



IMPLEMENTASI ALGORITMA SEQUENTIAL SEARCH DALAM MELAKUKAN PENCARIAN DATA SKRIPSI STMIK KAPUTAMA

Daviena Elfira ¹, Dilla Ayu Azwari ², Hani Harmaini ³, Safrizal ⁴

^{1,2,3} Program Studi Sistem Informasi, Fakultas Ilmu Komputer, STMIK Kaputama Binjai

⁴ Program Studi Sistem Informasi, Fakultas Ekonomi Sains dan Teknologi, Universitas Muhammadiyah Asahan

E-mail: dvna2801@gmail.com ¹, dillaayuazwari@gmail.com ², haniharmaini46@gmail.com ³, rizalsyl75@gmail.com ⁴

ABSTRACT

The management of thesis data at STMIK Kaputama still encounters challenges in the retrieval process, especially as the amount of data increases each year while most search activities remain manual. This condition often leads to slow performance, inefficiency, and potential inaccuracies in decision-making. This study aims to describe the implementation of the Sequential Search algorithm formula into a web-based thesis data search prototype that operates entirely on the client side. A qualitative descriptive-analytical approach was used, focusing on literature review and direct examination of the algorithmic formula. The prototype was developed using HTML for the interface structure, CSS for visual layout, and JavaScript to execute the algorithm's logical operations. Thesis data are imported through a CSV file and transformed into a structured dataset, which is then processed using the Sequential Search formula to compare the search key sequentially with each data entry. Testing with a dataset of ten entries shows that the system successfully executes the search process according to the mathematical model: when a match is found, the prototype displays the position and complete attributes of the corresponding entry; when no match exists, the system clearly indicates that the data is not found. These results confirm that translating the Sequential Search formula into JavaScript code can be applied effectively and produces search behavior consistent with its theoretical foundation. The developed prototype demonstrates that this algorithm remains suitable as a simple and structured retrieval method for small- to medium-scale data environments at STMIK Kaputama.

Keywords: *Sequential Search, thesis data retrieval, search algorithm, JavaScript, web-based system.*

ABSTRAK

Pengelolaan data skripsi di lingkungan STMIK Kaputama masih menghadapi tantangan pada aspek pencarian, terutama ketika data terus bertambah setiap tahun sementara proses temu kembali

Article History

Received: Oktober 2025

Reviewed: Oktober 2025

Published: Oktober 2025

Plagiarism Checker No 235

Prefix DOI :

[10.8734/Kohesi.v1i2.365](https://doi.org/10.8734/Kohesi.v1i2.365)

Copyright : Author

Publish by : Kohesi



This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/)



sebagian besar dilakukan secara manual. Kondisi ini sering menyebabkan pencarian menjadi lambat, tidak efisien, dan berpotensi menimbulkan kesalahan dalam pengambilan keputusan. Penelitian ini bertujuan menjelaskan penerapan rumus algoritma *Sequential Search* ke dalam sebuah prototipe sistem pencarian data skripsi berbasis web yang bekerja sepenuhnya pada sisi klien. Penelitian menggunakan pendekatan kualitatif deskriptif-analitis dengan fokus pada studi pustaka dan pembacaan langsung terhadap rumus algoritmik. Implementasi dilakukan menggunakan HTML sebagai kerangka antarmuka, CSS untuk tampilan visual, dan JavaScript untuk mengeksekusi logika algoritma. Data skripsi dimasukkan melalui file CSV, kemudian diolah menjadi himpunan berstruktur yang selanjutnya diproses oleh rumus *Sequential Search* untuk membandingkan kunci pencarian terhadap setiap entri secara berurutan. Pengujian menggunakan dataset berisi sepuluh entri menunjukkan bahwa sistem mampu menjalankan proses pencarian sesuai rumus matematika yang digunakan: ketika data ditemukan, sistem menampilkan posisi dan atribut lengkap entri terkait; ketika tidak ditemukan, sistem memberikan informasi yang jelas bahwa tidak ada entri yang sesuai. Hasil ini menegaskan bahwa transformasi rumus *Sequential Search* ke dalam kode JavaScript dapat diterapkan secara efektif dan menghasilkan perilaku pencarian yang konsisten dengan teori. Prototipe sistem yang dikembangkan membuktikan bahwa algoritma ini masih relevan sebagai solusi pencarian sederhana dan terstruktur untuk skala data kecil hingga menengah di lingkungan STMIK Kaputama.

Kata kunci: *Sequential Search, pencarian data skripsi, algoritma pencarian, JavaScript, sistem berbasis web.*

1. PENDAHULUAN

Institusi pendidikan tinggi, seperti STMIK Kaputama, mengelola volume data akademik yang terus meningkat setiap tahunnya. Salah satu bentuk data yang paling penting adalah koleksi karya ilmiah mahasiswa, khususnya data buku skripsi. Seiring bertambahnya jumlah koleksi, proses untuk menemukan kembali (*retrieval*) data skripsi tertentu secara manual menjadi tidak lagi efisien dan rentan terhadap ketidakakuratan (Febryanto, 2022). Pengelolaan koleksi buku secara manual, terutama dalam hal pendataan dan pencarian, sering kali belum maksimal dan dapat menimbulkan ketidakakuratan dalam pelaporan (Setyawati Wahyuni et al., n.d.). Oleh karena itu, diperlukan sebuah mekanisme pencarian yang sistematis untuk mengakses data tersebut secara efektif.

Dalam ilmu komputer, proses menemukan data atau nilai tertentu di dalam sekumpulan data dikenal sebagai proses pencarian (*searching*) (Suparman & Azizah, 2022). Proses ini



merupakan salah satu operasi fundamental dalam pengolahan data. Terdapat berbagai jenis algoritma pencarian yang telah dikembangkan, namun salah satu yang paling dasar dan sering menjadi titik awal pembahasan adalah algoritma *Sequential Search* (pencarian sekuensial), yang juga dikenal dengan istilah pencarian beruntun (Rahmanto et al., n.d.). Algoritma ini merupakan metode pencarian yang paling sederhana jika dibandingkan dengan algoritma pencarian lainnya (Setyawati Wahyuni et al., n.d.).

Secara konseptual, fokus utama dari algoritma *sequential search* adalah pada proses perbandingan data secara *linear*. Logika atau "rumus" dari algoritma ini bekerja dengan cara membandingkan data yang dicari (kunci pencarian atau *key*) dengan setiap elemen yang ada dalam kumpulan data (biasanya disimpan dalam sebuah *array* atau *list*), satu per satu, secara berurutan (Suparman & Azizah, 2022). Proses logis ini dapat dijabarkan sebagai berikut: (1) Proses dimulai dari elemen pertama (indeks ke-0). (2) Data pada elemen saat ini ($Data[i]$) dibandingkan dengan kunci pencarian (x). (3) Jika $Data[i]$ sama dengan x , maka pencarian dianggap berhasil (ditemukan) dan proses dihentikan. (4) Jika tidak sama, proses berlanjut ke elemen berikutnya ($i + 1$) dan langkah perbandingan diulangi. (5) Proses ini akan terus berlangsung hingga data ditemukan atau hingga seluruh elemen dalam kumpulan data telah diperiksa (Suparman & Azizah, 2022). Karakteristik utama dari logika ini adalah data yang tersimpan tidak perlu diurutkan terlebih dahulu (Ardi Widodo et al., 2021).

Penelitian ini bertujuan untuk menerapkan dan menganalisis implementasi algoritma *sequential search* dalam konteks pencarian data buku skripsi di STMIK Kaputama. Untuk memvisualisasikan penerapan rumus dan logika ini secara praktis, penelitian ini akan menguraikan proses implementasi sebuah prototipe sistem pencarian sederhana. Prototipe ini dikembangkan sebagai aplikasi berbasis web dengan menggunakan teknologi fundamental yaitu *HTML* untuk membangun struktur antarmuka pencarian, *CSS* untuk melakukan penataan visual agar antarmuka lebih baik, dan *JavaScript* untuk mengeksekusi logika inti (rumus) dari algoritma *sequential search* itu sendiri. Berbeda dengan penelitian yang berfokus pada performa *server-side* atau kecepatan *database* (Ardi Widodo et al., 2021), fokus utama jurnal ini adalah pada penjabaran proses algoritmik di sisi klien (*client-side*) dan bagaimana logika *sequential search* (Suparman & Azizah, 2022) secara langsung diterapkan dalam kode *JavaScript* untuk berinteraksi dengan himpunan data skripsi dan menampilkan hasilnya kepada pengguna.



2. METODE PENELITIAN

Penelitian ini menggunakan metode penelitian kualitatif dengan pendekatan deskriptif-analitis. Fokus utama dari penelitian ini adalah studi pustaka (*literature review*) dan analisis algoritma. Metode ini dipilih karena tujuan penelitian adalah untuk membedah dan menganalisis proses logis (rumus) dari algoritma *sequential search* itu sendiri, bukan untuk mengukur kinerja implementasi sistemnya. Data teoretis dikumpulkan dari berbagai jurnal ilmiah dan literatur akademis yang relevan (seperti (Rahmanto et al., n.d.); (Suparman & Azizah, 2022); (Ardi Widodo et al., 2021)) untuk membangun landasan teoretis yang kuat. Analisis dilakukan dengan cara membedah logika algoritma, menyajikannya dalam bentuk pseudocode, dan mengilustrasikannya menggunakan contoh dataset hipotetis yang relevan dengan data skripsi STMIK Kaputama.

2.1 Algoritma Sequential Search

Algoritma *Sequential Search* atau pencarian beruntun merupakan metode pencarian data paling sederhana dalam ilmu komputer (Setyawati Wahyuni et al., n.d.). Konsep dasarnya adalah membandingkan kunci pencarian (*key*) dengan setiap elemen data di dalam kumpulan data (biasanya berupa array atau list), satu per satu, hingga data yang dicari ditemukan atau seluruh elemen telah diperiksa (Febryanto, 2022).

Kelebihan utama dari algoritma ini terletak pada **kesederhanaannya**, karena tidak memerlukan data dalam keadaan terurut (sorted) (Ardi Widodo et al., 2021). Namun, karena bersifat linear, waktu eksekusi akan meningkat seiring dengan bertambahnya jumlah data.

Rumus dasar dari logika **Sequential Search** dapat dituliskan sebagai berikut:

Diketahui:

Diketahui: $D = [D1, D2, D3, \dots, Dn]$ dan kunci pencarian K

Proses Pencarian:

Untuk setiap $i = 1$ sampai n : $\sum$ *Jika Di=K, maka data ditemukan pada posisi ke-i*
Jika tidak maka lanjut ke elemen berikutnya

Jika seluruh D_i telah diperiksa dan tidak ada yang sama dengan K , maka:



Hasil = "Data tidak ditemukan"

Dengan demikian, proses kerja algoritma dapat dirumuskan sebagai:

$$Output = \sum \begin{matrix} "Data ditemukan", & \text{jika } D_i=K \\ "Data tidak ditemukan", & \text{jika } i>n \end{matrix}$$

2.2 Cara Kerja Algoritma Sequential Search

Algoritma membandingkan kunci pencarian (K) terhadap setiap elemen data secara berurutan sambil memakai bendera (*flag*) found untuk menandai apakah data sudah ditemukan.

Dataset Contoh (n = 6)

Tabel 1. Ilustrasi dataset sampel

Nomor Buku	Judul	Nama Penulis	Tahun terbit	Jurusan
1002	Aplikasi Booking Bengkel Motor AHASS Sriwijaya Motor Berbasis Android	Putra Dermawan	2023	Sistem Informasi
1005	Deteksi Face Spoofing pada Sistem Absensi PT Semen Baturaja Menggunakan CNN	Silvia Andini	2024	Teknik Informatika
1001	Sistem Informasi Pemetaan Titik Banjir Kelurahan 15 Ulu Menggunakan GIS	Yoga Pratomo	2024	Sistem Informasi
1009	Analisis Sentimen Pelanggan PDAM Tirta Musi Menggunakan Naive Bayes	Hendra Kurniawan	2025	Teknik Informatika
1004	Aplikasi Mobile Penjadwalan Praktikum Laboratorium Teknik Mesin Polsri	Nabila Ramadhani	2025	Sistem Informasi
1003	Sistem Rekomendasi Produk UMKM Kuliner Palembang Menggunakan Collaborative Filtering	Zulfikar Hadi	2024	Sistem Informasi
1006	Deteksi Plat Nomor Kendaraan Menggunakan YOLOv5 di Parkiran Kantor Gubernur	Stefani Gunawan	2025	Teknik Informatika

Rumus Logika Operasional

Input: ($D = [D_1, \dots, D_6]$, K dengan setiap D_i memuat atribut (No. Buku, Judul, Penulis, Tahun, Jurusan).

Kunci pencarian (K) bisa berupa judul (string penuh), penulis, atau atribut lain (d disesuaikan kriteria yang dipakai).

Langkah-langkah:

1. Inisialisasi:



$$i = 1, found = 0$$

2. Perulangan

$$SELAMA \ i \leq 6 \text{ dan } found = 0: \sum_{\substack{\text{Jika Judul}(Di) = K \Rightarrow found = 1 \\ \text{Jika tidak} \Rightarrow i \leftarrow i + 1}}$$

3. Keputusan akhir:

$$\sum_{\substack{\text{Jika } found = 1 \Rightarrow \text{"Data ditemukan"} \text{ di indeks } i \text{ (laporkan seluruh atribut } Di) \\ \text{Jika } found = 0 \Rightarrow \text{"Data tidak ditemukan"}}$$

Contoh A – Pencarian Judul (Data Ditemukan)

K = “Rancang Bangun Sistem IoT untuk Monitoring Kesehatan dengan Algoritma Kalman Filter”. Perbandingan berlangsung pada D_1, D_2, D_3 tidak cocok; pada D_4 cocok, sehingga $found = 1$ dan proses berhenti.

Hasil: ditemukan pada $i = 4$ dengan atribut lengkap: **SKP-004, Dewi Kartika, 2023, Teknik Informatika.**

Contoh B – Pencarian Judul (Data Tidak Ditemukan)

K = “Implementasi Data Science untuk Prediksi Tren Ekonomi menggunakan Algoritma ARIMA”. Seluruh judul D_1 sampai dengan D_6 tidak cocok; loop berakhir ketika $i = 7$ dengan $found = 0$.

Hasil: **data tidak ditemukan.**

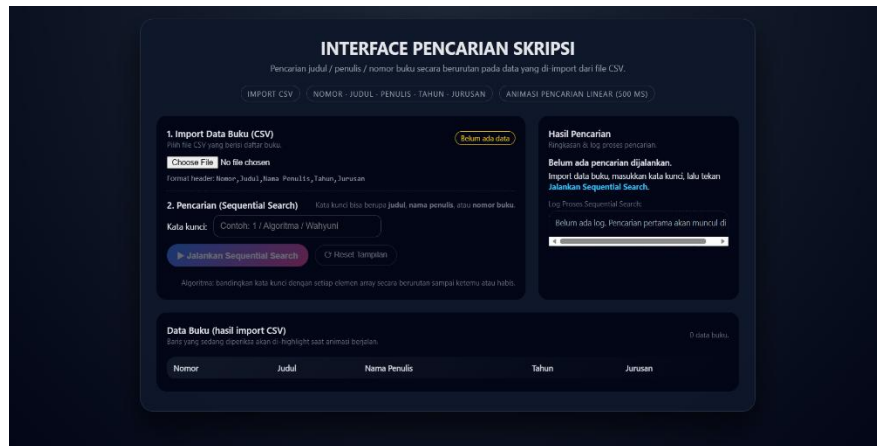
3. HASIL DAN PEMBAHASAN

3.1 Implementasi Sistem Pencarian

Implementasi algoritma Sequential Search pada penelitian ini direalisasikan melalui sebuah prototipe aplikasi web yang berfungsi sebagai alat pencarian data skripsi. Sistem ini dibangun berbasis client-side menggunakan kombinasi HTML, CSS, dan *JavaScript*, sesuai dengan fokus penelitian yang menekankan penerapan logika algoritmik pada sisi pengguna (client) sebagaimana dikaji dalam dokumen referensi.

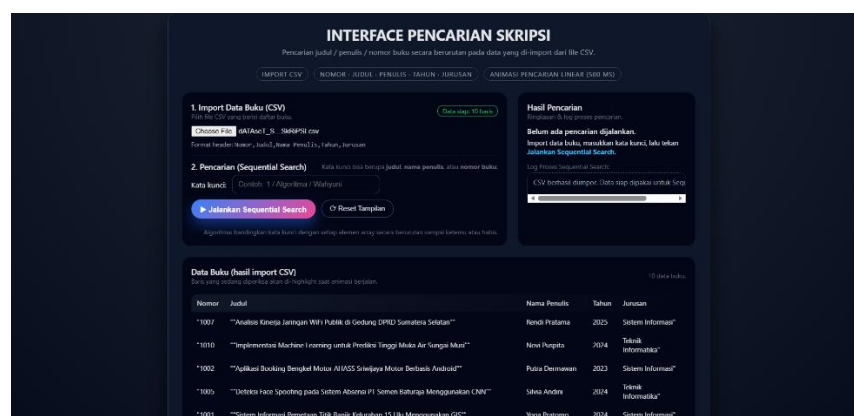
Antarmuka sistem dirancang agar sederhana dan mudah dioperasikan. Pada tahap awal, pengguna akan melihat tampilan dasar aplikasi yang terdiri dari menu *import CSV*, kolom pencarian, serta tabel kosong yang menunggu data untuk ditampilkan. Gambar berikut menunjukkan tampilan awal tersebut sebelum pengguna mengunggah file dataset:

Gambar 3.1. Tampilan awal antarmuka sebelum data di-import



Tahap selanjutnya adalah proses *import file CSV* yang berisi daftar skripsi. Sistem menerima file dengan struktur header: *Nomor*, *Judul*, *Nama Penulis*, *Tahun*, *Jurusan*. Setelah pengguna memilih file, *JavaScript* memproses data dan mengubahnya menjadi *array* objek yang siap digunakan. Ketika proses *import* berhasil, sistem menampilkan pesan konfirmasi serta memunculkan tabel data. Tahapan ini ditampilkan pada gambar berikut:

Gambar 3.2. Sistem setelah file berhasil di-import dan siap diproses

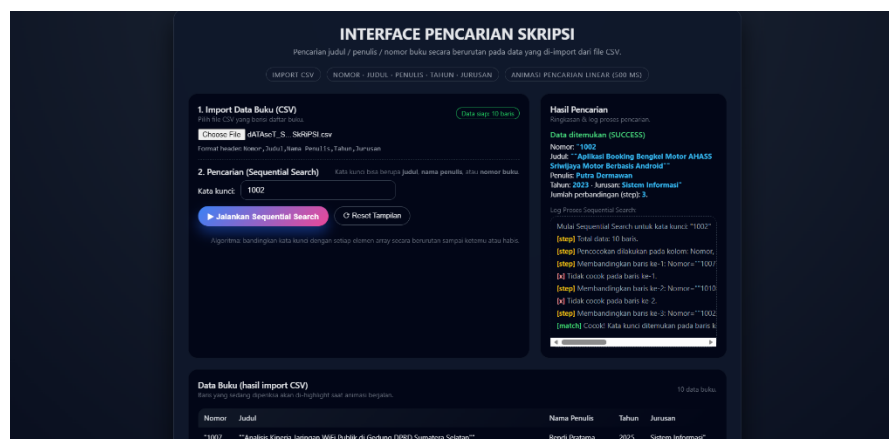


Setelah data berhasil dimuat, pengguna dapat memasukkan kata kunci pencarian, yang dapat berupa nomor buku, judul skripsi, atau nama penulis. *JavaScript* kemudian menjalankan algoritma *Sequential Search*: membandingkan kata kunci dengan setiap elemen secara

berurutan, dimulai dari indeks pertama hingga ditemukan kecocokan atau sampai seluruh data diperiksa, sebagaimana prinsip algoritma yang dijelaskan dalam teori Sequential Search .

Ketika proses pencarian dijalankan, sistem menampilkan animasi pemeriksaan baris, log pencarian *step-by-step*, dan hasil akhir. Apabila data cocok ditemukan, seluruh atribut ditampilkan lengkap beserta jumlah langkah (steps) yang dibutuhkan. Gambar berikut menunjukkan tampilan hasil pencarian ketika data ditemukan:

Gambar 3.3. Hasil pencarian ketika kata kunci cocok



Implementasi ini memberikan pengalaman pencarian yang interaktif dan informatif, sekaligus memperlihatkan bagaimana algoritma *Sequential Search* bekerja secara visual. Seluruh proses berjalan secara *real-time* di sisi klien tanpa membutuhkan server backend, menjadikannya solusi yang ringan namun efektif untuk kebutuhan pencarian pada dataset berskala kecil-menengah.

3.2 Pengujian Proses Pencarian Menggunakan Dataset

Pengujian algoritma *Sequential Search* pada penelitian ini dilakukan menggunakan sebuah himpunan data skripsi berjumlah 10 entri. Setiap entri memuat lima atribut, yaitu Nomor Buku, Judul, Nama Penulis, Tahun Terbit, dan Jurusan. Dataset ini disusun untuk merepresentasikan kondisi riil data skripsi di lingkungan STMIK Kaputama dan sekaligus menjadi dasar implementasi rumus *Sequential Search* yang telah dipaparkan pada bagian landasan teori.

Dataset yang digunakan disajikan pada Tabel 2.

Tabel 2. Dataset sampel skripsi untuk pengujian algoritma Sequential Search



Nomor Buku	Judul	Nama Penulis	Tahun terbit	Jurusan
1007	Analisis Kinerja Jaringan WiFi Publik di Gedung DPRD Sumatera Selatan	Rendi Pratama	2025	Sistem Informasi
1010	Implementasi Machine Learning untuk Prediksi Tinggi Muka Air Sungai Musi	Novi Puspita	2024	Teknik Informatika
1002	Aplikasi Booking Bengkel Motor AHASS Sriwijaya Motor Berbasis Android	Putra Dermawan	2023	Sistem Informasi
1005	Deteksi Face Spoofing pada Sistem Absensi PT Semen Baturaja Menggunakan CNN	Silvia Andini	2024	Teknik Informatika
1001	Sistem Informasi Pemetaan Titik Banjir Kelurahan 15 Ulu Menggunakan GIS	Yoga Pratomo	2024	Sistem Informasi
1009	Analisis Sentimen Pelanggan PDAM Tirta Musi Menggunakan Naive Bayes	Hendra Kurniawan	2025	Teknik Informatika
1004	Aplikasi Mobile Penjadwalan Praktikum Laboratorium Teknik Mesin Polsri	Nabila Ramadhani	2025	Sistem Informasi
1008	Monitoring Suhu Ruang Server Diskominfo Palembang Berbasis IoT	Azis Rahman	2023	Teknik Informatika
1003	Sistem Rekomendasi Produk UMKM Kuliner Palembang Menggunakan Collaborative Filtering	Zulfikar Hadi	2024	Sistem Informasi
1006	Deteksi Plat Nomor Kendaraan Menggunakan YOLOv5 di Parkiran Kantor Gubernur	Stefani Gunawan	2025	Teknik Informatika

Secara formal, dataset pada Tabel 2 dapat direpresentasikan sebagai himpunan berurut:

$$D = [D_1, D_2, D_3, \dots, D_{10}]$$

dengan setiap elemen (D_i) memuat lima komponen atribut:

$$D_i = (Nomor_i, Judul_i, Penulis_i, Tahun_i, Jurusan_i)$$

Pada saat proses pencarian dijalankan, pengguna menentukan **kunci pencarian** (K) yang dapat berupa salah satu nilai atribut (misalnya nomor buku, judul, atau nama penulis). Sistem kemudian menerapkan rumus *Sequential Search* sebagai berikut:

$$\text{Untuk setiap } i = 1 \text{ sampai } n: \begin{cases} \text{Jika } D_i = K, \text{ maka data ditemukan pada posisi ke-} i \\ \text{Jika tidak maka, } i \leftarrow i + 1 \end{cases}$$

dengan $n = 10$ sebagai jumlah total data pada tabel dan *Atribut* (D_i) adalah atribut yang dipilih sebagai dasar pencarian (misalnya $Nomor_i$ ketika pencarian berdasarkan nomor buku, atau $Judul_i$ ketika pencarian berdasarkan judul).

Dalam implementasi sistem, mekanisme di atas dioperasikan melalui fungsi JavaScript yang melakukan perulangan dari D_1 hingga D_{10} secara berurutan. Pada setiap iterasi, nilai atribut yang menjadi kriteria pencarian akan dibandingkan dengan kunci (K). Jika



terjadi kecocokan, fungsi mengembalikan indeks (i) beserta seluruh atribut data yang bersangkutan dan proses pencarian dihentikan. Sebaliknya, apabila hingga $i = 10$ tidak ditemukan kecocokan, sistem menyimpulkan bahwa data yang dicari tidak terdapat dalam himpunan (D) dan menampilkan pesan “data tidak ditemukan”.

Dengan demikian, poin 3.2 ini menegaskan bahwa pengujian algoritma dilakukan secara langsung di atas dataset nyata yang telah terstruktur, sehingga penerapan rumus *Sequential Search* tidak hanya bersifat teoretis, tetapi juga teruji melalui data skripsi yang menyerupai kondisi penggunaan di lingkungan STMIK Kaputama.

3.3 Hasil Pencarian (Kasus 1: Data Ditemukan)

Pengujian pertama dilakukan untuk melihat bagaimana algoritma *Sequential Search* bekerja ketika kunci pencarian memiliki kecocokan dengan salah satu entri data. Pada skenario ini digunakan kunci pencarian berupa nomor buku:

$$K = 1002$$

Kunci tersebut dibandingkan dengan atribut *Nomor Buku* pada setiap entri data secara berurutan. Prosedur pencarian mengikuti rumus dasar algoritma *Sequential Search* sebagaimana telah dijelaskan pada landasan teori :

$$\text{Untuk setiap } i = 1 \text{ sampai } n: \begin{cases} \text{Jika } (Di) = K, \text{ maka data ditemukan pada posisi ke-} i \\ \text{Jika } \text{Judul } (Di) = K, \quad i \leftarrow i + 1 \end{cases}$$

dengan Di menyatakan entri data ke- i dan $n = 10$ sebagai jumlah total data pada Tabel 1.

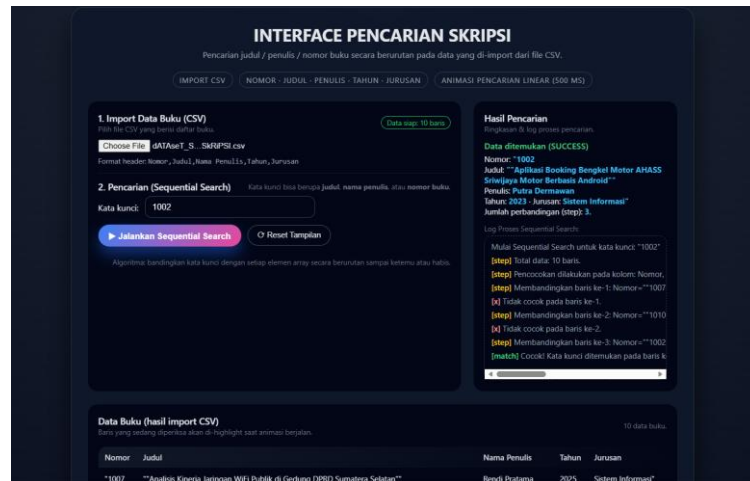
Proses pencarian dimulai dari entri pertama. Pada entri (D_1), nomor buku bernilai 1007 sehingga tidak sama dengan kunci 1002, sehingga algoritma melanjutkan ke entri berikutnya. Pada D_2 , nomor buku bernilai 1010 dan kembali tidak cocok. Pencocokan baru terjadi pada D_3 , di mana nomor buku bernilai 1002 dan sama dengan kunci pencarian. Ketika kondisi $Nomor(D_3) = K$ terpenuhi, variabel penanda (*found*) bernilai benar dan proses pencarian dihentikan pada iterasi ketiga.

Dengan demikian, data yang dicari ditemukan pada posisi ke-3, dengan rincian: nomor buku 1002, judul *Aplikasi Booking Bengkel Motor AHASS Sriwijaya Motor Berbasis Android*, nama penulis Putra Dermawan, tahun terbit 2023, dan jurusan Sistem Informasi. Jumlah langkah

pencarian yang dilakukan adalah tiga kali perbandingan, yang menunjukkan bahwa waktu eksekusi algoritma bergantung pada posisi elemen yang dicari dalam deret data.

Visualisasi hasil implementasi pada prototipe sistem ditunjukkan pada Gambar 3.3, yang menampilkan keluaran data yang ditemukan beserta informasi jumlah langkah pencarian dan log proses pemeriksaan dari baris pertama hingga baris ketiga.

Gambar 3.3 Tampilan sistem saat pencarian berhasil menemukan data



Dengan demikian, pengujian kasus pertama ini membuktikan bahwa implementasi algoritma *Sequential Search* pada prototipe sistem bekerja sesuai rumus dan logika dasar yang telah dijelaskan dalam landasan teori, serta mampu memberikan hasil pencarian secara akurat, sistematis, dan terstruktur.

3.4 Hasil Pencarian (Kasus 2: Data Tidak Ditemukan)

Pengujian kedua dilakukan untuk melihat bagaimana algoritma *Sequential Search* berperilaku ketika kunci pencarian tidak memiliki kecocokan dengan data mana pun pada himpunan. Pada pengujian ini digunakan kunci berupa judul skripsi fiktif:

$$K = \text{"Sistem Prediksi Pelanggan Cerdas Menggunakan Deep Learning"}$$

Judul tersebut tidak terdapat pada seluruh dataset, sehingga algoritma harus melakukan pemeriksaan terhadap setiap elemen secara berurutan. Prosedur pencarian tetap mengikuti rumus dasar *Sequential Search*:

$$\text{Untuk setiap } i = 1 \text{ sampai } n: \sum \begin{cases} \text{Jika } (D_i) = K, \text{ maka data ditemukan pada posisi ke-} i \\ \text{Jika Judul } (D_i) = K, \quad i \leftarrow i + 1 \end{cases}$$

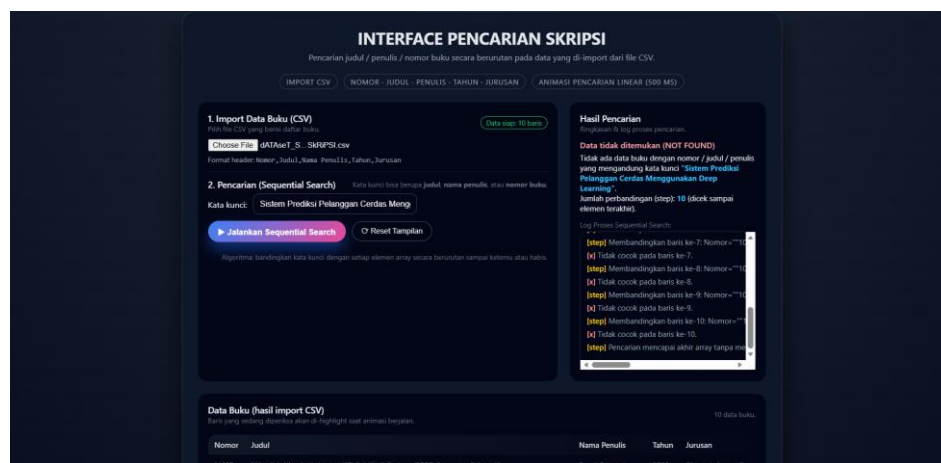
Proses dimulai dari elemen pertama hingga terakhir. Pada baris pertama hingga baris keenam, perbandingan tidak menghasilkan kecocokan. Hal yang sama terjadi pada baris ketujuh sampai baris kesepuluh. Dengan demikian, seluruh elemen telah diperiksa tanpa menemukan kecocokan, yang secara matematis ditandai dengan:

$$found = 0 \text{ untuk seluruh } i = 1 \text{ hingga } 10$$

Karena tidak terdapat entri yang sesuai dengan kunci pencarian, algoritma menyimpulkan bahwa data tidak ditemukan dalam himpunan. Situasi ini merupakan *worst-case scenario*, di mana jumlah langkah yang diperlukan sama dengan jumlah total data, yaitu sepuluh langkah pemeriksaan.

Visualisasi dari implementasi pencarian yang tidak menemukan hasil ditunjukkan pada Gambar 3.4 berikut, yang menampilkan tampilan sistem setelah proses pencarian selesai tanpa kecocokan:

Gambar 3.4 Tampilan sistem saat pencarian tidak menemukan kecocokan



Dengan demikian, pengujian kasus kedua ini membuktikan bahwa implementasi algoritma Sequential Search pada prototipe sistem bekerja tepat sebagaimana yang dijelaskan dalam landasan teori. Proses pencarian tetap berjalan secara sistematis hingga seluruh elemen diperiksa, dan sistem mampu memberikan keluaran berupa informasi *data tidak ditemukan* secara jelas, terstruktur, dan konsisten dengan karakteristik algoritma tersebut.

KESIMPULAN

Penelitian ini menyimpulkan bahwa rumus dan logika algoritma *Sequential Search* dapat diimplementasikan secara langsung dan efektif ke dalam sebuah prototipe sistem pencarian



data skripsi berbasis web. Rumus pencarian yang semula disajikan dalam bentuk notasi matematis dan uraian konseptual berhasil diterjemahkan ke dalam kode JavaScript yang bekerja di sisi klien, mulai dari proses inialisasi variabel, perulangan dari elemen pertama hingga terakhir, hingga pengujian kondisi kecocokan antara kunci pencarian dengan atribut data pada setiap entri .

Integrasi rumus ke dalam sistem tampak jelas pada alur kerja aplikasi: pengguna mengunggah dataset skripsi dalam format CSV, data kemudian dipetakan menjadi himpunan $D = [D1, D2, \dots, Dn]$, dan sistem menjalankan proses pencarian sesuai rumus *Sequential Search* dengan membandingkan kunci (K) terhadap atribut tertentu (misalnya nomor buku atau judul) secara berurutan. Hasil pengujian menunjukkan bahwa sistem mampu menampilkan keluaran yang sesuai dengan rumus, baik ketika data ditemukan (menampilkan posisi dan seluruh atribut entri yang cocok) maupun ketika data tidak ditemukan (memberikan informasi bahwa tidak ada entri yang sesuai). Dengan demikian, penelitian ini membuktikan bahwa implementasi rumus *Sequential Search* pada prototipe sistem tidak hanya tepat secara teoritis, tetapi juga berfungsi secara praktis untuk mendukung kebutuhan pencarian data skripsi di lingkungan STMIK Kaputama secara sederhana dan terstruktur.

DAFTAR PUSTAKA

- Alridho, S., & Khairunnisa, D. (n.d.). *Paper Implementasi Algoritma Sequential Search Dalam Game Edukasi Penyusunan Kalimat Terstruktur*.
- Ardi Widodo, K., Adi Wibowo, S., & Vendyansyah, N. (2021). PENERAPAN SEQUENTIAL SEARCH UNTUK PENGELOLAAN DATA BARANG. *Antivirus: Jurnal Ilmiah Teknik Informatika*, 15(1), 86-97. <https://doi.org/10.35457/antivirus.v15i1.1385>
- Febryanto, A. (2022). *Penerapan Algoritma Sequential Search untuk Mencari Data Siswa Pada Sekolah Menengah Kejuruan Negeri 3 Bengkalis*. 2(1), 51-59.
- Gunawan, G., Veronika, N. D. M., & Wijaya, A. (2019). Sistem Informasi Katalog Buku dan Skripsi Dengan Pencarian Menggunakan Algoritme Interpolation Search (Studi Kasus: Perpustakaan Universitas Muhammadiyah Bengkulu). *Pseudocode*, 6(2), 164-171. <https://doi.org/10.33369/pseudocode.6.2.164-171>
- Musalam Jessar, R., Khasanah, F. N., & Suraji, R. (n.d.). IMPLEMENTASI ALGORITMA SEQUENTIAL SEARCH DALAM SISTEM PRESENSI KARYAWAN ONLINE BERBASIS WEBSITE. In *JTID) INTEGRASI PUBLIKASI DIGITAL* (Vol. 1, Issue 1).
- Rahmanto, Y., Alfian, J., & Indra Borman, R. (n.d.). *Penerapan Algoritma Sequential Search pada Aplikasi Kamus Bahasa Ilmiah Tumbuhan 21*.



Rismayadi, A. A., & Jamaliah, L. (n.d.). *IMPLEMENTASI ALGORITMA SEQUENTIAL SEARCHING PADA APLIKASI E-OFFICE.*

Setyawati Wahyuni, W., Andryana, S., Rahman, B., Teknologi Komunikasi dan Informatika, F., Nasional Ps Minggu, U., Jakarta Selatan, K., & Khusus Ibukota Jakarta, D. (n.d.). *PENGUNAAN ALGORITMA SEQUENTIAL SEARCHING PADA APLIKASI PERPUSTAKAAN BERBASIS WEB.*

Suparman, P., & Azizah, A. L. (2022). PENERAPAN ALGORITMA SEQUENTIAL SEARCH PADA DATA NILAI SISWA DI SMK BINA NASIONAL INFORMATIKA MENGGUNAKAN PHP DAN MYSQL DI SMK BINA NASIONAL INFORMATIKA. *Jurnal Informatika SIMANTIK*, 7(1).
<https://www.simantik.fst-panca-sakti.ac.id/>