sementara itu, Amato et al. [14] meneliti sejumlah varian *Binary Search* pada data terurut dengan mempertimbangkan variasi ukuran dataset dan arsitektur komputer. Saputra et al. [15] membandingkan algoritma pencarian dan pengurutan dalam pengolahan data mahasiswa dengan menggunakan dataset berukuran 1.000, 5.000, dan 10.000 data. Secara keseluruhan, penelitian-penelitian tersebut menjadi dasar untuk mengevaluasi kinerja algoritma pencarian dengan mempertimbangkan karakteristik algoritma, ukuran dataset, dan metrik pengujian yang digunakan.

Berdasarkan hasil kajian terdahulu, penelitian ini berfokus pada analisis kinerja *Linear Search* dan *Binary Search* melalui platform berbasis web dengan tiga skenario pencarian, yaitu target berada pada posisi awal, target berada pada posisi akhir, dan data tidak ditemukan. Pengujian dilakukan menggunakan dataset dengan beberapa variasi ukuran melalui pengukuran jumlah operasi perbandingan dan waktu eksekusi. Pendekatan ini bertujuan untuk menganalisis perbedaan kinerja kedua algoritma berdasarkan ukuran dataset dan skenario pencarian yang diterapkan.

Untuk menunjang proses analisis, penelitian ini mengembangkan AlgoMetrics sebagai platform online yang mengintegrasikan pengujian *Linear Search* dan *Binary Search* dalam satu lingkungan pengujian. Data disimpan dalam format JSON, sedangkan dataset yang digunakan pada *Binary Search* terlebih dahulu diurutkan berdasarkan Nomor Induk Mahasiswa (NIM) sebelum proses pencarian dilakukan. Evaluasi kinerja kedua algoritma berfokus pada jumlah operasi perbandingan dan waktu eksekusi pencarian. Kontribusi penelitian ini terletak pada pengembangan platform pencarian berbasis web yang memungkinkan hasil pengujian kedua algoritma diamati dan dibandingkan secara sistematis berdasarkan skenario pengujian yang telah ditentukan.

Berdasarkan uraian tersebut, penelitian ini bertujuan untuk mengembangkan AlgoMetrics sekaligus mengevaluasi dan membandingkan kinerja *Linear Search* dan *Binary Search* dalam proses pencarian data mahasiswa. Pengujian menggunakan dataset berformat JSON dengan ukuran masing-masing 1.000, 5.000, dan 10.000 data, yang mencakup tiga skenario pencarian, yaitu target pada posisi awal, target pada posisi akhir, dan data yang tidak ditemukan. Hasil pengujian dianalisis berdasarkan jumlah operasi perbandingan dan waktu eksekusi untuk mengetahui perbedaan kinerja kedua algoritma pada setiap ukuran dataset dan skenario pencarian. Hasil penelitian ini diharapkan dapat memberikan gambaran empiris mengenai karakteristik kinerja kedua algoritma serta menjadi bahan

This Journal is licensed under a Creative Commons Attribution 4.0 International License

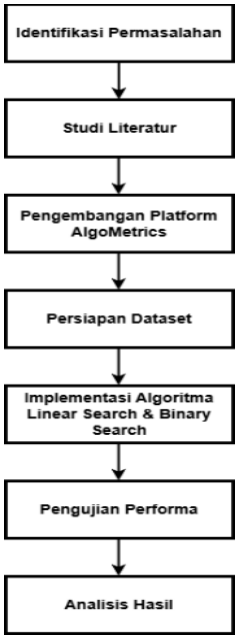
JURIKOM (Jurnal Riset Komputer), Vol. 13 No. 5, Oktober 2026
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v15i5.9973
Hal 1530-1540
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

pertimbangan dalam memilih algoritma pencarian yang sesuai dengan karakteristik data dan kebutuhan Sistem Informasi Akademik.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Pelaksanaan penelitian disusun melalui tahapan yang sistematis untuk memastikan proses pengembangan platform dan evaluasi kinerja algoritma berjalan secara terarah dan terorganisasi. Penelitian ini terdiri atas tujuh tahapan utama, meliputi identifikasi masalah, studi literatur, pengembangan platform AlgoMetrics, persiapan dataset, implementasi algoritma, pengujian kinerja, dan analisis hasil. Setiap tahapan saling berkaitan dalam membentuk alur penelitian yang mendukung evaluasi kinerja algoritma *Linear Search* dan *Binary Search* dalam pencarian data mahasiswa pada Sistem Informasi Akademik. Urutan tahapan penelitian tersebut ditampilkan pada Gambar 1.



Gambar 1. Tahapan Penelitian

Mengacu pada Gambar 1, penelitian ini dilakukan dengan metode eksperimen melalui serangkaian tahapan yang terstruktur. Proses penelitian diawali dengan mengidentifikasi permasalahan terkait efisiensi pencarian data mahasiswa dalam Sistem Informasi Akademik, kemudian dilanjutkan dengan studi literatur untuk membangun landasan teori dan menelaah penelitian terdahulu yang relevan. Tahap selanjutnya mencakup pengembangan platform AlgoMetrics serta penyiapan dataset yang akan digunakan dalam pengujian. [9], [16]. Setelah itu, algoritma *Linear Search* dan *Binary Search* diimplementasikan menggunakan JavaScript dan diuji pada dataset dengan ukuran 1.000, 5.000, dan 10.000 data. Pengujian dilakukan melalui tiga skenario, yaitu pencarian data pada posisi awal, posisi akhir, dan kondisi ketika data tidak ditemukan. Kinerja kedua algoritma diukur berdasarkan jumlah perbandingan (*comparison count*) dan waktu eksekusi (*execution time*). Data hasil pengujian kemudian dianalisis untuk mengetahui dan membandingkan kinerja kedua algoritma pada setiap skenario yang diterapkan [16].

2.2 Implementasi Algoritma *Linear Search* dan *Binary Search*

Penelitian ini menerapkan algoritma pencarian melalui platform berbasis web AlgoMetrics yang dikembangkan sebagai sarana untuk mengevaluasi kinerja algoritma *Linear Search* dan *Binary Search* dalam mencari data mahasiswa berdasarkan atribut Nomor Induk Mahasiswa (NIM). Kedua algoritma diimplementasikan menggunakan bahasa pemrograman JavaScript agar proses pencarian dapat dijalankan secara langsung melalui antarmuka web. Meskipun menggunakan dataset yang sama, kedua algoritma memiliki mekanisme pencarian yang berbeda sesuai dengan karakteristik masing-masing.

Copyright © 2026 The **Rohana Labora Munthe**, Page 1532
This Journal is licensed under a Creative Commons Attribution 4.0 International License

JURIKOM (Jurnal Riset Komputer), Vol. 13 No. 5, Oktober 2026
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v15i5.9973
Hal 1530-1540
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Penerapan *Linear Search* dilakukan dengan memeriksa elemen dataset secara berurutan, dimulai dari elemen pertama hingga data yang dicari ditemukan atau seluruh elemen selesai diperiksa [5]. Metode ini tidak mensyaratkan dataset dalam kondisi terurut, sehingga dapat langsung digunakan pada data mahasiswa yang tersedia [17], [18]. Dalam proses tersebut, sistem membandingkan NIM yang dimasukkan oleh pengguna dengan nilai NIM pada setiap record hingga menemukan data yang sesuai.

Berbeda dengan *Linear Search*, penerapan *Binary Search* mensyaratkan dataset telah diurutkan berdasarkan atribut NIM sebelum pencarian dilakukan. Proses pencarian diawali dengan memeriksa elemen yang terletak di tengah dataset, kemudian membandingkan nilai NIM yang dicari dengan nilai pada elemen tersebut. Hasil perbandingan digunakan untuk menentukan apakah pencarian dilanjutkan pada bagian kiri atau kanan dataset. Proses ini dilakukan secara berulang hingga data yang dicari ditemukan atau tidak ada lagi bagian yang dapat diperiksa. Dengan mekanisme tersebut, *Binary Search* dapat menemukan data melalui jumlah langkah yang lebih sedikit dibandingkan dengan metode pencarian berurutan [19], [20].

Dalam platform AlgoMetrics, pengujian kedua algoritma dilakukan menggunakan ukuran dataset dan skenario pencarian yang setara. *Linear Search* dapat dijalankan secara langsung tanpa mengurutkan dataset terlebih dahulu, sedangkan *Binary Search* memerlukan dataset yang telah disusun berdasarkan NIM. Perbedaan persyaratan tersebut menjadi pertimbangan dalam analisis agar hasil pengujian dapat ditafsirkan sesuai dengan karakteristik masing-masing algoritma. Pengukuran waktu dalam penelitian ini hanya mencakup proses pencarian dan tidak memperhitungkan waktu yang diperlukan untuk mengurutkan dataset pada *Binary Search*.

2.3 Pengujian Performa

Pengujian kinerja dilakukan untuk mengevaluasi dan membandingkan performa algoritma *Linear Search* dan *Binary Search* dalam proses pencarian data mahasiswa melalui platform berbasis web AlgoMetrics. Pengujian dilaksanakan menggunakan dataset dengan ukuran 1.000, 5.000, dan 10.000 data. Terdapat tiga skenario pengujian yang diterapkan, yaitu pencarian data pada posisi awal, posisi akhir, serta kondisi ketika data tidak ditemukan. Evaluasi kinerja kedua algoritma didasarkan pada jumlah perbandingan (*comparison count*) dan waktu eksekusi (*execution time*) [21].

Dataset yang digunakan disimpan dalam format JavaScript Object Notation (JSON) dengan empat atribut, yaitu NIM, nama, program studi, dan alamat email mahasiswa. Dalam pengujian kedua algoritma, NIM berfungsi sebagai atribut utama untuk melakukan pencarian. Sebelum pengujian dilaksanakan, dataset disiapkan dengan menyesuaikan persyaratan masing-masing algoritma. *Linear Search* diterapkan pada dataset tanpa pengurutan awal, sedangkan *Binary Search* menggunakan dataset yang telah diurutkan berdasarkan NIM. Adapun struktur dataset yang digunakan dalam penelitian ini ditampilkan pada Tabel 1.

Tabel 1. Struktur Dataset Mahasiswa

Atribut	Tipe Data	Keterangan
NIM	String	Nomor algoritma ini di uji Induk Mahasiswa sebagai identitas
Nama	String	Nama lengkap mahasiswa.
Program Studi	String	Program studi mahasiswa.
Email	String	Alamat surat elektronik mahasiswa.

Sebelum pengujian dilaksanakan, dataset disiapkan berdasarkan kebutuhan masing-masing algoritma. Linear Search diterapkan pada dataset tanpa pengurutan awal, sedangkan *Binary Search* menggunakan dataset yang telah diurutkan berdasarkan atribut NIM. Penyesuaian ini dilakukan dengan mempertimbangkan karakteristik kedua algoritma. *Linear Search* mampu menelusuri elemen secara berurutan tanpa mengharuskan dataset berada dalam kondisi terurut, sementara *Binary Search* membutuhkan data yang telah diurutkan agar proses pencarian dapat berjalan dengan tepat [21], [14].

Skenario pengujian ditetapkan berdasarkan tiga variasi ukuran dataset, yaitu 1.000, 5.000, dan 10.000 data, serta tiga kondisi pencarian, meliputi target pada posisi awal, target pada posisi akhir, dan target yang tidak ditemukan [13]. Pada setiap ukuran dataset, kedua algoritma diuji menggunakan kondisi pencarian yang sama dengan tetap memperhatikan persyaratan pengurutan pada *Binary Search*. Pengujian pada posisi awal dan akhir bertujuan untuk mengamati pengaruh letak target terhadap jumlah perbandingan dan waktu eksekusi. Sementara itu, pengujian terhadap data yang tidak ditemukan dilakukan untuk mengetahui kinerja algoritma ketika target tidak tersedia dalam dataset.

Pengujian dengan kondisi data tidak ditemukan diterapkan pada ketiga ukuran dataset untuk masing-masing algoritma. Setiap kombinasi ukuran dataset dan algoritma diuji sebanyak tiga kali, kemudian hasilnya dicatat sebagaimana ditampilkan pada Tabel 2.

Copyright © 2026 The **Rohana Labora Munthe**, Page 1533
This Journal is licensed under a Creative Commons Attribution 4.0 International License

JURIKOM (Jurnal Riset Komputer), Vol. 13 No. 5, Oktober 2026
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v15i5.9973
Hal 1530-1540
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Tabel 2. Skenario Pengujian Posisi Awal dan Akhir Dataset

Skenario	Algoritma	Jumlah Data	Posisi Target
S1	*Linear Search*	1.000	Awal
S2	*Binary Search*	1.000	Awal
S3	*Linear Search*	1.000	Akhir
S4	*Binary Search*	1.000	Akhir
S5	*Linear Search*	5.000	Awal
S6	*Binary Search*	5.000	Awal
S7	*Linear Search*	5.000	Akhir
S8	*Binary Search*	5.000	Akhir
S9	*Linear Search*	10.000	Awal
S10	*Binary Search*	10.000	Awal
S11	*Linear Search*	10.000	Akhir
S12	*Binary Search*	10.000	Akhir

Pengujian dilaksanakan menggunakan laptop Lenovo yang dilengkapi prosesor Intel Core i5 dan RAM berkapasitas 16 GB melalui browser pada tautan https://ronald-learn.github.io/perbandingan-linear-binary-search/. Platform AlgoMetrics diakses melalui browser tersebut untuk menguji kinerja algoritma *Linear Search* dan *Binary Search*. Waktu eksekusi diukur menggunakan mekanisme pengukuran waktu yang tersedia pada platform AlgoMetrics, sedangkan jumlah perbandingan dihitung berdasarkan operasi perbandingan yang berlangsung selama proses pencarian.

Selama pengujian, sistem merekam nilai *comparison count* dan *execution time*. *Comparison count* menunjukkan jumlah operasi perbandingan yang dilakukan oleh algoritma hingga target ditemukan. Hasil pengukuran dicatat pada setiap pelaksanaan pencarian, kemudian dibandingkan berdasarkan ukuran dataset, posisi target, dan algoritma yang digunakan. Karena setiap skenario hanya diuji satu kali, nilai waktu eksekusi yang diperoleh mencerminkan hasil dari pelaksanaan tersebut dan tidak merepresentasikan nilai rata-rata dari beberapa pengulangan [22].

Kondisi keterurutan data turut diperhatikan dalam analisis hasil pengujian. *Linear Search* dapat dijalankan tanpa pengurutan awal, sedangkan *Binary Search* membutuhkan dataset yang telah diurutkan berdasarkan NIM. Dengan demikian, waktu yang diperlukan untuk mengurutkan data harus dibedakan dari waktu pencarian dalam pelaporan hasil. Jika pengukuran *execution time* hanya mencakup proses pencarian, waktu pengurutan dataset tidak termasuk dalam hasil pengukuran tersebut sehingga perlu diperhitungkan secara terpisah ketika membandingkan keseluruhan proses pengolahan data. Selanjutnya, hasil pengujian dianalisis berdasarkan jumlah perbandingan dan waktu eksekusi untuk menjelaskan kinerja kedua algoritma dalam setiap skenario pencarian.

2.4 Analisis Hasil

Analisis hasil dilakukan untuk menilai kinerja algoritma *Linear Search* dan *Binary Search* berdasarkan data yang diperoleh dari setiap skenario pengujian. Evaluasi dilakukan dengan membandingkan nilai *execution time* dan *comparison count* pada dataset yang terdiri atas 1.000, 5.000, dan 10.000 data mahasiswa, dengan target pencarian yang terletak di posisi awal maupun akhir dataset. *Execution tim*e digunakan untuk mengukur durasi pencarian, sedangkan *comparison count* menunjukkan banyaknya operasi perbandingan yang dilakukan hingga target ditemukan. Data hasil pengujian kemudian disajikan dalam bentuk tabel dan grafik untuk memudahkan proses interpretasi serta perbandingan kinerja kedua algoritma. Analisis mempertimbangkan perubahan ukuran dataset dan letak target pencarian. Hasil evaluasi tersebut digunakan untuk mengidentifikasi perbedaan kinerja *Linear Search* dan *Binary Search* dalam setiap skenario pengujian, sekaligus menjelaskan pengaruh karakteristik masing-masing algoritma terhadap proses pencarian data mahasiswa pada Sistem Informasi Akademik.

3. HASIL DAN PEMBAHASAN

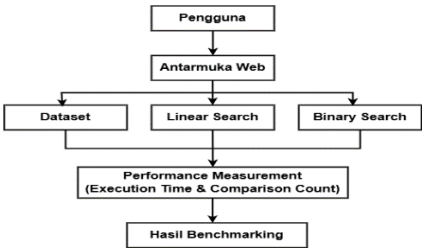
3.1 Implementasi Platform AlgoMetrics

Penelitian ini menghasilkan AlgoMetrics, sebuah platform berbasis web yang dikembangkan sebagai sarana eksperimen untuk mengukur dan menganalisis kinerja algoritma *Linear Search* dan *Binary Search* dalam pencarian data mahasiswa pada Sistem Informasi Akademik. Platform ini menyediakan sejumlah fitur utama, meliputi pengunggahan dataset mahasiswa, pencarian data berdasarkan Nomor Induk Mahasiswa (NIM), pengukuran

This Journal is licensed under a Creative Commons Attribution 4.0 International License

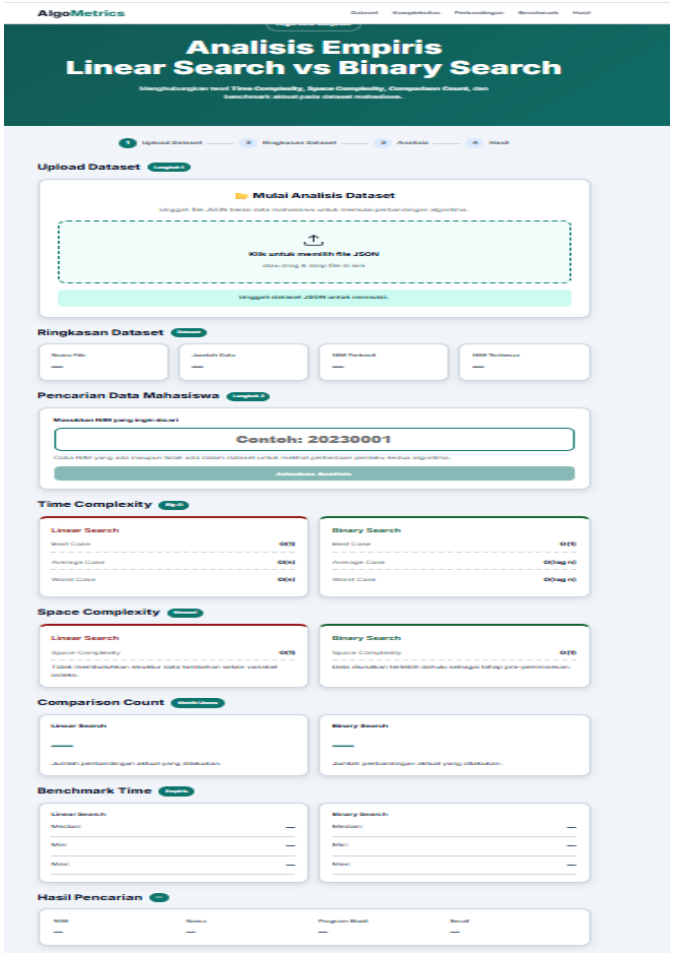
JURIKOM (Jurnal Riset Komputer), Vol. 13 No. 5, Oktober 2026
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v15i5.9973
Hal 1530-1540
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

execution time dan *comparison count*, serta penyajian hasil benchmarking yang digunakan untuk mengevaluasi dan membandingkan kinerja kedua algoritma.



Gambar 2. Arsitektur Fungsional Platform AlgoMetrics

Mengacu pada Gambar 2, penggunaan platform dimulai dengan mengakses antarmuka web, dilanjutkan dengan mengunggah dataset mahasiswa dan menentukan algoritma pencarian yang akan diterapkan. Selanjutnya, sistem memproses pencarian berdasarkan algoritma yang dipilih serta secara otomatis mengukur nilai execution time dan *comparison count*. Hasil benchmarking kemudian ditampilkan untuk menjadi dasar dalam menilai kinerja algoritma yang diuji.



Gambar 3. Halaman Utama Platform AlgoMetrics

Gambar 3 menampilkan antarmuka utama AlgoMetrics setelah dataset berhasil dimasukkan ke dalam sistem. Pada tampilan ini, platform menyajikan ringkasan informasi dataset yang mencakup jumlah data, jumlah atribut, dan ukuran dataset. Pengguna kemudian dapat memasukkan NIM sebagai kata kunci pencarian serta memilih algoritma *Linear Search* atau *Binary Search* yang akan digunakan. Setelah pencarian selesai, sistem menampilkan data

Copyright © 2026 The **Rohana Labora Munthe**, Page 1535
This Journal is licensed under a Creative Commons Attribution 4.0 International License

JURIKOM (Jurnal Riset Komputer), Vol. 13 No. 5, Oktober 2026
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v15i5.9973
Hal 1530-1540
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

mahasiswa yang berhasil ditemukan beserta hasil pengukuran *execution time*, *comparison count*, *time complexity*, dan *space complexity*. Informasi tersebut dapat digunakan untuk mengevaluasi dan membandingkan kinerja kedua algoritma secara langsung dalam lingkungan pengujian yang sama.

3.2 Hasil Pengujian

Pengujian kinerja algoritma *Linear Search* dan *Binary Search* dilaksanakan menggunakan platform AlgoMetrics dengan dataset yang berisi 1.000, 5.000, dan 10.000 data mahasiswa. Proses pengujian meliputi tiga skenario, yaitu pencarian target pada posisi awal, posisi akhir, dan kondisi ketika data tidak ditemukan. Evaluasi dilakukan berdasarkan jumlah operasi perbandingan (*comparison count*) serta waktu eksekusi (*execution time*). Dataset yang digunakan untuk *Binary Search* terlebih dahulu diurutkan berdasarkan Nomor Induk Mahasiswa (NIM), sedangkan *Linear Search* menggunakan urutan data yang telah ditetapkan dalam pengujian. Posisi awal menunjukkan elemen pertama yang diperiksa, sementara posisi akhir mengacu pada elemen terakhir dalam urutan pemeriksaan. Pengujian pada kedua posisi tersebut bertujuan untuk mengetahui pengaruh letak target terhadap jumlah perbandingan dan waktu eksekusi masing-masing algoritma.

Tabel 3. Hasil Pengujian Algoritma *Linear Search* dan *Binary Search* pada Posisi Awal dan Akhir Dataset

Jumlah Data	Posisi	Algoritma	Jumlah Perbandingan	Waktu (µs)
1.000	Awal	*Linear Search*	1	0,00
1.000	Awal	*Binary Search*	9	0,60
1.000	Akhir	*Linear Search*	1.000	15,80
1.000	Akhir	*Binary Search*	10	0,60
5.000	Awal	*Linear Search*	1	0,00
5.000	Awal	*Binary Search*	12	0,80
5.000	Akhir	*Linear Search*	5.000	80,40
5.000	Akhir	*Binary Search*	13	1,00
10.000	Awal	*Linear Search*	1	0,00
10.000	Awal	*Binary Search*	13	0,80
10.000	Akhir	*Linear Search*	10.000	207,30
10.000	Akhir	*Binary Search*	14	1,00

Berdasarkan Tabel 3, jumlah perbandingan yang dilakukan oleh *Linear Search* bertambah seiring dengan meningkatnya jumlah data ketika elemen yang dicari terletak pada posisi akhir. Pada dataset yang berisi 1.000, 5.000, dan 10.000 data, algoritma *Linear Search* masing-masing melakukan 1.000, 5.000, dan 10.000 perbandingan. Sementara itu, *Binary Search* hanya memerlukan 9–14 perbandingan pada seluruh skenario pengujian. Hasil tersebut memperlihatkan bahwa *Binary Search* membutuhkan jumlah perbandingan yang lebih sedikit dibandingkan *Linear Search*, dengan syarat dataset telah diurutkan berdasarkan NIM sebelum proses pencarian dilakukan.

Tabel 4. Hasil Pengujian ketika Data Tidak Ditemukan

Jumlah Data	Pengujian	Algoritma	Jumlah Perbandingan	Median (µs)	Minimum (µs)	Maksimum (µs)
1.000	1	*Linear Search*	1.000	15,80	14,0	24,40
1.000	1	*Binary Search*	9	0,60	0,40	2,00
1.000	2	*Linear Search*	1.000	14,80	13,00	21,00
1.000	2	*Binary Search*	9	0,60	0,40	2,80
1.000	3	*Linear Search*	1.000	14,20	13,20	21,80
1.000	3	*Binary Search*	10	0,80	0,60	2,40
5.000	1	*Linear Search*	5.000	72,40	65,80	114,40
5.000	1	*Binary Search*	12	0,80	0,60	1,60
5.000	2	*Linear Search*	5.000	71,80	66,60	114,20
5.000	2	*Binary Search*	12	1,00	0,60	2,80
5.000	3	*Linear Search*	5.000	79,80	66,20	177,20
5.000	3	*Binary Search*	13	1,20	0,60	2,60
10.000	1	*Linear Search*	10.000	244,90	234,60	337,60
10.000	1	*Binary Search*	13	1,00	0,80	2,00
10.000	2	*Linear Search*	10.000	73,40	67,40	94,20
10.000	2	*Binary Search*	14	0,80	0,40	1,40
10.000	3	*Linear Search*	10.000	73,00	66,80	102,80
10.000	3	*Binary Search*	14	0,80	0,40	1,40

This Journal is licensed under a Creative Commons Attribution 4.0 International License

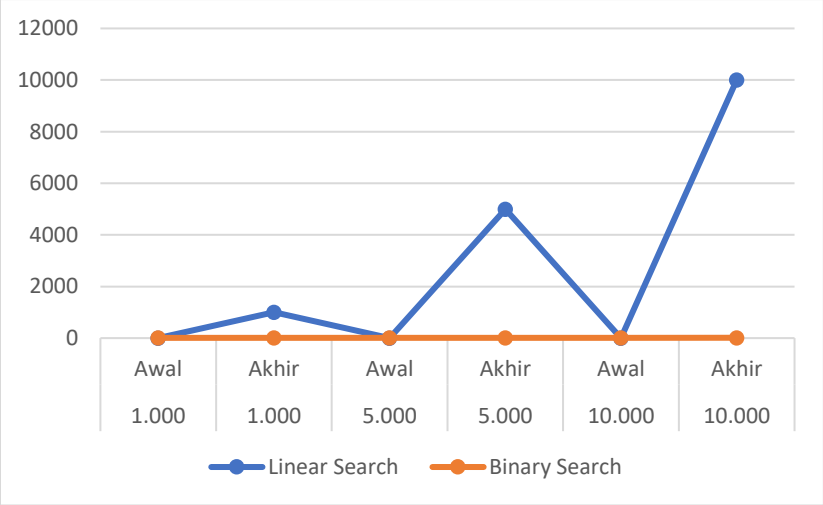
JURIKOM (Jurnal Riset Komputer), Vol. 13 No. 5, Oktober 2026
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v15i5.9973
Hal 1530-1540
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Tabel 4 menyajikan hasil pengujian pada kondisi ketika NIM yang dicari tidak terdapat dalam dataset. Algoritma *Linear Search* melakukan perbandingan sebanyak jumlah data dalam dataset, yakni 1.000, 5.000, dan 10.000 perbandingan sesuai dengan ukuran dataset. Sementara itu, *Binary Search* memerlukan 9–10 perbandingan untuk dataset berukuran 1.000 data, 12–13 perbandingan pada dataset 5.000 data, serta 13–14 perbandingan pada dataset 10.000 data. Hasil pengujian ini memperlihatkan bahwa *Binary Search* membutuhkan jumlah perbandingan yang relatif lebih sedikit daripada *Linear Search* pada seluruh ukuran dataset yang diuji.

Perbedaan waktu eksekusi yang diperoleh pada setiap pengujian menunjukkan bahwa hasil pengukuran dapat mengalami perubahan meskipun menggunakan ukuran dataset yang sama. Pada dataset berisi 10.000 data, median waktu eksekusi *Linear Search* dalam tiga kali pengujian masing-masing tercatat sebesar 244,90 µs, 73,40 µs, dan 73,00 µs. Adapun *Binary Search* menghasilkan median waktu eksekusi sebesar 1,00 µs, 0,80 µs, dan 0,80 µs. Hasil tersebut menunjukkan bahwa *Binary Search* memiliki waktu eksekusi yang lebih rendah pada ketiga pengujian. Meskipun demikian, variasi waktu eksekusi pada *Linear Search* menunjukkan bahwa evaluasi perbandingan kinerja perlu mempertimbangkan hasil pengukuran dari setiap pengujian.

3.3 Analisis Jumlah Perbandingan (*Comparison Count*)

Analisis *comparison count* bertujuan untuk membandingkan jumlah operasi perbandingan yang dilakukan oleh algoritma *Linear Search* dan *Binary Search* pada dataset dengan ukuran 1.000, 5.000, dan 10.000 data. Pengujian dilakukan dalam tiga skenario, yaitu pencarian pada posisi awal, posisi akhir, serta kondisi ketika data yang dicari tidak ditemukan. Hasil perbandingan jumlah operasi ditampilkan pada Gambar 4, sementara hasil pengujian untuk kondisi data tidak ditemukan disajikan dalam Tabel 4.



Gambar 4. Perbandingan *Comparison Count* Algoritma *Linear Search* dan *Binary Search*

Hasil pengujian yang disajikan pada Tabel 3 menunjukkan bahwa jumlah perbandingan algoritma *Linear Search* dipengaruhi oleh letak data yang dicari. Apabila target berada pada posisi awal, algoritma ini hanya memerlukan satu perbandingan untuk setiap ukuran dataset, yaitu 1.000, 5.000, maupun 10.000 data. Namun, jika target terletak pada posisi akhir, jumlah perbandingan bertambah sesuai dengan ukuran dataset, yakni masing-masing 1.000, 5.000, dan 10.000 perbandingan.

Sementara itu, *Binary Search* menghasilkan jumlah perbandingan yang relatif lebih rendah dibandingkan *Linear Search*. Pada skenario pencarian di posisi awal dan akhir, algoritma ini memerlukan 9–14 perbandingan untuk seluruh ukuran dataset yang diuji. Temuan tersebut menunjukkan bahwa pertambahan ukuran dataset tidak menyebabkan peningkatan jumlah perbandingan *Binary Search* sebesar yang terjadi pada *Linear Search*. Hal ini berkaitan dengan mekanisme kerja *Binary Search* yang membagi ruang pencarian menjadi sekitar setengah pada setiap tahap pencarian.

Pola serupa juga terlihat pada hasil pengujian ketika data tidak ditemukan, sebagaimana ditampilkan dalam Tabel 4. Pada kondisi tersebut, *Linear Search* melakukan 1.000, 5.000, dan 10.000 perbandingan sesuai dengan ukuran dataset. Sebaliknya, *Binary Search* hanya membutuhkan 9–10 perbandingan pada dataset berisi 1.000 data, 12–13 perbandingan pada dataset 5.000 data, serta 13–14 perbandingan pada dataset 10.000 data.

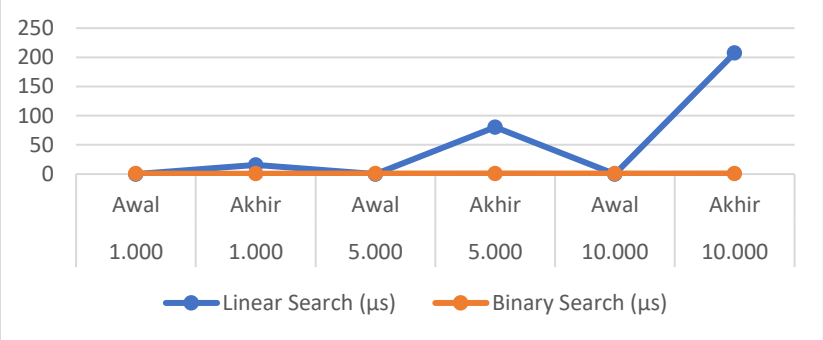
Copyright © 2026 The **Rohana Labora Munthe**, Page 1537
This Journal is licensed under a Creative Commons Attribution 4.0 International License

JURIKOM (Jurnal Riset Komputer), Vol. 13 No. 5, Oktober 2026
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v15i5.9973
Hal 1530-1540
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Secara umum, hasil pengujian memperlihatkan bahwa *Binary Search* memerlukan jumlah operasi perbandingan yang lebih sedikit dibandingkan *Linear Search* pada skenario pencarian di posisi akhir maupun ketika data tidak ditemukan. Namun, sebelum pencarian dilakukan, *Binary Search* mengharuskan dataset diurutkan terlebih dahulu berdasarkan atribut yang digunakan sebagai kunci pencarian, yaitu NIM.

3.4 Analisis Waktu Eksekusi (*Execution Time*)

Analisis waktu eksekusi (*execution time*) bertujuan untuk membandingkan durasi yang diperlukan oleh algoritma *Linear Search* dan *Binary Search* dalam menemukan data mahasiswa pada dataset yang terdiri atas 1.000, 5.000, dan 10.000 data. Hasil pengujian untuk pencarian pada posisi awal dan akhir disajikan pada Gambar 5, sedangkan hasil pengujian ketika data tidak ditemukan ditampilkan pada Tabel 4. Pada Gambar 5, perbedaan rentang waktu eksekusi kedua algoritma menyebabkan nilai *Binary Search* tampak berdekatan dengan sumbu horizontal. Dengan demikian, perbandingan waktu eksekusi perlu disertai dengan peninjauan terhadap nilai numerik yang disajikan dalam Tabel 3.



Gambar 5. Perbandingan *Execution Time* Algoritma *Linear Search* dan *Binary Search*

Hasil pengujian pada Tabel 3 memperlihatkan bahwa waktu eksekusi *Linear Search* dipengaruhi oleh posisi data yang dicari serta ukuran dataset. Pada skenario pencarian di posisi awal, waktu eksekusi yang tercatat adalah 0,00 µs untuk seluruh ukuran dataset. Angka tersebut menunjukkan bahwa durasi proses pencarian sangat singkat sehingga dibulatkan menjadi 0,00 µs, bukan berarti pencarian berlangsung tanpa memerlukan waktu. Sementara itu, pada posisi akhir, waktu eksekusi *Linear Search* meningkat dari 15,80 µs pada dataset berukuran 1.000 data menjadi 80,40 µs pada dataset 5.000 data dan 207,30 µs pada dataset 10.000 data.

Di sisi lain, *Binary Search* menghasilkan waktu eksekusi dalam rentang 0,60–1,00 µs pada skenario pencarian di posisi awal dan akhir. Hasil tersebut memperlihatkan bahwa algoritma *Binary Search* memiliki waktu eksekusi yang relatif rendah pada skenario pengujian yang dilakukan. Perbedaan waktu eksekusi ini sejalan dengan jumlah perbandingan yang lebih sedikit, meskipun durasi pencarian juga dapat dipengaruhi oleh kondisi perangkat serta proses pengukuran.

Untuk skenario ketika data tidak ditemukan, Tabel 4 menyajikan hasil dari tiga kali pengujian pada setiap ukuran dataset. Pada algoritma *Linear Search*, median waktu eksekusi untuk dataset berisi 1.000 data berada pada rentang 14,20–15,80 µs, sedangkan pada dataset 5.000 data berada pada rentang 71,80–79,80 µs. Adapun pada dataset 10.000 data, median waktu eksekusi berkisar antara 73,00–244,90 µs. Perbedaan tersebut menunjukkan bahwa hasil pengukuran waktu eksekusi dapat bervariasi antarpengujian meskipun ukuran dataset yang digunakan sama. Oleh sebab itu, hasil pengujian perlu dianalisis dengan memperhatikan variasi pengukuran yang terjadi.

Pada *Binary Search*, median waktu eksekusi dalam kondisi data tidak ditemukan berkisar antara 0,60–0,80 µs untuk dataset 1.000 data, 0,80–1,20 µs untuk dataset 5.000 data, serta 0,80–1,00 µs untuk dataset 10.000 data. Rentang nilai tersebut menunjukkan bahwa waktu eksekusi *Binary Search* tetap relatif rendah pada seluruh ukuran dataset yang diuji.

Secara keseluruhan, *Binary Search* menghasilkan waktu eksekusi yang lebih rendah dibandingkan *Linear Search* pada skenario pencarian di posisi akhir dan ketika data tidak ditemukan. Meskipun demikian, variasi hasil pengukuran *Linear Search*, khususnya pada dataset 10.000 data, memperlihatkan bahwa waktu eksekusi tidak selalu mengalami peningkatan secara konsisten dalam setiap pengujian. Dengan demikian, evaluasi kinerja kedua algoritma perlu memperhitungkan jumlah perbandingan dan variasi waktu eksekusi, bukan hanya berpatokan pada satu hasil pengukuran.

3.5 Pembahasan

Copyright © 2026 The **Rohana Labora Munthe**, Page 1538
This Journal is licensed under a Creative Commons Attribution 4.0 International License

JURIKOM (Jurnal Riset Komputer), Vol. 13 No. 5, Oktober 2026
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v15i5.9973
Hal 1530-1540
https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Hasil pengujian memperlihatkan adanya perbedaan kinerja antara algoritma *Linear Search* dan *Binary Search* dalam proses pencarian data mahasiswa. Algoritma *Linear Search* menelusuri data secara berurutan sehingga jumlah perbandingan yang dilakukan bergantung pada letak data yang dicari. Jika target berada pada posisi awal, algoritma ini hanya memerlukan satu perbandingan. Sebaliknya, apabila target terletak di posisi akhir atau tidak ditemukan, seluruh elemen dalam dataset harus diperiksa.

Sementara itu, *Binary Search* membagi ruang pencarian menjadi dua bagian pada setiap tahap, sehingga jumlah perbandingan yang diperlukan lebih sedikit pada ukuran dataset yang diuji. Akan tetapi, algoritma ini mengharuskan dataset diurutkan berdasarkan NIM sebelum proses pencarian dimulai. Dengan demikian, perbandingan kinerja kedua algoritma dalam penelitian ini dilakukan dengan memperhatikan proses pencarian serta persyaratan dataset pada masing-masing algoritma.

Gambar 4 dan Gambar 5 memperlihatkan perbedaan jumlah perbandingan dan waktu eksekusi kedua algoritma pada skenario pencarian di posisi awal dan akhir. Adapun hasil yang disajikan dalam Tabel 4 menunjukkan bahwa *Binary Search* juga memerlukan jumlah perbandingan yang lebih sedikit ketika data tidak ditemukan. Berdasarkan waktu eksekusi, *Binary Search* mencatat durasi yang relatif rendah, sedangkan hasil pengujian *Linear Search* menunjukkan adanya variasi waktu, terutama pada dataset yang berisi 10.000 data.

Secara keseluruhan, *Binary Search* menunjukkan kinerja pencarian yang lebih baik berdasarkan jumlah perbandingan dan waktu eksekusi dalam skenario yang diuji. Meskipun demikian, hasil tersebut perlu ditafsirkan dengan mempertimbangkan bahwa dataset telah diurutkan berdasarkan NIM dan waktu yang diperlukan untuk proses pengurutan tidak termasuk dalam pengukuran. Oleh karena itu, pemilihan algoritma perlu disesuaikan dengan kondisi dataset dan kebutuhan pencarian dalam sistem informasi akademik.

4. KESIMPULAN

Berdasarkan proses pengembangan dan pengujian AlgoMetrics, dapat disimpulkan bahwa platform berbasis web ini mampu digunakan untuk mengevaluasi serta membandingkan kinerja algoritma *Linear Search* dan *Binary Search* dalam mencari data mahasiswa pada dataset berformat JSON. Pengujian dilakukan menggunakan dataset yang terdiri atas 1.000, 5.000, dan 10.000 data melalui tiga skenario, yaitu pencarian pada posisi awal, posisi akhir, dan kondisi ketika data tidak ditemukan. Hasil pengujian memperlihatkan bahwa jumlah perbandingan pada *Linear Search* dipengaruhi oleh posisi data yang dicari. Pada posisi awal, algoritma ini hanya memerlukan satu perbandingan. Sebaliknya, pada posisi akhir dan ketika data tidak ditemukan, jumlah perbandingan mencapai keseluruhan jumlah data dalam dataset. Sementara itu, *Binary Search* memerlukan jumlah perbandingan yang lebih sedikit, yakni sekitar 9–14 perbandingan pada seluruh skenario pengujian. Ditinjau dari waktu eksekusi, *Binary Search* menghasilkan durasi pencarian yang lebih rendah dibandingkan *Linear Search* pada skenario posisi akhir dan ketika data tidak ditemukan. Namun, hasil pengukuran *Linear Search* menunjukkan variasi waktu eksekusi, khususnya pada dataset berukuran 10.000 data. Oleh sebab itu, evaluasi kinerja kedua algoritma perlu dilakukan dengan mempertimbangkan jumlah perbandingan dan hasil pengukuran waktu eksekusi secara bersamaan. Secara keseluruhan, penelitian ini menunjukkan bahwa *Binary Search* lebih efisien dalam proses pencarian berdasarkan skenario yang telah diuji. Meskipun demikian, penerapan algoritma tersebut memerlukan dataset yang telah diurutkan berdasarkan NIM, sementara waktu pengurutan belum dimasukkan dalam pengukuran. Dengan demikian, AlgoMetrics dapat dimanfaatkan sebagai sarana evaluasi dan pembelajaran untuk memahami perbedaan kinerja kedua algoritma dalam pencarian data mahasiswa.

REFERENCES

[1] A. G. Mannang, "Implementasi Sistem Informasi Berbasis Web untuk Pengelolaan Data Akademik," *Jurnal Sistem Informasi,* vol. 1, no. 1, pp. 16-22, 2025.

[2] R. I. Surachman, F. Supriadi, A. Saeppani, and F. Mahardika, "Perbandingan Kinerja Algoritma *Linear Search* dan *Binary Search* dalam Pencarian Data," *Infoman's,* vol. 19, no. 2, 2025.

[3] P. Cahya, "Analisis Komprehensif Algoritma *Binary Search*: Kompleksitas, Implementasi, dan Perbandingan Kinerja," *Jurnal Inovasi Informatika dan Bisnis Digital, vol. 1, no. 3, pp. 80–82, 2026, doi: 10.59934/jiibd.v1i3.2528*.

[4] A. Zamroni, "Perbandingan efisiensi algoritma linear search dengan algoritma *binary search*," *Maliki Interdisciplinary Journal*, vol. 3, pp. 86-92, 2025.

Copyright © 2026 The **Rohana Labora Munthe**, Page 1539
This Journal is licensed under a Creative Commons Attribution 4.0 International License

[5] F. A. Farizi, A. Amarudin, and R. I. Borman, "Perbandingan Kinerja *Linear Search* dan *Binary Search* pada Pencarian Data Mahasiswa Berdasarkan Kompleksitas Teoretis dan Pengujian Empiris," *Jurnal Manajemen Informatika Medicom,* vol. 13, no. 2, pp. 56-65, 2026.

[6] S. A. Leana, M. A. P. Ginting, M. B. Izdihar, T. S. A. Pratama, D. A. A. Manik, and I. Gunawanl, "Perbandingan Efisiensi Linear dan Binary Search dalam Pencarian Nama Siswa pada Struktur Data Array," *Jurnal Riset Sistem Informasi dan Aplikasi Komputer,* vol. 1, no. 2, pp. 45-50, 2025.

[7] N. A. O. Mevia, Y. K. Marbun, M. D. Putri, and Y. R. Sitompul, "Perbandingan Algoritma Divide and Conquer dan Searching pada Pengolahan Data Nilai Mahasiswa Berbasis Web," *Teknik: Jurnal Ilmu Teknik dan Informatika,* vol. 6, no. 1, pp. 60-79, 2026, doi: 10.51903/teknik.v1i1.1145.

[8] H. Amisyah and A. K. Hidayah, "Analisis Performa Waktu Eksekusi dan Penggunaan Memori Algoritma *Linear Search* dan *Binary Search* pada Sistem Pencarian Buku Perpustakaan Digital," *Journal of Information System Research (JOSH),* vol. 7, no. 4, pp. 1257–1266, 2026, doi: 10.47065/josh.v7i4.10067.

[9] D. I. J. and D. S. G. A. Rahman, "Algoritma *Linear Search* dan *Binary Search* Berdasarkan Ukuran dan Kondisi Keterurutan Data," *Infoman's,* vol. 19, no. 2, Nov 2025.

[10] A. Kurhade and N. N. Takawale, "Comparative Study of Searching Algorithm: *Linear Search* and *Binary Search,*" International Journal of Advanced Research in Computer and Communication Engineering," vol. 13, no. 5, May 2024, doi: 10.17148/IJARCCE.2024.13506.

[11] S. S. W. Hidayat, A. Andriani, A. D. Indriyanti, and E. H. A. Prastyo, "Evaluasi Kinerja *Linear Search, Binary Search*, dan SQL Query pada Sistem Informasi Berbasis Web," *TIN: Terapan Informatika Nusantara,* vol. 7, no. 1, pp. 662–674, 2026, doi: 10.47065/tin.v7i1.10121.

[12] E. Lutfina, N. Inayati, and G. W. Saraswati, "Analisis perbandingan kinerja metode rekursif dan metode iteratif dalam algoritma *linear search*. Komputika," *Jurnal Sistem Komputer,* vol. 11, no. 2, pp. 143–150, 2022, doi: 10.34010/komputika.v11i2.5493.

[13] W. Istiono, "Speed analysis of *binary search* and interpolation search for searching identification numbers on national identity cards," *Asian Journal of Research in Computer Science,* vol. 15, no. 4, pp. 34-41, 2023, doi: 10.9734/ajrcos/2023/v15i4328.

[14] 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," *Software: Practice and Experience,* vol. 53, no. 2, pp. 318–346, 2023, doi: 10.1002/spe.3150.

[15] M. S. A. M. M. R. S. A. D. &. W. J. Saputra, "Comparative analysis of searching and sorting algorithms in student data processing," *International Journal of Education, Information Technology and Others (IJEIT),* vol. 8, no. 3, pp. 148–157, 2025.

[16] T. Dobravec, "Automating algorithm experiments with ALGator: From problem modeling to reproducible results," *Software: Practice and Experience,* vol. 56, no. 1, pp. 26-41, 2026, doi: 10.1002/spe.70020.

[17] N. N. Diana, "Penerapan Algoritma Sequential Search pada Sistem Pencarian Data Naskah Dinas," *Maliki Interdisciplinary Journal (MIJ),* vol. 2, no. 12, p. 941–949, 2024.

[18] H. Lubis and M. F. Ramadhan, "Sistem informasi pengelolaan data inventory menggunakan algoritma sequential search pada bengkel mobil BPM Autowork," *Jurnal Sistem Informasi Universitas Suryadarma,* vol. 11, no. 1, p. 163–172, 2024, doi: 10.35968/jsi.v11i2.1250.

[19] J. Tabea, K. Santa, and A. Hasibuan, "Penerapan Algoritma Binary Search pada Aplikasi JDIH Universitas Negeri Manado," *REMIK: Riset dan E-Jurnal Manajemen Informatika Komputer,* vol. 10, no. 1, p. 127–134, 2026, doi: 10.33395/remik.v10i1.15620.

[20] N. A. Purba, R. Dewi, T. D. Kartika, and Safrizal, Purba, "Implementasi Algoritma *Binary Search* pada Sistem Temu Balik Informasi Berbasis Web dengan Input File Excel," *JISTI: Jurnal Informatika dan Sistem Teknologi Informasi,* vol. 2, no. 1, pp. 35-40, 2026.

[21] N. N. Yasmin, L. Sofia, A. R. P. Sabrina, A. A.-Z. Putri, and I. P. Pujiono, "Interpolation Searching Algorithm vs Algoritma Pencarian Tradisional: Analisis Efisiensi Memori dan Waktu Komputasi," *Jurnal Sistem Informasi dan Sistem Komputer (SIMKOM),* vol. 10, no. 2, p. 212–223, 2025, doi: 10.51717/simkom.v10i2.857.

[22] S. Luan, Z. Gu, A. Saremi, L. Freidovich, L. Jiang, and S. Wan, "Timing Performance Benchmarking of Out-of-Distribution Detection Algorithms," *Journal of Signal Processing Systems,* vol. 95, no. 1, p. 1355–1370, 2023, doi: 10.1007/s11265-023-01852-0.

This Journal is licensed under a Creative Commons Attribution 4.0 International License