



Evaluasi Kinerja Linear Search, Binary Search, dan SQL Query pada Sistem Informasi Berbasis Web

Sri Shinta Wahyuni Hidayat*, Anita Andriani, Aries Dwi Indriyanti, Edwin Hari Agus Prastyo

Fakultas Teknologi Informasi, Program Studi Teknik Informatika, Universitas Hasyim Asy'ari, Jombang, Indonesia

Email: ¹srihidayat@mhs.unhasy.ac.id, ²anitaandriani@unhasy.ac.id, ³ariesdwiindriyanti@unhasy.ac.id,

⁴edwinhariagusprastyo@unhasy.ac.id

Email Penulis Korespondensi: srihidayat@mhs.unhasy.ac.id

Abstrak—Pengelolaan data pada sistem informasi pendidikan memerlukan mekanisme pencarian yang mampu menyediakan akses informasi secara cepat dan akurat. Meskipun berbagai algoritma pencarian telah banyak diterapkan pada sistem informasi berbasis web, kajian yang membandingkan mekanisme pencarian pada level aplikasi dengan pencarian yang memanfaatkan fitur optimasi basis data masih relatif terbatas. Penelitian ini bertujuan untuk merancang dan mengembangkan sistem informasi manajemen data santri berbasis web serta mengevaluasi kinerja metode *Linear Search*, *Binary Search*, dan *SQL Query* dalam proses pencarian data. Sistem dikembangkan menggunakan framework Laravel dan basis data MySQL dengan pendekatan *Waterfall* yang meliputi analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. Pengujian dilakukan menggunakan 1.000 data santri dengan skenario pencarian berdasarkan Nomor Induk Santri (NIS). Evaluasi kinerja difokuskan pada pengukuran waktu eksekusi pencarian yang dilakukan pada lingkungan pengujian yang sama untuk setiap metode. Hasil penelitian menunjukkan bahwa *SQL Query* menghasilkan waktu eksekusi rata-rata sebesar 0,0065 detik, lebih cepat dibandingkan *Binary Search* sebesar 0,0248 detik dan *Linear Search* sebesar 0,0369 detik. Hasil tersebut menunjukkan bahwa pemanfaatan mekanisme *indexing* dan *optimasi query* pada sistem manajemen basis data mampu meningkatkan efisiensi pencarian dibandingkan implementasi algoritma pencarian pada level aplikasi. Namun demikian, hasil penelitian ini masih terbatas pada skala data sebanyak 1.000 record sehingga belum dapat digeneralisasikan pada lingkungan data berskala besar. Kontribusi penelitian ini adalah memberikan evaluasi empiris mengenai karakteristik kinerja metode pencarian pada sistem informasi berbasis web sebagai dasar pertimbangan dalam pemilihan strategi pencarian data yang sesuai dengan kebutuhan aplikasi.

Kata Kunci: *Linear Search; Binary Search; SQL Query; Query Optimization; Sistem Informasi Berbasis We*

Abstract—Data management in educational information systems requires efficient search mechanisms to provide fast and accurate access to information. Although various search algorithms have been widely implemented in web-based information systems, studies comparing application-level search algorithms with database-level search mechanisms that utilize query optimization features remain limited. This study aims to design and develop a web-based student data management information system and evaluate the performance of *Linear Search*, *Binary Search*, and *SQL Query* methods in data retrieval processes. The system was developed using the Laravel framework and MySQL database through the Waterfall approach, consisting of requirements analysis, system design, implementation, and testing stages. Performance testing was conducted using 1,000 student records with search scenarios based on Student Identification Numbers (NIS). The evaluation focused on measuring search execution time under the same testing environment for each method. The results showed that *SQL Query* achieved the best performance with an average execution time of 0.0065 seconds, outperforming *Binary Search* at 0.0248 seconds and *Linear Search* at 0.0369 seconds. These findings indicate that database indexing and query optimization mechanisms can significantly improve search efficiency compared to application-level search algorithms. However, the results are limited to a dataset of 1,000 records and cannot yet be generalized to large-scale data environments. This study contributes empirical evidence regarding the performance characteristics of different search methods in web-based information systems and provides practical insights for selecting appropriate data retrieval strategies.

Keywords: *Linear Search; Binary Search; SQL Query; Query Optimization; Web-Based Information System*

1. PENDAHULUAN

Pengelolaan data pada sistem informasi berbasis web tidak hanya ditentukan oleh desain antarmuka dan kelengkapan fitur, tetapi juga oleh kemampuan sistem dalam melakukan pencarian data secara cepat dan efisien. Pada lingkungan pendidikan dengan volume data pengguna yang terus meningkat, kinerja proses pencarian menjadi salah satu faktor penting yang mempengaruhi kualitas layanan administrasi. Berbagai penelitian menunjukkan bahwa optimasi akses data melalui teknik pengindeksan (*indexing*) dan optimasi query memiliki peran penting dalam meningkatkan kinerja sistem informasi berbasis web, terutama ketika volume data semakin besar (Saidu et al., 2024; Sun et al., 2024). Oleh karena itu, evaluasi terhadap mekanisme pencarian data menjadi aspek penting dalam pengembangan sistem informasi yang memerlukan akses data secara cepat, akurat, dan efisien.

Pondok pesantren merupakan lembaga pendidikan yang mengelola berbagai data administratif, seperti data identitas santri, asal daerah, data wali, dan administrasi pendidikan lainnya. Peningkatan jumlah santri menyebabkan kompleksitas pengelolaan data administrasi semakin tinggi, sehingga proses pencarian dan pengelolaan informasi menjadi lebih sulit dilakukan secara manual. Mengelola data santri secara manual memiliki banyak kekurangan, seperti proses pencarian data yang lambat, risiko kehilangan dokumen, data yang bisa saja duplikat, dan efisiensi kerja administrasi yang rendah (Kunal Shelke et al., 2025). Kondisi ini menjadi tantangan bagi lembaga pendidikan seperti pondok pesantren dalam meningkatkan mutu pelayanan administrasi. Oleh karena itu, penggunaan sistem informasi untuk mengelola data santri yang berbasis web menjadi solusi yang tepat untuk mendukung proses administrasi agar lebih efektif dan efisien (Lontaan & Sinadia, 2024).



Proses pengelolaan data santri di Pondok Pesantren Roudlotul Quran Darul Falah III masih dilakukan secara konvensional sehingga pencarian informasi membutuhkan waktu yang relatif lama ketika jumlah data terus bertambah. Permasalahan tersebut tidak hanya berkaitan dengan digitalisasi data, tetapi juga mencakup efisiensi mekanisme pencarian yang digunakan oleh sistem. Dalam sistem informasi berbasis web, kinerja pencarian sangat dipengaruhi oleh metode yang digunakan untuk menemukan data, baik melalui algoritma pencarian pada tingkat aplikasi maupun melalui mekanisme optimasi yang tersedia pada sistem manajemen basis data (DBMS) (Fan et al., 2025). Oleh karena itu, analisis terhadap berbagai pendekatan pencarian data menjadi penting untuk menentukan metode yang paling sesuai dalam mendukung kebutuhan pengelolaan data santri.

Linear Search masih banyak digunakan pada proses pencarian data yang tidak terurut dan pada koleksi data berukuran kecil karena implementasinya sederhana dan tidak memerlukan proses pengurutan data sebelumnya. Namun, algoritma ini memiliki kompleksitas waktu $O(n)$, sehingga waktu pencarian meningkat secara linier terhadap jumlah data yang diproses dan menyebabkan penurunan performa pada dataset berukuran besar (Lutfina et al., 2022).

Binary Search menawarkan kompleksitas waktu $O(\log n)$ sehingga lebih efisien dibandingkan Linear Search pada data yang telah terurut. Oleh karena itu, metode ini cocok diterapkan pada pencarian data berdasarkan Nomor Induk Santri (NIS) yang bersifat numerik dan terstruktur. Meskipun DBMS modern telah menyediakan mekanisme optimasi pencarian melalui indeks dan *query optimizer*, algoritma pencarian pada tingkat aplikasi masih digunakan pada berbagai skenario seperti pemrosesan data dalam memori, manipulasi koleksi data lokal, serta pemrosesan hasil query sebelum ditampilkan kepada pengguna (Lutfina et al., 2022; Singh et al., 2025). Dengan demikian, algoritma pencarian pada tingkat aplikasi masih memiliki relevansi pada kondisi tertentu dan tidak sepenuhnya dapat digantikan oleh mekanisme pencarian pada DBMS. Di sisi lain, penelitian mengenai optimasi data dasar menunjukkan bahwa kinerja SQL Query sangat dipengaruhi oleh indeks struktur, strategi eksekusi query, dan karakteristik DBMS yang digunakan (Saidu et al., 2024; Sun et al., 2024). Oleh karena itu, evaluasi ketiga pendekatan tersebut tetap relevan untuk memberikan gambaran empiris mengenai perbedaan mekanisme pencarian pada tingkat aplikasi dan tingkat basis data dalam sistem informasi berbasis web.

Pada sistem basis data modern, proses pencarian data umumnya memanfaatkan struktur indeks seperti B-Tree dan Hash Index yang memungkinkan akses data dilakukan secara lebih efisien dibandingkan pencarian sekuensial. Selain itu, DBMS juga dilengkapi mekanisme query optimizer yang mampu menentukan strategi eksekusi terbaik berdasarkan karakteristik data dan indeks yang tersedia. Berbagai penelitian menunjukkan bahwa kombinasi teknik indexing dan query optimization mampu meningkatkan performa pencarian data secara signifikan pada sistem berbasis web (Saidu et al., 2024; Sun et al., 2024). Oleh karena itu, SQL Query tidak dapat disamakan secara langsung dengan Binary Search maupun Linear Search karena proses pencariannya melibatkan optimasi internal DBMS yang lebih kompleks.

Penelitian terdahulu mengenai optimasi pencarian data menunjukkan kecenderungan yang berbeda. Lutfina et al. (2022) berfokus pada evaluasi implementasi Linear Search menggunakan pendekatan rekursif dan iteratif. Sun et al. (2024) menitikberatkan pada optimasi pencarian melalui indexing dan query optimization pada DBMS modern. Temuan-temuan tersebut menunjukkan bahwa penelitian terdahulu cenderung mengevaluasi algoritma pencarian pada tingkat aplikasi dan optimasi pencarian pada tingkat basis data secara terpisah, sehingga belum memberikan gambaran komprehensif mengenai karakteristik kinerja kedua pendekatan tersebut dalam satu lingkungan sistem informasi berbasis web.

Perkembangan penelitian terbaru menunjukkan bahwa fokus optimasi pencarian data telah bergeser dari perbaikan algoritma pencarian konvensional menuju optimalisasi struktur indeks, query optimizer, dan strategi eksekusi pada DBMS modern. Penelitian mengenai indexing database menunjukkan bahwa struktur indeks seperti B-Tree dan Hash Index mampu mengurangi biaya akses data secara signifikan dibandingkan pencarian sekuensial pada tingkat aplikasi (Saidu et al., 2024). Selain itu, studi mengenai query optimization menunjukkan bahwa performa pencarian tidak hanya dipengaruhi oleh algoritma pencarian, tetapi juga oleh mekanisme internal DBMS dalam menentukan execution plan yang optimal (Sun et al., 2024). Kondisi ini menunjukkan bahwa evaluasi pencarian data pada sistem informasi modern perlu mempertimbangkan interaksi antara algoritma aplikasi dan mekanisme optimasi basis data. Selain itu, belum ditemukan penelitian yang mengevaluasi perbedaan karakteristik pencarian pada tingkat aplikasi dan tingkat basis data dalam konteks sistem informasi pengelolaan data santri berbasis web.

Berdasarkan kondisi tersebut, terdapat kesenjangan penelitian (research gap), yaitu belum adanya kajian empiris yang membandingkan algoritma pencarian pada tingkat aplikasi (*Linear Search* dan *Binary Search*) dengan mekanisme pencarian pada tingkat basis data (SQL Query) dalam satu lingkungan sistem informasi berbasis web yang sama. Selain itu, belum banyak penelitian yang menjelaskan kapan algoritma pencarian manual masih relevan digunakan meskipun DBMS modern telah menyediakan mekanisme pengindeksan dan optimasi kueri. Oleh karena itu, penelitian ini berupaya mengevaluasi kinerja ketiga pendekatan tersebut pada kasus pengelolaan data santri serta memberikan bukti empiris mengenai karakteristik kinerja masing-masing metode dalam sistem informasi berbasis web.

Penelitian ini menawarkan solusi dalam bentuk pengembangan sebuah sistem informasi untuk mengelola data santri yang berbasis web dengan menerapkan tiga metode pencarian yang berbeda, yaitu *Linear Search*, *Binary Search*, dan *SQL Query*. Kebaruan penelitian ini terletak pada evaluasi empiris tiga pendekatan pencarian dalam satu lingkungan sistem informasi berbasis web yang sama pada kasus pengelolaan data santri, sehingga memungkinkan identifikasi kondisi ketika algoritma pencarian pada tingkat aplikasi masih relevan dibandingkan mekanisme pencarian berbasis DBMS. Sistem ini dibangun menggunakan framework Laravel karena kemampuan framework ini

dalam mendukung pengembangan aplikasi web secara terstruktur dengan mengadopsi konsep Model View Controller (MVC) dan memiliki integrasi data yang efisien (Al-Hawari, 2022). Selain itu, perancangan sistem menggunakan Unified Modeling Language (UML) untuk mempermudah proses analisis kebutuhan dan desain sistem secara sistematis (Fitrani & Puspitaningrum, 2023).

Penelitian ini bertujuan untuk merancang dan mengembangkan sistem informasi pengelolaan data santri berbasis web serta menganalisis karakteristik kinerja Linear Search, Binary Search, dan SQL Query dalam proses pencarian data. Kontribusi penelitian ini terletak pada penyediaan bukti empiris mengenai perbedaan pendekatan pencarian pada tingkat aplikasi dan tingkat basis data dalam konteks sistem informasi pendidikan. Hasil penelitian diharapkan dapat menjadi referensi bagi pengembang sistem informasi dalam menentukan strategi pencarian data yang sesuai dengan kebutuhan sistem serta memberikan masukan bagi pengembangan penelitian lanjutan terkait optimasi akses data pada aplikasi berbasis web. Penelitian ini juga diharapkan dapat memberikan dasar empiris bagi pengembang sistem dalam menentukan kapan algoritma pencarian pada tingkat aplikasi masih relevan digunakan dan kapan mekanisme pencarian berbasis DBMS menjadi pilihan yang lebih optimal.

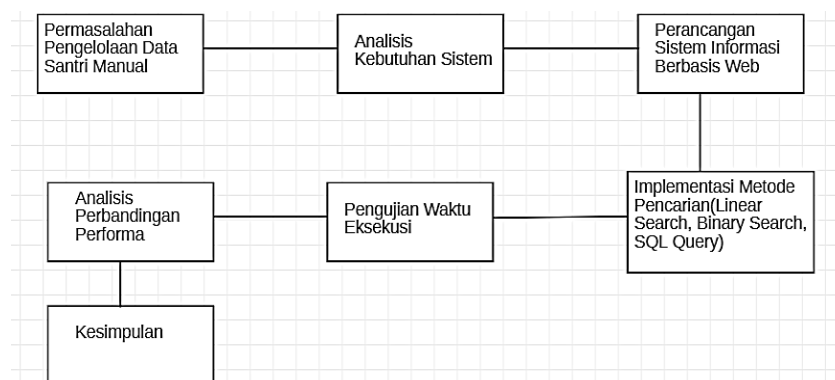
2. METODOLOGI PENELITIAN

2.1 Kerangka Dasar Penelitian

Penelitian ini merupakan penelitian terapan (*applied research*) yang bertujuan untuk merancang dan mengembangkan sistem informasi pengelolaan data santri berbasis web serta mengevaluasi kinerja metode pencarian data menggunakan *Linear Search*, *Binary Search*, dan *SQL Query*. Penelitian ini dilaksanakan di Pondok Pesantren Roudlotul Quran Darul Falah III Jombang yang masih melakukan pengelolaan data santri secara konvensional sehingga proses pencarian dan pengelolaan informasi administrasi belum berjalan secara optimal. Pemanfaatan sistem informasi berbasis web pada lingkungan pendidikan diketahui mampu meningkatkan efisiensi pengelolaan data dan kualitas layanan administrasi (Lontaan & Sinadia, 2024; Pasaribu & Argadikusuma, 2024). Dataset yang digunakan dalam penelitian ini terdiri atas 1.000 data santri yang meliputi identitas santri, Nomor Induk Santri (NIS), asal daerah, dan nama orang tua atau wali. Dataset tersebut digunakan sebagai data pengujian untuk menganalisis karakteristik kinerja masing-masing metode pencarian dan tidak dimaksudkan untuk merepresentasikan lingkungan data berskala besar.

Hipotesis penelitian ini adalah terdapat perbedaan karakteristik kinerja antara metode *Linear Search*, *Binary Search*, dan *SQL Query* dalam proses pencarian data santri. Variabel independen pada penelitian ini adalah metode pencarian yang terdiri atas Linear Search, Binary Search, dan SQL Query, sedangkan variabel dependennya adalah waktu eksekusi pencarian data. Analisis data dilakukan menggunakan pendekatan deskriptif komparatif dengan membandingkan rata-rata waktu eksekusi masing-masing metode pada dataset yang sama untuk mengetahui karakteristik dan efisiensi setiap metode pencarian (Lutfina et al., 2022).

Kerangka pemikiran penelitian dimulai dari identifikasi masalah pada proses pengelolaan data santri, kemudian dilanjutkan dengan analisis kebutuhan sistem, perancangan dan implementasi sistem informasi berbasis web, penerapan metode pencarian, pengujian kinerja, dan analisis hasil pengujian. Tahapan penelitian tersebut disajikan pada Gambar 1 untuk memberikan gambaran alur penelitian secara sistematis.



Gambar 1. Kerangka Pemikiran Penelitian

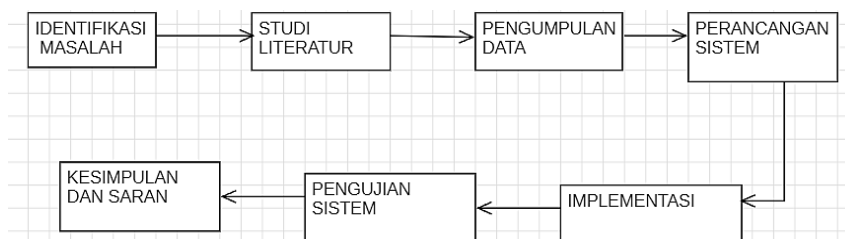
2.2 Tahapan Penelitian

Model pengembangan sistem yang digunakan dalam penelitian ini adalah model *Waterfall*. Pemilihan model Waterfall didasarkan pada karakteristik penelitian yang memiliki kebutuhan sistem yang telah teridentifikasi sejak awal dan ruang lingkup pengembangan yang relatif stabil, sehingga proses pengembangan dapat dilakukan secara bertahap dan terstruktur (Siregar & Arpan, 2025). Meskipun metode Agile lebih banyak digunakan pada pengembangan perangkat lunak modern, model Waterfall masih relevan digunakan pada penelitian akademik yang berfokus pada perancangan, implementasi, dan evaluasi sistem dengan kebutuhan sistem yang relatif stabil dan tidak mengalami perubahan signifikan selama proses pengembangan.

Tahapan penelitian dimulai dari analisis kebutuhan sistem. Pada tahap ini dilakukan observasi dan wawancara untuk mengidentifikasi permasalahan pada proses pengelolaan data santri serta menentukan kebutuhan fungsional dan nonfungsional sistem. Tahap berikutnya adalah studi literatur yang dilakukan dengan mengumpulkan referensi mengenai sistem informasi berbasis web, algoritma pencarian data, optimasi query, dan teknik pengindeksan basis data sebagai landasan teoritis penelitian (Saidu et al., 2024; Sun et al., 2024). Tahap selanjutnya adalah perancangan sistem yang meliputi perancangan antarmuka pengguna, desain basis data, dan pemodelan sistem menggunakan Unified Modeling Language (UML). Penggunaan UML membantu proses analisis dan perancangan sistem secara sistematis sehingga mempermudah tahap implementasi (Fitriani & Puspitaningrum, 2023).

Tahap implementasi dilakukan dengan membangun sistem menggunakan framework Laravel, bahasa pemrograman PHP, dan basis data MySQL. Tahap pengujian dilakukan dengan menerapkan tiga metode pencarian, yaitu Linear Search, Binary Search, dan SQL Query untuk mengukur waktu eksekusi pencarian data. Untuk menjaga konsistensi hasil pengujian, seluruh metode diuji pada lingkungan perangkat keras dan perangkat lunak yang sama. Pengujian dilakukan pada kondisi cache yang dikendalikan sehingga setiap metode dieksekusi pada lingkungan yang identik. Setiap skenario pengujian diulang sebanyak lima kali untuk memperoleh nilai waktu eksekusi yang lebih representatif, sehingga diperoleh nilai waktu eksekusi yang lebih representatif. Pengendalian terhadap konfigurasi pengujian merupakan aspek penting dalam penelitian eksperimental pada sistem basis data karena kinerja pencarian dipengaruhi oleh konfigurasi sistem, strategi eksekusi query, dan mekanisme optimasi internal DBMS (Saidu et al., 2024).

Tahap terakhir adalah analisis hasil pengujian untuk mengevaluasi karakteristik kinerja masing-masing metode pencarian dan menentukan pendekatan yang paling sesuai untuk diterapkan pada sistem informasi pengelolaan data santri berbasis web. Tahapan penelitian secara keseluruhan ditunjukkan pada Gambar 2, yang terdiri atas tahap analisis kebutuhan, studi literatur, perancangan sistem, implementasi, pengujian, dan analisis hasil yang dilakukan secara berurutan sesuai model Waterfall.



Gambar 2. Flowchart Tahapan Penelitian

2.3 Perancangan Sistem

Perancangan sistem dilakukan menggunakan pendekatan *Unified Modeling Language (UML)* untuk memodelkan kebutuhan fungsional dan hubungan antar komponen sistem secara sistematis. Penggunaan UML dapat membantu proses analisis, perancangan, dan dokumentasi sistem sehingga mempermudah tahap implementasi perangkat lunak (Fitriani & Puspitaningrum, 2023). Pada penelitian ini, model UML yang digunakan adalah Use Case Diagram untuk menggambarkan interaksi antara pengguna dan sistem.

Sistem informasi pengelolaan data santri memiliki dua aktor utama, yaitu admin dan pengguna. Admin memiliki hak akses untuk mengelola data santri, melakukan pencarian data, menambah, mengubah, menghapus data, serta mencetak laporan. Sementara itu, pengguna hanya diberikan hak untuk melakukan pencarian data santri. Use Case Diagram sistem ditunjukkan pada Gambar 3.



Gambar 3. Sistem Use Case Diagram

Gambar 3 menunjukkan bahwa admin memiliki akses penuh terhadap seluruh fungsi sistem, sedangkan pengguna hanya dapat mengakses fitur pencarian data. Pembagian hak akses tersebut diterapkan untuk menjaga keamanan dan integritas data pada sistem.

Selain pemodelan fungsional, penelitian ini juga merancang basis data untuk mendukung implementasi dan pengujian metode pencarian. Basis data menggunakan MySQL karena mendukung mekanisme indexing dan query optimization yang dapat meningkatkan efisiensi akses data pada aplikasi berbasis web (Saidu et al., 2024; Sun et al., 2024). Basis data terdiri atas beberapa tabel, yaitu tabel pengguna, tabel santri, dan tabel log pencarian. Namun, tabel yang digunakan secara langsung dalam proses pengujian adalah tabel santri dan tabel log pencarian.

Tabel 1. Struktur Tabel Santri

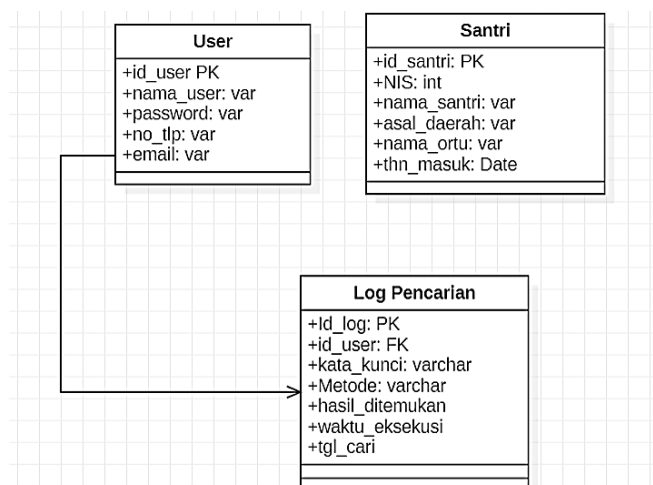
Nama Field	Type	Keterangan
id_santri	Integer	Primary Key
nis	Integer	Unique
nama_santri	Varchar	Nama Santri
asal_daerah	Varchar	Asal Daerah
nama_ortu	Varchar	Nama Orang Tua

Tabel santri berfungsi sebagai sumber data utama saat melakukan pencarian dengan menggunakan metode *Linear Search*, *Binary Search*, dan *SQL Query*. Struktur tabel santri ditunjukkan pada Tabel 1. Data pada field *nama_santri*, *asal_daerah*, dan *nama_ortu* digunakan pada implementasi *Linear Search* karena proses pencarian dilakukan terhadap data yang tidak memerlukan pengurutan (*sorting*) terlebih dahulu. Sementara itu, field *nis* digunakan pada implementasi *Binary Search* karena data tersebut bersifat numerik dan disusun secara terurut. Selain tabel santri, sistem juga menggunakan tabel log pencarian untuk merekam aktivitas pencarian dan menyimpan hasil pengujian waktu eksekusi. Struktur tabel untuk log pencarian terlihat di Tabel 2.

Tabel 2. Struktur Tabel Log Pencarian

Nama Field	Type	Keterangan
id_log	Integer	Primary Key
id_user	Integer	Foreign Key
kata_kunci	Varchar(100)	Input Pencarian
metode	ENUM('linear','biner','query_sql')	Metode Pencarian
hasil_ditemukan	Integer	Jumlah Hasil
waktu_eksekusi	Float	Waktu Eksekusi
tgl_cari	Datetime	Waktu Pencarian

Tabel log pencarian digunakan sebagai sumber data utama dalam proses analisis perbandingan kinerja metode pencarian karena menyimpan informasi mengenai metode yang digunakan, kata kunci pencarian, jumlah data yang ditemukan, dan waktu eksekusi setiap proses pencarian. Hubungan antarentitas pada basis data digambarkan menggunakan Entity Relationship Diagram (ERD) sebagaimana ditunjukkan pada Gambar 4.



Gambar 4. Entity Relationship Diagram (ERD)

Gambar 4 menunjukkan hubungan antara tabel pengguna, tabel santri, dan tabel log pencarian yang digunakan dalam proses pengelolaan data serta pencatatan hasil pengujian. Struktur basis data tersebut dirancang untuk mendukung implementasi dan evaluasi kinerja metode *Linear Search*, *Binary Search*, dan *SQL Query* dalam sistem informasi pengelolaan data santri berbasis web.



2.4 Implementasi Algoritma Pencarian

Implementasi metode pencarian untuk menganalisis karakteristik kinerja tiga pendekatan pencarian data, yaitu *Linear Search*, *Binary Search*, dan *SQL Query* pada sistem informasi pengelolaan data santri berbasis web. Penelitian ini tidak bertujuan membandingkan kompleksitas algoritma secara teoritis, tetapi mengevaluasi karakteristik kinerja pendekatan pencarian yang umum digunakan pada sistem informasi berbasis web. Ketiga metode diimplementasikan menggunakan dataset yang sama sehingga perbedaan waktu eksekusi yang diperoleh dapat dibandingkan secara objektif.

Pada implementasinya, metode *Linear Search* dan *Binary Search* dilakukan pada tingkat aplikasi (*application level*) menggunakan data yang telah diambil dari basis data ke dalam memori aplikasi. Sementara itu, metode *SQL Query* dilakukan secara langsung pada basis data dengan memanfaatkan mekanisme optimasi internal DBMS, seperti *indexing* dan *query optimizer*. Seluruh pengujian dilakukan pada lingkungan perangkat keras dan perangkat lunak yang sama untuk menjaga konsistensi hasil pengukuran.

2.4.1 Implementasi Linear Search

Metode *Linear Search* digunakan untuk melakukan pencarian data yang tidak teratur atau tidak memiliki urutan tertentu. Pada penelitian ini, metode *Linear Search* diterapkan pada field *nama_santri*, *asal_daerah*, dan *nama_ortu*. Proses pencarian dilakukan dengan cara membandingkan data satu per satu, dimulai dari data yang pertama hingga data yang terakhir, sampai data yang dicari ditemukan. Tahapan proses *Linear Search* adalah sebagai berikut:

- Sistem menerima input kata kunci dari pengguna.
- Sistem membaca data santri dari database.
- Sistem membandingkan kata kunci dengan setiap data secara berurutan.
- Jika data ditemukan, sistem menampilkan hasil pencarian.
- Jika data tidak ditemukan, sistem menampilkan pesan data tidak ditemukan.

Kompleksitas waktu metode *Linear Search* adalah:

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

Kompleksitas tersebut menunjukkan bahwa waktu pencarian meningkat secara linier terhadap jumlah data yang diproses. Oleh karena itu, metode *Linear Search* cenderung mengalami penurunan performa pada dataset yang lebih besar meskipun implementasinya sederhana dan mudah diterapkan (Lutfina et al., 2022).

2.4.2 Implementasi Binary Search

Metode *Binary Search* digunakan untuk melakukan pencarian pada data yang sudah diurutkan. Dalam penelitian ini, metode *Binary Search* diterapkan pada field *NIS* karena data tersebut bersifat numerik dan telah disusun secara terurut sehingga memenuhi prasyarat penggunaan algoritma *Binary Search*. Proses pencarian dilakukan dengan membagi data menjadi dua bagian berulang kali sampai data tersebut ditemukan. Tahapan proses *Binary Search* adalah sebagai berikut:

- Sistem menerima input *NIS* dari pengguna.
- Sistem mengambil data yang telah diurutkan berdasarkan *NIS*.
- Sistem menentukan data tengah.
- Sistem membandingkan data tengah dengan data yang dicari.
- Jika data cocok, hasil ditampilkan.
- Jika data lebih kecil atau besar, pencarian dilanjutkan pada bagian data yang sesuai.

Kompleksitas waktu *Binary Search* adalah:

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

Metode ini mampu mengurangi jumlah operasi pencarian secara signifikan dibandingkan *Linear Search* karena ruang pencarian diperkecil secara bertahap pada setiap iterasi. Oleh karena itu, *Binary Search* lebih efisien pada data yang telah terurut (Singh et al., 2025).

2.4.3 Implementasi SQL Query

Metode *SQL Query* digunakan untuk melakukan pencarian secara langsung pada basis data dengan memanfaatkan kemampuan optimasi yang tersedia pada DBMS. Berbeda dengan *Linear Search* dan *Binary Search* yang bekerja pada tingkat aplikasi, *SQL Query* memanfaatkan mekanisme internal basis data, seperti *indexing* dan *query optimizer*, untuk menentukan strategi pencarian yang paling efisien. Tahapan proses *SQL Query* adalah sebagai berikut:

- Sistem menerima input kata kunci dari pengguna.
- Sistem mengirimkan query ke database.
- Database memproses query menggunakan indeks.
- Hasil pencarian dikembalikan ke sistem.
- Sistem menampilkan hasil pencarian kepada pengguna.



Pada penelitian ini, field *nis* diberikan indeks unik (*unique index*) sehingga proses pencarian berdasarkan NIS dapat memanfaatkan struktur indeks yang disediakan oleh DBMS. Sementara itu, pencarian pada field teks dilakukan menggunakan query sesuai kebutuhan implementasi sistem. Penggunaan struktur indeks memungkinkan DBMS melakukan pencarian secara lebih efisien dibandingkan pencarian sekuensial pada tingkat aplikasi.

Berbagai penelitian menunjukkan bahwa penerapan *indexing* dan *query optimization* dapat meningkatkan performa pencarian data secara signifikan pada sistem berbasis web (Saidu et al., 2024; Sun et al., 2024).. Oleh karena itu, *SQL Query* digunakan sebagai pembanding untuk mengevaluasi perbedaan karakteristik pencarian pada tingkat aplikasi dan tingkat basis data dalam sistem informasi pengelolaan data santri.

Untuk menjaga validitas perbandingan, seluruh metode diuji pada lingkungan perangkat keras dan perangkat lunak yang sama serta pada kondisi *database caching* dan *opcode caching* yang dikendalikan sebelum setiap pengujian dilakukan. Pengendalian tersebut dilakukan untuk meminimalkan pengaruh mekanisme cache terhadap hasil pengukuran waktu eksekusi sehingga perbandingan kinerja antar metode dapat dilakukan secara lebih objektif.

2.5 Teknik Pengujian Sistem

Teknik pengujian sistem pada penelitian ini dilakukan untuk mengukur dan membandingkan kinerja metode pencarian data yang diterapkan pada sistem informasi pengelolaan data santri yang berbasis web. Pengujian dilakukan dengan menggunakan pendekatan *execution time testing*, yaitu pengukuran waktu yang dibutuhkan oleh setiap metode dalam menemukan data yang dicari (Lutfina et al., 2022). Ketiga metode pencarian, yaitu *Linear Search*, *Binary Search*, dan *SQL Query*, diuji menggunakan dataset yang sama sehingga hasil pengukuran dapat dibandingkan secara objektif. Dataset yang digunakan berasal dari data santri di Pondok Pesantren Roudlotul Quran Darul Falah III Jombang dengan jumlah total 1.000 data. Jumlah data tersebut digunakan sebagai representasi dataset operasional pada objek penelitian dan tidak dimaksudkan untuk merepresentasikan skala big data atau sistem berskala sangat besar. Untuk mengamati pengaruh pertambahan jumlah data terhadap waktu pencarian, pengujian dilakukan pada beberapa skenario ukuran dataset, yaitu 100, 300, 500, 700, dan 1.000 data. Setiap skenario pengujian dilakukan sebanyak sepuluh kali percobaan untuk memperoleh nilai rata-rata waktu eksekusi yang lebih representatif serta mengurangi variasi pengukuran yang dapat disebabkan oleh aktivitas sistem di luar proses pengujian. Hasil pengujian kemudian dianalisis menggunakan metode deskriptif komparatif dengan membandingkan rata-rata waktu eksekusi masing-masing metode pencarian pada setiap skenario jumlah data.

Untuk menjaga validitas hasil pengujian, seluruh pengujian dilakukan pada lingkungan perangkat keras dan perangkat lunak yang sama. Selain itu, pengujian dilakukan dengan memperhatikan kondisi eksekusi sistem, termasuk mekanisme database caching dan opcode caching, karena kedua mekanisme tersebut dapat mempengaruhi waktu respons aplikasi dan basis data. Setiap metode diuji pada kondisi konfigurasi yang sama sehingga perbedaan waktu eksekusi yang diperoleh lebih mencerminkan perbedaan karakteristik metode pencarian yang diuji daripada pengaruh faktor eksternal. Pengendalian terhadap lingkungan pengujian merupakan aspek penting dalam evaluasi kinerja sistem basis data karena performa pencarian dipengaruhi oleh konfigurasi sistem, strategi eksekusi query, dan mekanisme optimasi internal DBMS (Saidu et al., 2024; Sun et al., 2024).

Parameter utama yang dipakai dalam pengujian adalah waktu yang dibutuhkan untuk melakukan pencarian, yang diukur dalam detik. Pengukuran waktu eksekusi dilakukan menggunakan fungsi `microtime(true)` pada PHP yang mampu memberikan pengukuran waktu dengan presisi tinggi (*high-resolution timing*). Waktu eksekusi dihitung dengan melihat selisih waktu sebelum dan sesudah proses pencarian:

$$\text{Execution Time} = \text{End Time} - \text{Start Time} \quad (3)$$

Nilai *Start Time* merupakan waktu ketika proses pencarian dimulai, sedangkan *End Time* merupakan waktu ketika hasil pencarian berhasil diperoleh dan ditampilkan oleh sistem. Pengujian dilakukan dengan menggunakan alat yang sama agar hasil pengujian tetap konsisten dan valid. Spesifikasi perangkat yang digunakan untuk pengujian dapat dilihat di Tabel 3.

Tabel 3. Spesifikasi Perangkat Pengujian

Komponen	Spesifikasi
Device	Laptop HP 14-em0xxx
Prosesor	AMD Ryzen 5 Radeon Graphics (~2.0GHz)
Grafik	Grafis Radeon
Storage	477 GB
RAM	8 GB
Sistem Operasi	Windows 11 64-bit
Web Server	Apache 2.4 (XAMPP)
PHP Version	8.2.12
MySQL Version	MariaDB 10.4.32

Seluruh pengujian dilakukan pada lingkungan *localhost* menggunakan web server Apache dan basis data MariaDB. Konfigurasi *database caching* dan *opcode caching* dipertahankan dalam kondisi yang sama pada setiap

skenario pengujian sehingga pengaruh mekanisme cache terhadap hasil pengukuran dianggap konstan untuk seluruh metode yang dibandingkan.

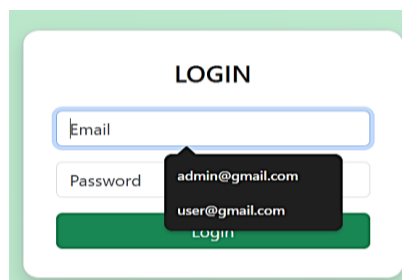
3. HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan sistem informasi pengelolaan data santri yang berbasis web yang digunakan sebagai media implementasi dan evaluasi kinerja metode pencarian data, yaitu *Linear Search*, *Binary Search*, dan *SQL Query*. Sistem dikembangkan menggunakan framework Laravel dengan bahasa pemrograman PHP dan basis data MySQL (MariaDB) untuk menyimpan data. Penggunaan framework Laravel dipilih karena mendukung arsitektur Model View Controller (MVC) yang mempermudah pengembangan aplikasi web secara terstruktur dan pemeliharaan sistem (Al-Hawari, 2022). Pengembangan sistem bertujuan untuk membantu proses administrasi data santri sekaligus menyediakan lingkungan pengujian yang seragam untuk membandingkan karakteristik kinerja ketiga metode pencarian.

Fitur utama yang tersedia pada sistem meliputi pengelolaan data santri, pencarian data menggunakan tiga metode pencarian yang berbeda, pencatatan log pencarian, serta pembuatan laporan data santri. Seluruh fitur dirancang berdasarkan hasil analisis kebutuhan pengguna sehingga sistem tidak hanya mendukung pengelolaan data santri secara efektif dan efisien, tetapi juga menyediakan lingkungan pengujian yang konsisten untuk membandingkan kinerja metode pencarian.

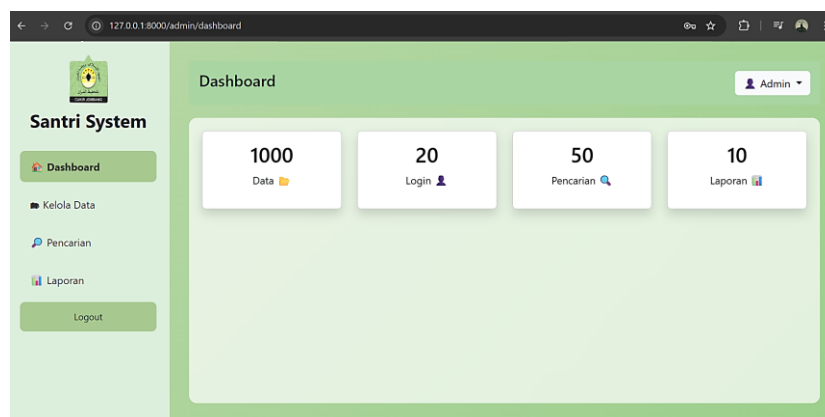
3.1 Implementasi Sistem

Sistem ini diimplementasikan dalam bentuk aplikasi berbasis web, yang bisa diakses melalui browser. Implementasi sistem difokuskan pada penyediaan fitur pengelolaan data dan fasilitas pengujian metode pencarian sehingga proses evaluasi dapat dilakukan secara terukur dan objektif. Sebelum mengakses sistem, pengguna diwajibkan melakukan autentikasi melalui halaman login. Halaman ini berfungsi untuk memverifikasi identitas pengguna dan mengatur hak akses berdasarkan peran yang dimiliki.



Gambar 5. Halaman Login Sistem

Gambar 5 menunjukkan antarmuka login yang digunakan untuk memasukkan nama pengguna dan kata sandi sebelum pengguna dapat mengakses sistem. Mekanisme autentikasi diterapkan untuk menjaga keamanan data serta membatasi hak akses pengguna sesuai dengan peran yang dimiliki. Setelah proses autentikasi berhasil, pengguna akan diarahkan menuju halaman dashboard yang menampilkan menu utama sistem.

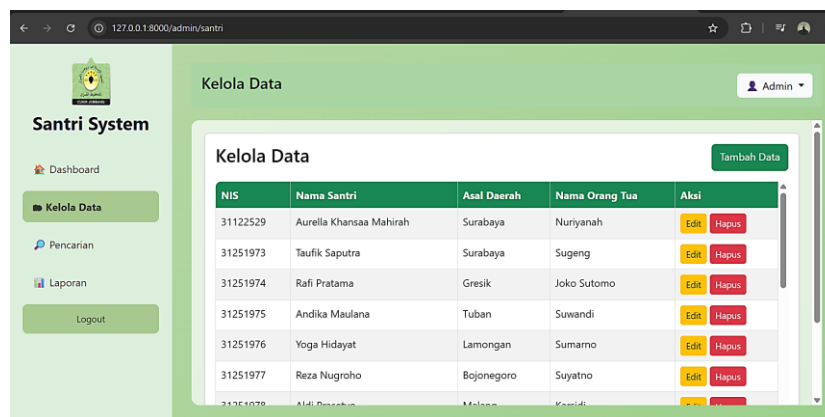


Gambar 6. Dashboard Sistem

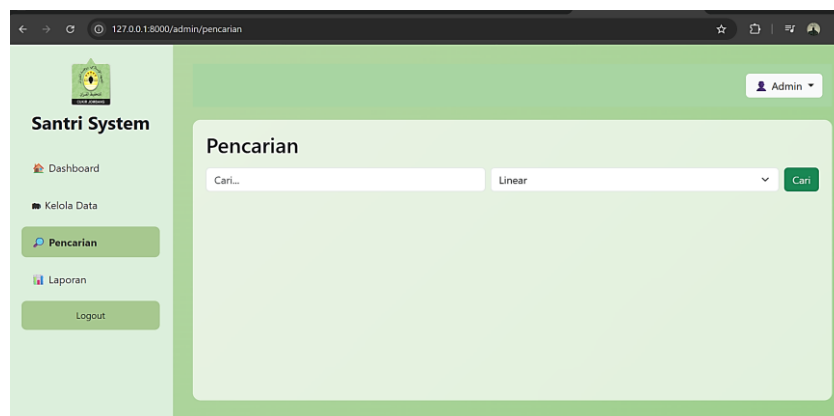
Gambar 6 menunjukkan halaman dashboard yang berisi menu pengelolaan data santri, pencarian data, log pencarian, dan pembuatan laporan. Dashboard berfungsi sebagai pusat navigasi untuk mengakses seluruh fitur yang tersedia pada sistem. Proses pengelolaan data santri dilakukan melalui halaman data santri yang menyediakan fungsi tambah, ubah, hapus, dan pencarian data.

Gambar 7 menunjukkan halaman pengelolaan data santri yang digunakan admin untuk melakukan pengelolaan data administrasi santri. Data yang dikelola melalui halaman ini disimpan pada tabel santri dan digunakan sebagai

dataset utama dalam implementasi serta pengujian metode pencarian. Fitur pencarian data merupakan komponen utama dalam penelitian ini karena digunakan untuk mengimplementasikan dan mengevaluasi kinerja metode *Linear Search*, *Binary Search*, dan *SQL Query*.

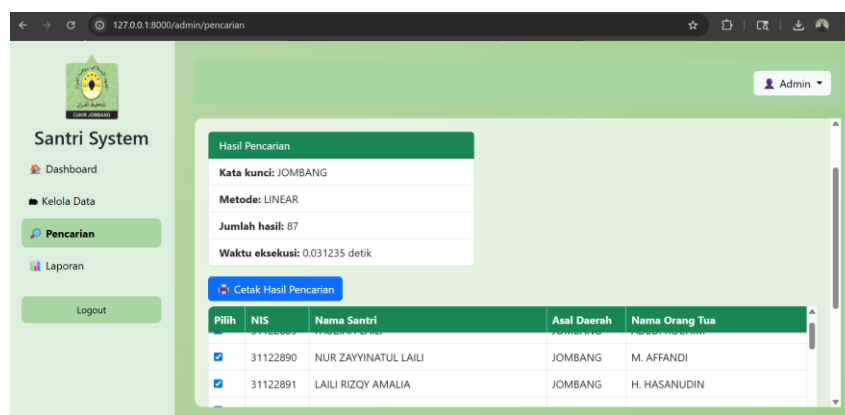


Gambar 7. Halaman Data Santri



Gambar 8. Halaman Pencarian Data Santri

Gambar 8 menunjukkan antarmuka pencarian data yang memungkinkan pengguna memasukkan kata kunci pencarian dan memilih metode pencarian yang akan digunakan. Fitur ini dirancang agar seluruh metode pencarian dijalankan pada antarmuka dan lingkungan eksekusi yang sama sehingga hasil pengukuran dapat dibandingkan secara objektif. Setiap proses pencarian yang dilakukan oleh pengguna dicatat secara otomatis melalui fitur log pencarian



Gambar 9. Hasil Pencarian Data Santri

Gambar 9 menunjukkan hasil pencarian yang menampilkan data santri yang ditemukan beserta informasi mengenai metode pencarian yang digunakan dan waktu eksekusi pencarian. Informasi tersebut digunakan sebagai data pengujian untuk menganalisis dan membandingkan kinerja masing-masing metode pencarian berdasarkan waktu eksekusi yang dihasilkan. Berdasarkan hasil implementasi, seluruh fitur sistem dapat berjalan sesuai dengan kebutuhan fungsional yang telah ditentukan. Sistem juga mampu menyediakan lingkungan pengujian yang seragam sehingga proses evaluasi kinerja *Linear Search*, *Binary Search*, dan *SQL Query* dapat dilakukan secara terukur dan objektif. Hasil pengujian dan analisis perbandingan kinerja ketiga metode pencarian tersebut dibahas pada subbab berikutnya.

3.2 Hasil Pengujian Metode Pencarian

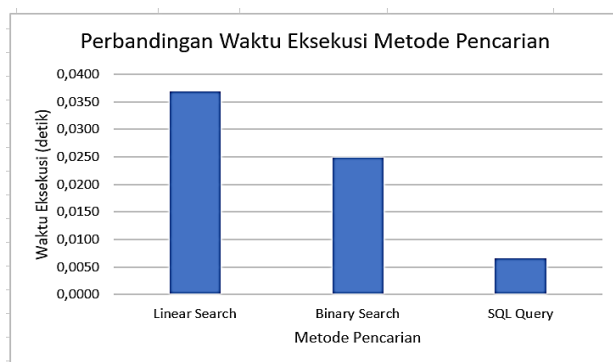
Pengujian dilakukan menggunakan dataset operasional sebanyak 1000 data santri yang tersimpan pada basis data MySQL/MariaDB. Jumlah data tersebut disesuaikan dengan jumlah data aktual yang tersedia pada objek penelitian dan digunakan untuk mengevaluasi karakteristik relatif ketiga metode pencarian dalam lingkungan sistem yang sama. Setiap skenario pengujian dilakukan sebanyak sepuluh kali percobaan dan nilai yang digunakan dalam analisis adalah rata-rata waktu eksekusi. Pengukuran dilakukan menggunakan fungsi `microtime(true)` pada PHP dalam lingkungan perangkat keras dan perangkat lunak yang sama.

3.2.1 Pengujian Berdasarkan NIS

Pengujian dilakukan menggunakan data numerik berupa Nomor Induk Santri (NIS) dengan kata kunci 31122550. dengan kata kunci pencarian 31122550. Hasil pengujian ditunjukkan pada Tabel 4.

Tabel 4. Hasil Pengujian Berdasarkan NIS

Metode	Rata-rata Waktu Eksekusi (detik)	Hasil Ditemukan
Linear Search	0.0369	1
Binary Search	0.0248	1
SQL Query	0.0065	1



Gambar 10. Perbandingan Waktu Eksekusi Metode Pencarian Berdasarkan NIS

Berdasarkan Tabel 4 dan Gambar 10, *SQL Query* menghasilkan waktu eksekusi paling rendah, yaitu 0,0065 detik, diikuti oleh *Binary Search* sebesar 0,0248 detik dan *Linear Search* sebesar 0,0369 detik. Hasil ini menunjukkan bahwa pemanfaatan *indexing* dan *query optimizer* pada DBMS mampu meningkatkan efisiensi pencarian data dibandingkan algoritma pencarian pada tingkat aplikasi. Selain itu, *Binary Search* menunjukkan kinerja yang lebih baik daripada *Linear Search* karena memiliki kompleksitas waktu $O(\log n)$, sedangkan *Linear Search* memiliki kompleksitas waktu $O(n)$. Temuan ini sejalan dengan penelitian Saidu et al. (2024) dan Sun et al. (2024) yang menyatakan bahwa optimasi query dan struktur indeks dapat mempercepat akses data secara signifikan.

3.2.2 Pengujian Berdasarkan Data Teks

Pengujian berikutnya dilakukan menggunakan data teks berupa nama lengkap, Sebagian nama, dan asal daerah. Hasil pengujian ditunjukkan pada Tabel 5.

Tabel 5. Hasil Pengujian Berdasarkan Data Teks

Jenis Pencarian	Metode	Rata-rata Waktu Eksekusi (detik)	Hasil Ditemukan
Nama Lengkap	Linear Search	0.0396	1
Nama Lengkap	SQL Query	0.0069	1
Nama Sebagian	Linear Search	0.0369	2
Nama Sebagian	SQL Query	0.0069	2
Asal Daerah	Linear Search	0.0368	87
Asal Daerah	SQL Query	0.0081	87

Berdasarkan Tabel 5, *SQL Query* secara konsisten menghasilkan waktu pencarian yang lebih rendah dibandingkan *Linear Search* pada seluruh skenario pencarian data teks. Hal ini menunjukkan bahwa proses pencarian yang memanfaatkan optimasi internal DBMS lebih efisien dibandingkan pencarian sekuensial pada tingkat aplikasi. Selain itu, waktu eksekusi *Linear Search* cenderung meningkat karena seluruh data harus diperiksa satu per satu. Kondisi ini sesuai dengan karakteristik kompleksitas waktu $O(n)$ yang dimiliki oleh algoritma *Linear Search*.

3.2.3 Pengujian Data Tidak Ditemukan

Pengujian dilakukan menggunakan kata kunci yang tidak tersedia pada basis data. Hasil pengujian ditunjukkan pada Tabel 6.

**Tabel 6.** Pengujian Data Tidak Ditemukan

Metode	Rata-rata Waktu Eksekusi (detik)	Hasil Ditemukan
Linear Search	0.0353	Tidak ditemukan
Binary Search	0.0223	Tidak ditemukan
SQL Query	0.0062	Tidak ditemukan

Berdasarkan Tabel 6, seluruh metode mampu menangani kondisi ketika data tidak ditemukan. Namun, *SQL Query* tetap menunjukkan waktu eksekusi paling rendah, sedangkan *Linear Search* membutuhkan waktu paling lama karena harus memeriksa seluruh data sebelum menyimpulkan bahwa data yang dicari tidak tersedia.

3.2.4 Pengujian Berulang (*Stress Test*)

Pengujian dilakukan dengan melakukan pencarian berulang sebanyak 10 kali menggunakan kata kunci yang sama. Hasil pengujian ditunjukkan pada Tabel 7.

Tabel 7. Hasil Pengujian Berulang

Metode	Rata-rata Waktu Eksekusi (detik)	Hasil Ditemukan
Linear Search	0.0330	1
Binary Search	0.0198	1
SQL Query	0.0059	1

Berdasarkan Tabel 7, ketiga metode menunjukkan waktu eksekusi yang relatif stabil pada pengujian berulang. *SQL Query* tetap menjadi metode tercepat, diikuti oleh *Binary Search* dan *Linear Search*. Hasil ini menunjukkan bahwa implementasi sistem mampu mempertahankan konsistensi kinerja pada kondisi pengujian yang sama. Secara keseluruhan, *SQL Query* merupakan metode yang paling efisien pada seluruh skenario pengujian. Meskipun demikian, *Binary Search* masih memberikan kinerja yang baik pada data numerik yang telah terurut, sedangkan *Linear Search* tetap relevan digunakan pada dataset berukuran kecil atau pada kondisi ketika data tidak memungkinkan untuk diurutkan maupun diindeks terlebih dahulu.

3.3 Pembahasan

Berdasarkan seluruh skenario pengujian, hipotesis penelitian yang menyatakan bahwa terdapat perbedaan kinerja antara metode *Linear Search*, *Binary Search*, dan *SQL Query* dapat diterima. Perbedaan kinerja tersebut terlihat dari rata-rata waktu eksekusi yang dihasilkan oleh masing-masing metode pada setiap skenario pengujian. Hasil ini menunjukkan adanya perbedaan karakteristik kinerja antar metode pencarian pada dataset yang digunakan dalam penelitian ini. Perbedaan kinerja tersebut menunjukkan bahwa pemilihan algoritma merupakan faktor penting dalam menentukan efisiensi sistem karena setiap algoritma memiliki karakteristik kompleksitas komputasi yang berbeda dan akan menghasilkan performa yang berbeda pula ketika diterapkan pada ukuran data tertentu (Sundaramoorthy & Karunanidhi, 2025).

Pada pengujian berdasarkan Nomor Induk Santri (NIS), metode *Binary Search* menghasilkan waktu eksekusi yang lebih rendah dibandingkan *Linear Search*. Hasil ini sesuai dengan karakteristik *Binary Search* yang bekerja dengan membagi ruang pencarian menjadi dua bagian secara berulang sehingga memiliki kompleksitas waktu $O(\log n)$, sedangkan *Linear Search* memiliki kompleksitas waktu $O(n)$ karena harus memeriksa data secara sekuensial. Oleh karena itu, *Binary Search* lebih sesuai diterapkan pada data numerik yang telah terurut, seperti data NIS.

Meskipun demikian, metode *SQL Query* menunjukkan performa terbaik pada seluruh skenario pengujian dengan rata-rata waktu eksekusi yang paling rendah. Hasil ini menunjukkan bahwa mekanisme pencarian pada tingkat basis data lebih efisien dibandingkan algoritma pencarian pada tingkat aplikasi. Hasil ini juga didukung oleh penelitian mengenai optimasi sistem basis data yang menunjukkan bahwa pemanfaatan mekanisme pencarian pada DBMS dapat menghasilkan waktu respons yang lebih baik dibandingkan implementasi pencarian pada tingkat aplikasi, terutama ketika sistem memanfaatkan strategi optimasi query dan struktur indeks secara efektif (Li et al., 2024). Kinerja tersebut dipengaruhi oleh penggunaan indexing, query optimizer, dan execution plan yang dimiliki oleh DBMS sehingga proses pencarian tidak selalu dilakukan secara sekuensial terhadap seluruh data. Penelitian mengenai optimasi query pada sistem basis data juga menunjukkan bahwa strategi optimasi query yang tepat dapat meningkatkan efisiensi eksekusi dan mengurangi waktu respons pencarian secara signifikan (Du et al., 2024). Temuan ini sejalan dengan penelitian Saidu et al. (2024) dan Sun et al. (2024) yang menyatakan bahwa optimasi query dan struktur indeks dapat meningkatkan efisiensi akses data secara signifikan pada sistem berbasis web. Penerapan indexing pada basis data relasional juga terbukti mampu mempercepat proses pencarian dan meningkatkan performa query, terutama pada tabel yang memiliki jumlah data yang terus bertambah (Anchlia, 2024).

Dalam pengujian data teks, metode *SQL Query* menunjukkan waktu eksekusi yang lebih rendah dibandingkan dengan *Linear Search*. Selain memanfaatkan optimasi internal DBMS, *SQL Query* memiliki kemampuan melakukan pencarian menggunakan operator LIKE, sehingga pengguna dapat mencari data menggunakan sebagian kata kunci. Kondisi ini memberikan fleksibilitas yang lebih tinggi dibandingkan pencarian sekuensial pada tingkat aplikasi dan dapat mengurangi beban pemrosesan pada sisi aplikasi. Kemampuan DBMS dalam mengelola proses pencarian dan pengambilan data secara terpusat juga berkontribusi terhadap peningkatan efisiensi sistem informasi berbasis web.



Kondisi tersebut terlihat pada penelitian ini, di mana SQL Query secara konsisten menghasilkan waktu eksekusi yang lebih rendah dibandingkan Linear Search pada seluruh skenario pengujian. Temuan ini sejalan dengan penelitian sebelumnya yang menyatakan bahwa pemanfaatan mekanisme pengolahan data pada basis data dapat mengurangi beban komputasi pada aplikasi dan meningkatkan performa sistem secara keseluruhan (Domdouzis, 2025).

Di sisi lain, metode *Linear Search* tetap memiliki kelebihan dari segi implementasi karena tidak memerlukan proses pengurutan data maupun struktur indeks tambahan. Oleh karena itu, metode ini masih relevan digunakan pada dataset berukuran kecil atau pada kondisi ketika data tidak memungkinkan untuk diurutkan dan diindeks terlebih dahulu. Hasil penelitian ini sejalan dengan temuan Lutfina et al. (2022) yang menyatakan bahwa Linear Search memiliki implementasi yang sederhana, tetapi kurang efisien ketika jumlah data meningkat.

Pengujian pada kondisi data yang tidak ditemukan menunjukkan bahwa seluruh metode dapat menangani pencarian tanpa menghasilkan kesalahan sistem. Selain itu, hasil pengujian berulang menunjukkan bahwa ketiga metode memiliki kinerja yang relatif stabil pada lingkungan pengujian yang sama. Namun, SQL Query tetap menunjukkan konsistensi waktu eksekusi yang lebih baik dibandingkan metode lainnya.

Secara empiris, hasil penelitian ini menunjukkan bahwa algoritma pencarian pada tingkat aplikasi masih memiliki relevansi pada kondisi tertentu. *Binary Search* dapat menjadi alternatif yang efisien ketika data telah terurut dan diproses di dalam memori aplikasi, sedangkan *Linear Search* sesuai digunakan pada data yang sederhana dan berukuran kecil. Akan tetapi, untuk sistem informasi berbasis web yang mengelola data dalam basis data relasional, pemanfaatan *SQL Query* dengan dukungan mekanisme *indexing* dan *query optimization* merupakan pendekatan yang lebih efektif dan efisien. Selain aspek performa, penerapan optimasi query dan pengelolaan data yang baik juga berpengaruh terhadap kualitas layanan sistem informasi karena dapat meningkatkan kecepatan respons dan pengalaman pengguna ketika melakukan pencarian data secara berulang (Pagay-Cierva, 2024).

Kontribusi utama penelitian ini adalah memberikan bukti empiris mengenai perbedaan karakteristik pencarian pada tingkat aplikasi dan tingkat basis data dalam satu lingkungan sistem informasi yang sama. Penelitian ini tidak hanya membandingkan kompleksitas algoritma secara teoritis, tetapi juga mengevaluasi implementasinya secara langsung pada sistem informasi pengelolaan data santri berbasis web. Pendekatan evaluasi berbasis implementasi nyata seperti ini penting dilakukan karena hasil pengujian pada lingkungan operasional dapat memberikan gambaran yang lebih representatif mengenai perilaku sistem dibandingkan analisis teoritis semata (Rifan et al., 2024). Selain itu, sistem yang dikembangkan memberikan manfaat praktis bagi pondok pesantren dalam meningkatkan efisiensi pengelolaan data, mempercepat proses pencarian informasi, dan mendukung kegiatan administrasi secara lebih terstruktur. Penerapan sistem informasi berbasis web juga memungkinkan proses pengelolaan data dilakukan secara terpusat dan terintegrasi sehingga memudahkan penyimpanan, pengelolaan, dan akses data dalam jangka panjang (Sudrajat et al., 2024). Penerapan sistem informasi berbasis web pada bidang pendidikan terbukti mampu meningkatkan efektivitas pengelolaan data, mempercepat proses administrasi, dan mempermudah akses terhadap informasi yang dibutuhkan oleh pengguna sistem (Permana et al., 2024).

Berdasarkan hasil pengujian pada dataset sebanyak 1.000 data santri, SQL Query menunjukkan waktu eksekusi yang paling rendah dibandingkan Linear Search dan Binary Search. Temuan ini menunjukkan bahwa pemanfaatan mekanisme optimasi pada DBMS memberikan keuntungan pada lingkungan pengujian yang digunakan dalam penelitian ini. Namun, hasil tersebut tidak dapat digeneralisasikan untuk dataset dengan skala yang lebih besar atau lingkungan sistem yang berbeda. Oleh karena itu, penelitian lanjutan dengan jumlah data yang lebih besar dan skenario pengujian yang lebih beragam masih diperlukan untuk memperoleh pemahaman yang lebih komprehensif mengenai karakteristik kinerja masing-masing metode pencarian.

4. KESIMPULAN

Penelitian ini berhasil merancang dan mengimplementasikan sistem informasi pengelolaan data santri berbasis web menggunakan framework Laravel dan basis data MariaDB. Sistem yang dikembangkan mampu mendukung proses pengelolaan dan pencarian data santri secara lebih terstruktur dibandingkan pengelolaan secara manual. Hasil pengujian menunjukkan bahwa terdapat perbedaan kinerja antara metode *Linear Search*, *Binary Search*, dan *SQL Query*. Pada dataset operasional sebanyak 1.000 data santri, *SQL Query* menghasilkan waktu eksekusi paling rendah pada seluruh skenario pengujian, sedangkan *Binary Search* menunjukkan kinerja yang lebih baik dibandingkan *Linear Search* pada pencarian data numerik yang telah terurut. Hasil ini menunjukkan bahwa mekanisme pencarian berbasis DBMS dengan dukungan *indexing* dan *query optimization* memberikan kinerja yang lebih baik dibandingkan algoritma pencarian pada tingkat aplikasi pada lingkungan pengujian yang digunakan. Namun, hasil penelitian ini terbatas pada penggunaan dataset sebanyak 1.000 data sehingga tidak dapat digeneralisasi secara langsung untuk sistem dengan jumlah data yang jauh lebih besar. Oleh karena itu, penelitian selanjutnya disarankan menggunakan dataset yang lebih besar dan mengevaluasi metode pencarian lain serta pengaruh konfigurasi basis data dan mekanisme caching terhadap kinerja pencarian.

REFERENCES

- Al-Hawari, F. (2022). Software design patterns for data management features in web-based information Systems. *Journal of King Saud University - Computer and Information Sciences*, 34(10), 10028–10043.



- <https://doi.org/10.1016/j.jksuci.2022.10.003>
- Anchlia, A. (2024). Enhancing Query Performance Through Relational Database Indexing. *International Journal of Computer Trends and Technology*, 72(8), 130–133. <https://doi.org/10.14445/22312803/ijctt-v72i8p119>
- Domdouzis, K. (2025). Development of a Web-based System Dynamics Simulation and Benchmarking Environment for Medical Workforce Planning. *Procedia Computer Science*, 263, 276–284. <https://doi.org/10.1016/j.procs.2025.07.034>
- Du, Y., Cai, Z., & Ding, Z. (2024). Query Optimization in Distributed Database Based on Improved Artificial Bee Colony Algorithm. *Applied Sciences (Switzerland)*, 14(2), 846. <https://doi.org/10.3390/app14020846>
- Fan, Y., Weng, Q., Chen, Y., & Wang, X. S. (2025). Rethinking data in NL2SQL: a survey of what we have and what we expect. *Vicinagearth*, 2, 15. <https://doi.org/10.1007/s44336-025-00026-9>
- Fitriani, L. D., & Puspitaningrum, A. C. (2023). Utilization Of UML (Unified Modeling Language) In The Design Of Academic Information Systems Based On The OOAD Method. *Sistemasi*, 12(2), 614. <https://doi.org/10.32520/stmsi.v12i2.2871>
- Li, X., Wang, T., Wang, D., Zhang, H., Zhou, Y., & Zhou, D. (2024). Intelligent decision-making algorithm for airborne phased array radar search tasks based on a hierarchical strategy framework. *Chinese Journal of Aeronautics*, 37(11), 398–419. <https://doi.org/10.1016/j.cja.2024.09.006>
- Lontaan, R. J., & Sinadia, A. R. (2024). Design and Development of a Web-Based School Information System. *CogITO Smart Journal*, 10(2), 593–606. <https://doi.org/10.31154/cogito.v10i2.858.593-606>
- Lutfina, E., Inayati, N., & Saraswati, G. W. (2022). Analisis Perbandingan Kinerja Metode Rekursif dan Metode Iteratif dalam Algoritma Linear Search. *Komputika: Jurnal Sistem Komputer*, 11(2), 143–150. <https://doi.org/10.34010/komputika.v11i2.5493>
- Pagay-Cierva, M. J. (2024). Evaluation on the Performance of the Developed Mobile-Based Registrar School Record Management and E-Payment with SMS Notification. *International Journal of Research and Scientific Innovation*, XI(VII), 814–828. <https://doi.org/10.51244/ijrsi.2024.1107064>
- Pasaribu, J. S., & Argadikusuma, I. S. (2024). Design and Testing of a Web-Based Student Information Management System. *International Journal of Engineering, Science and Information Technology*, 4(4), 144–155. <https://doi.org/10.52088/ijesty.v4i4.594>
- Permana, A. I., Syahputra, Z., & Saragih, M. D. A. (2024). Design and Construction of a Web-Based Student Data. *Journal of Science and Social Research*, 7(4), 1549–1555. <http://jurnal.goretanpena.com/index.php/JSSR>
- Rifan, T., Akbar, F. A., & Nurlaili, A. L. (2024). Sistem Asesmen Pembelajaran Sql Dengan Metode Cosine Similarity Dan Sql Parser Berbasis Website. *JIPi (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 9(4), 2097–2111. <https://doi.org/10.29100/jipi.v9i4.5630>
- Saidu, I. C., Yusuf, M., Nemariyi, F. C., & George, A. C. (2024). Indexing techniques and structured queries for relational databases management systems. *Journal of the Nigerian Society of Physical Sciences*, 6(4), 1–12. <https://doi.org/10.46481/jnsps.2024.2155>
- Singh, V., Khanzadeh, M., Davis, V., Rush, H., Rossi, E., Shrader, J., & Lio', P. (2025). Bayesian Binary Search. *Algorithms*, 18(8), 1–27. <https://doi.org/10.3390/a18080452>
- Siregar, H., & Arpan, A. (2025). Rancangan Bangun Aplikasi Pembelajaran Online (E-Learning) di SMK Tritech Informatika Medan Berbasis Web dengan Metode Waterfall. *Jurnal Nasional Teknologi Komputer*, 5(3), 248–262. <https://publikasi.hawari.id/index.php/jnastek/article/view/210>
- Sudrajat, R., Ruchjana, B. N., Abdullah, A. S., & Budiarto, R. (2024). Web-based information system framework for the digitization of historical databases and endowments. *International Journal of Data and Network Science*, 8(1), 319–328. <https://doi.org/10.5267/j.ijdns.2023.9.022>
- Sun, B., Zhao, S., & Tian, G. (2024). SQL queries over encrypted databases: a survey. *Connection Science*, 36(1), Article 2323059. <https://doi.org/10.1080/09540091.2024.2323059>
- Sundaramoorthy, S., & Karunanidhi, G. (2025). A systematic analysis on performance and computational complexity of sorting algorithms. *Discover Computing*, 28(1), Article 250. <https://doi.org/10.1007/s10791-025-09724-w>
- Shelke, K., Khan, Y., & Kapse, A. S. (2025). Student Management System: A Web-Based Solution for Academic Administration. *International Journal of Scientific Research in Science and Technology*, 12(3), 50–54. <https://doi.org/10.32628/ijrsrst251222719>