

IMPLEMENTASI FITUR PENCARIAN JUDUL BERITA PADA WEBSITE BOLA.COM MENGGUNAKAN METODE SEQUENTIAL SEARCH

Dendi Nugroho, Betha Nurina Sari, Iqbal Maulana

Teknik Informatika, Universitas Singaperbangsa Karawang
Jl. H.S. Ronggowaluyo Telukjambe Timur Karawang, Indonesia
1910631170173@student.unsika.ac.id

ABSTRAK

Portal berita digital seperti Bola.com menyajikan ribuan arsip berita, yang membuat volume datanya sangat besar. Volume data ini seringkali menyulitkan pengguna untuk menemukan judul berita spesifik yang mereka cari di masa lalu. Oleh karena itu, fungsionalitas pencarian yang efektif menjadi aspek krusial untuk meningkatkan pengalaman pengguna. Penelitian ini bertujuan untuk mengimplementasikan algoritma *Sequential Search* pada sebuah prototipe website untuk melakukan pencarian data judul berita yang bersumber dari website Bola.com. Masalah utama yang ingin dipecahkan adalah bagaimana merancang dan membangun sebuah fitur pencarian fungsional menggunakan algoritma pencarian dasar. Penelitian ini menggunakan pendekatan pengembangan perangkat lunak *SDLC (Software Development Life Cycle)* dengan model *Waterfall*, yang melibatkan tahap analisis kebutuhan, perancangan sistem, implementasi kode, dan pengujian fungsional. Bahasa pemrograman yang digunakan untuk membangun prototipe website ini adalah JavaScript dengan dukungan framework *Next.js 15*, *Tailwind CSS*, dan *TypeScript* untuk menciptakan antarmuka yang modern dan responsif. Hasil dari penelitian ini adalah sebuah prototipe aplikasi website fungsional yang mampu melakukan pencarian judul berita. Algoritma *Sequential Search* berhasil diimplementasikan untuk memproses kueri pencarian pengguna terhadap dataset judul berita. Penelitian ini juga menyajikan hasil pengujian fungsional (*black-box testing*) untuk memvalidasi fungsionalitas dan menyertakan data pengukuran waktu eksekusi pencarian sebagai bahan evaluasi kinerja algoritma.

Kata kunci : *Sequential Search, Model Waterfall, Pencarian Judul Berita, Bola.com, nextJs*

1. PENDAHULUAN

Perkembangan teknologi informasi dan digitalisasi telah mendorong peningkatan aktivitas masyarakat dalam mengakses informasi melalui internet. Survei Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) melaporkan penetrasi internet Indonesia periode 2022-2023 mencapai 78,19% atau sekitar 215,6 juta pengguna. Kondisi ini membuat website, termasuk portal berita, menjadi salah satu media utama masyarakat untuk memperoleh informasi secara cepat dan aktual [5]. Salah satu portal berita olahraga populer di Indonesia adalah Bola.com yang secara rutin mempublikasikan berita setiap hari, sehingga arsip judul berita yang tersimpan menjadi sangat besar. Pada portal berita dengan volume arsip tinggi, pengguna dapat mengalami kesulitan

Pada penelitian ini dirancang prototipe website pencarian judul berita yang bersumber dari Bola.com dengan pendekatan implementatif: data judul dan kategori diperoleh melalui proses web scraping,

disimpan dalam format JSON, kemudian ditampilkan pada antarmuka (UI) untuk dilakukan pencarian. Metode pencarian yang diterapkan adalah algoritma *Sequential Search*, karena bersifat sederhana, mudah dipahami, serta sesuai sebagai dasar pembelajaran implementasi algoritma pencarian pada aplikasi web.

Beberapa penelitian terdahulu telah menerapkan *Sequential Search* pada berbagai konteks, seperti aplikasi perpustakaan berbasis web [1], sistem pencarian naskah dinas [2], aplikasi layanan jasa

berbasis Android [3], dan game edukasi berbasis penyusunan kalimat [4]. Namun, kajian yang secara spesifik memfokuskan implementasi praktis *Sequential Search* pada studi kasus portal berita olahraga dengan arsip besar sebagai prototipe pembelajaran masih relatif terbatas. Oleh sebab itu, penelitian ini diharapkan dapat menjadi referensi implementasi algoritma pencarian dasar pada data tekstual judul berita. Dalam konteks portal berita dengan arsip besar, fungsionalitas pencarian yang efisien menjadi faktor krusial untuk mempermudah pengguna menemukan informasi yang diinginkan dengan cepat dan akurat. Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun sebuah prototipe website yang berfokus pada implementasi algoritma pencarian dasar untuk studi kasus pencarian data judul berita yang bersumber dari Bola.com.

Metode yang akan diterapkan adalah algoritma *Sequential Search*. Pemilihan algoritma *Sequential Search* didasarkan pada beberapa pertimbangan. Pertama, algoritma ini memiliki keunggulan dalam kesederhanaan (*simplicity*) dan mudah dipahami, menjadikannya fondasi yang sangat baik untuk mempelajari proses implementasi algoritma pencarian. Penelitian ini tidak bertujuan untuk menyaingi algoritma kompleks yang digunakan Bola.com secara *live*, melainkan berfokus pada proses implementasi praktis dari teori algoritma dasar ke dalam sebuah aplikasi *website* modern.

Beberapa penelitian sebelumnya telah mengeksplorasi penggunaan algoritma *Sequential*

Search. Sebagai contoh, penelitian sebelumnya telah mengeksplorasi penerapan Sequential Search pada aplikasi berbasis web, seperti yang terdapat pada Jurnal JIPI (2022) dengan judul "Penggunaan Algoritma Sequential Searching pada Aplikasi Perpustakaan Berbasis Web" [1]. Penelitian tersebut berfokus pada pencarian data item perpustakaan (buku). Meskipun memiliki kesamaan dalam penggunaan algoritma dan platform berbasis web, penelitian tersebut berbeda dengan studi kasus pada portal berita seperti Bola.com.

Penelitian terdahulu juga ada yang mengangkat topik implementasi Sequential Search pada sistem manajemen dokumen, seperti studi yang berjudul "Sistem Pencarian Naskah Dinas Dengan Algoritma Sequential Search" [2]. Penelitian ini berfokus pada pencarian data naskah dinas yang bersifat terstruktur dan formal. Hal ini berbeda dengan penelitian ini yang berfokus pada pencarian judul berita pada portal Bola.com, yang datanya cenderung lebih dinamis, tidak terstruktur, dan memiliki volume publikasi harian yang tinggi.

Selain implementasi pada dokumen formal, penerapan algoritma ini juga telah dikembangkan pada platform mobile, sebagaimana dibahas dalam jurnal berjudul "Implementasi Algoritma Sequential Searching pada Aplikasi Jasa Pekerja Properti Berbasis Android". Penelitian tersebut menunjukkan fleksibilitas algoritma Sequential Search yang mampu berjalan efektif di lingkungan sistem operasi Android untuk pencarian data profil pekerja. Hal ini menjadi pembandingan yang relevan, di mana studi tersebut berfokus pada pencarian data jasa (profil orang) pada platform mobile, sedangkan penelitian ini memfokuskan implementasi pada lingkungan berbasis website untuk pengolahan data tekstual berupa judul berita [3].

Algoritma ini juga terbukti adaptif dalam media pembelajaran interaktif, sebagaimana diulas dalam jurnal berjudul "Implementasi Algoritma Sequential Search Dalam Game Edukasi Penyusunan Kalimat Terstruktur". Penelitian ini memanfaatkan Sequential Search untuk memverifikasi urutan kata menjadi kalimat yang valid dalam sebuah permainan. Referensi ini memberikan wawasan penting bahwa algoritma ini mampu menangani logika pencocokan struktur kalimat (*sentence structure*), yang memiliki korelasi teknis dengan penelitian ini dalam hal pemrosesan data teks (string), meskipun diterapkan pada domain yang berbeda yaitu game edukasi [4].

Meskipun penelitian-penelitian tersebut telah membahas algoritma pencarian, masih sedikit yang secara spesifik berfokus pada implementasi praktis algoritma Sequential Search sebagai studi kasus pada portal berita olahraga dengan volume data yang besar seperti Bola.com, terutama dalam konteks pengembangan *prototype* untuk tujuan akademis dan pembelajaran. Dengan adanya penelitian ini, diharapkan dapat memberikan kontribusi berupa *prototype* fungsional dan menjadi referensi akademis

mengenai penerapan praktis algoritma Sequential Search dalam konteks pencarian data tekstual (judul berita) pada portal berita.

Namun, kajian yang secara spesifik memfokuskan implementasi praktis Sequential Search pada studi kasus portal berita olahraga dengan arsip besar sebagai prototipe pembelajaran masih relatif terbatas. Oleh sebab itu, penelitian ini diharapkan dapat menjadi referensi implementasi algoritma pencarian dasar pada data tekstual judul berita.

Berdasarkan latar belakang tersebut, penelitian ini memfokuskan dua permasalahan utama:

- bagaimana memanfaatkan data hasil scraping (judul dan kategori) dari Bola.com agar dapat ditampilkan kembali pada UI web secara terstruktur
- bagaimana mengimplementasikan algoritma Sequential Search untuk mencari judul berita pada dataset JSON

Tujuan penelitian ini adalah membangun prototipe website yang menampilkan data judul dan kategori berita hasil scraping dari Bola.com, mengimplementasikan algoritma Sequential Search pada proses pencarian judul berita berbasis dataset JSON. Untuk mencapai tujuan tersebut, penelitian dilakukan melalui tahapan:

- pengumpulan data judul dan kategori berita dari Bola.com menggunakan teknik web scraping;
- praproses data (pembersihan teks seperlunya) dan penyimpanan dalam format JSON;
- perancangan dan pembangunan UI web untuk menampilkan dataset;
- implementasi algoritma Sequential Search pada sisi aplikasi untuk melakukan pencarian berdasarkan kata kunci pada judul/kategori.

2. TINJAUAN PUSTAKA

2.1. Algoritma

Asal-usul kata 'algoritma' dapat ditelusuri melalui buku berjudul 'Aljabar wal Muqabala' karya Abu Ja'far Muhammad Ibnu Musa Al-Kwarizmi, seorang matematikawan Persia. Buku tersebut memuat berbagai definisi mengenai algoritma. Dalam konteks ini, algoritma didefinisikan sebagai serangkaian langkah-langkah yang tersusun secara sistematis dan logis untuk menyelesaikan suatu masalah. Dalam definisi ini, ada dua kata kunci yang penting, yaitu 'sistematis' dan 'logis' [5].

2.2. Algoritma Sequential Search

Algoritma Sequential Search beroperasi dengan membandingkan setiap elemen dalam *array* secara linear. Keunggulan metode ini terletak pada fleksibilitasnya yang dapat diterapkan baik pada kumpulan data terurut maupun acak [6].

Secara teknis, proses pencarian dimulai dari elemen pertama dan berlanjut hingga elemen terakhir. Apabila data target ditemukan, algoritma akan menghentikan proses dan mengembalikan nilai indeks lokasi data

tersebut. Sebaliknya, jika penelusuran mencapai akhir tanpa hasil, sistem akan mengindikasikan bahwa data tidak tersedia dalam struktur yang diuji.

Proses pencarian menggunakan metode sequential search dapat dianggap sederhana. Hal ini disebabkan oleh langkah-langkah yang terlibat dalam metode ini yang relatif *straightforward* dan mudah dipahami. Meskipun sederhana, metode sequential search memiliki kompleksitas waktu yang linear, yang berarti waktu yang dibutuhkan untuk pencarian akan meningkat secara proporsional dengan ukuran array.

Berikut adalah langkah-langkah yang terjadi dalam metode pencarian sequential search:

- a. Melakukan pembacaan array data.
- b. Menentukan data yang ingin dicari.
- c. Memulai dari data pertama dan berlanjut hingga data terakhir, data yang dicari dibandingkan dengan setiap data dalam array. Jika data yang dicari tidak ditemukan, maka perbandingan dilakukan dengan seluruh data atau elemen array hingga selesai. Namun, jika data yang dicari ditemukan, proses perbandingan akan dihentikan.[6]

2.3. Penelitian Sebelumnya

Implementasi algoritma *Sequential Search* telah banyak diteliti efektivitasnya dalam berbagai sistem informasi manajemen. [7] [8] membuktikan bahwa algoritma ini mampu mengatasi masalah manipulasi data dan kesulitan pencarian berkas fisik pada sistem administrasi kepegawaian, seperti presensi dan pengajuan cuti. Keunggulan algoritma ini dalam melakukan pencarian data secara linier juga dikonfirmasi oleh [9] pada pengelolaan data barang, di mana algoritma mampu memproses 800 sampel data dengan rata-rata waktu 0,2 detik. Selain itu, implementasi serupa pada instansi pemerintahan dan pendidikan dilakukan oleh Yuniar dan Amin (2021) untuk naskah dinas serta Nurwanto (2021) [10] untuk sistem perpustakaan, yang keduanya menunjukkan hasil validitas pencarian yang tinggi.

Meskipun penelitian-penelitian tersebut telah berhasil menerapkan Sequential Search, mayoritas masih berfokus pada pengelolaan data administratif internal (arsip, inventaris, dan kepegawaian) dengan teknologi berbasis PHP konvensional. Penelitian ini hadir untuk mengisi celah tersebut dengan mengimplementasikan Sequential Search pada objek data yang bersifat publik dan dinamis, yaitu judul berita olahraga pada portal Bola.com. Berbeda dengan penelitian sebelumnya yang menekankan pada fungsionalitas administratif, penelitian ini berfokus pada peningkatan pengalaman pengguna (*User Experience*) melalui pengukuran waktu eksekusi secara real-time menggunakan teknologi pengembangan web modern berbasis JavaScript, yaitu Next.js dan Tailwind CSS.

3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan studi kasus pada portal berita Bola.com dengan menerapkan algoritma Sequential Search. Untuk membangun prototipe sistem sebagai bukti konsep (proof of concept), digunakan metode pengembangan System Development Life Cycle (SDLC) dengan model Waterfall [11], [12].

Pemilihan model *Waterfall* didasarkan pada alurnya yang terstruktur dan sekuensial, di mana setiap tahapan—mulai dari analisis kebutuhan, desain, implementasi, hingga pengujian—harus diselesaikan secara berurutan. Pendekatan ini memastikan pengembangan fitur pencarian judul berita berjalan secara sistematis sesuai dengan ruang lingkup penelitian yang telah didefinisikan secara jelas.

3.1. Menentukan Rumusan Masalah dan Tujuan

Setelah melakukan pengamatan dan pengambilan data (*scraping*) dari website Bola.com, maka dilakukan perumusan masalah sesuai dengan kebutuhan pengelolaan data judul berita yang tersimpan dalam format JSON. Adapun tujuan akhir pada penelitian ini adalah mengimplementasikan fitur pencarian judul berita dengan penerapan algoritma Sequential Search yang dinilai lebih efektif dan efisien untuk diterapkan dalam menelusuri data berita pada sistem tersebut.

3.2. Pengumpulan Data

Pada tahap ini, pengumpulan data dilakukan dengan menggunakan teknik observasi dan web scraping pada situs berita olahraga Bola.com. Proses pengambilan data difokuskan untuk mengakuisisi atribut judul berita beserta tautan (url) yang relevan. Data yang berhasil dikumpulkan kemudian distrukturisasi dan disimpan dalam format file JSON (*JavaScript Object Notation*), yang selanjutnya akan digunakan sebagai basis data (*dataset*) utama untuk simulasi dan pengujian algoritma pencarian pada sistem.

3.3. Analisis dan Perancangan Sistem

Tahap selanjutnya adalah melakukan analisis terhadap kebutuhan fungsional sistem, khususnya dalam memproses kata kunci pencarian terhadap tumpukan data JSON. Perancangan sistem kemudian dimodelkan menggunakan diagram UML (*Unified Modeling Language*) untuk menggambarkan struktur perangkat lunak, serta Flowchart untuk mendetailkan logika kerja algoritma Sequential Search. Selain itu, dilakukan juga perancangan antarmuka (*User Interface*) yang mampu menampilkan hasil pencarian beserta indikator waktu eksekusi (*execution time*) guna mempermudah proses komparasi performa nantinya.

3.4. Implementasi

Pada tahap ini, desain sistem yang telah dirancang sebelumnya diterjemahkan ke dalam baris kode program (coding) agar dapat dijalankan oleh

komputer. Penulis membangun sistem menggunakan bahasa pemrograman Typescript dengan dukungan Json, Fokus utama pada fase ini adalah memastikan bahwa setiap modul dan fitur yang dibangun sesuai dengan spesifikasi kebutuhan pengguna serta arsitektur yang telah didefinisikan pada tahap perancangan.

3.5. Pengujian

Setelah proses implementasi selesai, dilakukan serangkaian pengujian untuk memverifikasi keandalan sistem. Pengujian ini bertujuan untuk menemukan potensi kesalahan (bug) dan memastikan seluruh fungsi berjalan sesuai logika yang diharapkan. Metode yang digunakan dalam penelitian ini adalah Black Box Testing, di mana pengujian difokuskan pada fungsionalitas input dan output sistem tanpa melihat struktur kode internal. Hasil pengujian akan didokumentasikan untuk menjamin sistem siap digunakan (deployment ready) [13].

3.6. Hasil dan Kesimpulan

Bagian ini memaparkan hasil akhir dari perangkat lunak yang telah berhasil dikembangkan dan diuji. Pembahasan mencakup evaluasi terhadap kinerja sistem dalam menyelesaikan permasalahan yang telah dirumuskan di awal penelitian. Selain itu, ditarik kesimpulan mengenai efektivitas sistem yang dibangun, serta saran untuk pengembangan lebih lanjut di masa mendatang.

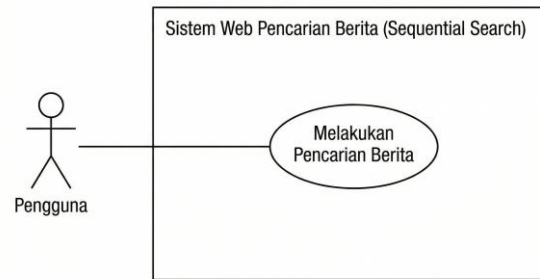
4. HASIL DAN PEMBAHASAN

4.1. Analisis Sistem

Analisis sistem dilakukan untuk membedah alur kerja aplikasi pencarian yang telah dibangun. Sistem ini bekerja dengan menerima input berupa kata kunci (keyword) dari pengguna, kemudian memprosesnya menggunakan algoritma pencarian terhadap sekumpulan data judul berita yang tersimpan dalam format JSON. Dalam prosesnya, sistem tidak melakukan permintaan (*request*) berulang ke server basis data eksternal, melainkan melakukan iterasi pada data lokal (*client-side*) yang telah di-load ke memori. Hal ini memungkinkan sistem untuk menampilkan hasil pencarian beserta durasi waktu eksekusi (*execution time*) secara *real-time* kepada pengguna.

4.2. Use Case Diagram

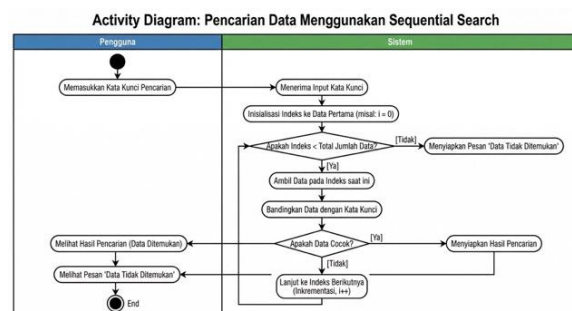
Perancangan interaksi antarmuka digambarkan melalui Use Case Diagram yang mendefinisikan aktor dan fungsionalitas utama sistem. Dalam sistem ini, terdapat satu aktor utama yaitu Pengguna (User). Aktor ini memiliki akses untuk melakukan aktivitas utama berupa "Melakukan Pencarian Berita". Saat aktor memasukkan kata kunci pada kolom pencarian, sistem secara otomatis akan merespons dengan menjalankan fungsi pencarian tanpa perlu melakukan login atau otentikasi yang kompleks, sesuai dengan tujuan efisiensi akses informasi.



Gambar 1. Use Case Diagram Pencarian Data Berita

4.3. Activity Diagram

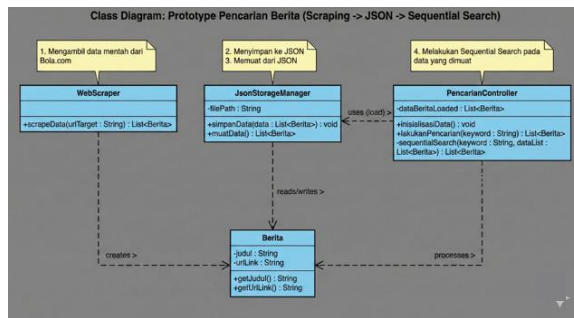
Alur logika operasional sistem dijelaskan secara rinci melalui Activity Diagram. Proses dimulai ketika pengguna mengakses halaman utama dan memasukkan kata kunci pada kolom pencarian. Setelah tombol cari ditekan atau pengguna menekan enter, sistem akan membaca file JSON yang berisi data judul berita. Selanjutnya, algoritma *Sequential Search* akan bekerja dengan menelusuri data satu per satu dari indeks awal hingga akhir. Jika data ditemukan (match), sistem akan menampung data tersebut ke dalam array hasil, dan jika proses iterasi selesai, sistem akan menampilkan daftar berita yang relevan beserta waktu tempuh pencarian ke layar pengguna.



Gambar 2. Use Case Diagram Pencarian Data Berita Menggunakan Sequential Search

4.4. Class Diagram

Struktur perangkat lunak dirancang menggunakan pola berorientasi objek yang terdiri dari empat kelas utama yang saling berinteraksi, sebagaimana ditunjukkan pada Gambar 4.X. Entitas dasar sistem direpresentasikan oleh kelas *Berita* yang memiliki atribut *judul* dan *urlLink* serta menjadi objek data utama yang dipertukarkan. Proses akuisisi data mentah ditangani oleh kelas *WebScraper* melalui metode *scrapeData()*, yang kemudian diserahkan kepada kelas *JsonStorageManager* untuk dikelola persistensinya dalam format JSON. Selanjutnya, kelas *PencarianController* bertindak sebagai pengendali logika utama yang memuat data ke memori dan mengeksekusi metode *sequentialSearch()* guna memfilter berita berdasarkan kata kunci dari pengguna.



Gambar 3. Class Diagram Pencarian Data Berita

4.5. Penerapan Sequential Search

Website Bola.com mengelola basis data judul berita yang luas dengan mengimplementasikan algoritma *Sequential Search* untuk memudahkan pencarian artikel atau berita tertentu. Algoritma ini bekerja dengan cara membandingkan kata kunci (*keyword*) yang dicari secara berurutan, mulai dari judul berita pertama hingga terakhir, sampai data yang diinginkan ditemukan.

Proses dimulai ketika pengguna menginput kata kunci pada kolom pencarian, kemudian sistem melakukan verifikasi ke database secara linear. Jika sistem menemukan judul berita yang cocok dengan

input pengguna, maka data tersebut akan ditampilkan pada halaman hasil pencarian. Namun, apabila hingga data terakhir tidak ditemukan kecocokan, sistem akan memberikan notifikasi bahwa berita tidak ditemukan dan proses pencarian akan dihentikan.

```
const dataBerita = [
  { id: 1, judul: "Hasil Pertandingan Liga Inggris Tadi Malam" },
  { id: 2, judul: "Jadwal Lengkap MotoGP 2025 di Mandalika" },
  { id: 3, judul: "Rumor Transfer: Bintang Muda Ini Diincar Real Madrid" },
  { id: 4, judul: "Analisis Taktik: Mengapa Timnas Indonesia Kalah?" },
  { id: 5, judul: "Bursa Transfer Liga 1: Siapa Saja Pemain Baru Persib?" },
  { id: 6, judul: "Hasil Liga Champions: Barcelona Tumbang di kandang" },
  { id: 7, judul: "Profil Pemain: Mengenal Bintang Baru Timnas Indonesia" }
];

// 2. Fungsi Inti Algoritma Sequential Search
function sequentialSearch(data, keyword) {
  const hasil = [];
  const keywordLower = keyword.toLowerCase();
  // Iterasi/pengecekan satu per satu dari awal sampai akhir
  for (let i = 0; i < data.length; i++) {
    const judulLower = data[i].judul.toLowerCase();
    if (judulLower.includes(keywordLower)) {
      hasil.push(data[i]);
    }
  }
  return hasil;
}

console.log(sequentialSearch(dataBerita, "liga"));
console.log(sequentialSearch(dataBerita, "indonesia"));
```

Gambar 4. Penerapan Algoritma Sequential Search

Implementasi *Sequential Search* pada sistem ini menggunakan JavaScript untuk memproses pencarian judul berita secara linear. Proses dimulai dengan menormalisasi kata kunci dan judul berita menjadi huruf kecil (*case-insensitive*). Selanjutnya, sistem melakukan iterasi menggunakan perulangan untuk memeriksa setiap elemen data satu per satu, mulai dari awal hingga akhir. Jika ditemukan kecocokan karakter antara kata kunci dan judul, data tersebut akan dikumpulkan ke dalam daftar hasil. Proses berakhir dengan menampilkan seluruh berita yang relevan atau

memberikan informasi jika data tidak ditemukan setelah seluruh database diperiksa.

```
// 2. ALGORITMA: Fungsi Sequential Search
function sequentialSearch(data, keyword) {
  let hasilDitemukan = [];

  // Iterasi dari awal sampai akhir
  for (let i = 0; i < data.length; i++) {
    const judulBerita = data[i].judul.toLowerCase();
    const kataKunciKecil = keyword.toLowerCase();

    // Cek kecocokan
    if (judulBerita.includes(kataKunciKecil)) {
      hasilDitemukan.push(data[i]);
    }
  }

  return hasilDitemukan;
}

// Output:
[SYSTEM] Memulai pencarian untuk kata kunci: "Timnas"
[DATABASE] Total data yang dipindai: 10 berita
[HASIL] Ditemukan 3 berita:
1. Prediksi Skor Timnas Indonesia vs Vietnam (2023-10-02)
2. Jadwal Timnas Indonesia di Piala Asia (2023-10-05)
3. Fiks! Justin Hubner Bisa Bela Timnas Indonesia (2023-10-07)
[PERFORMA] Waktu Eksekusi: 0.1650 milidetik (ms)
```

Gambar 5. Contoh Hasil Penerapan Algoritma Sequential Search Javascript

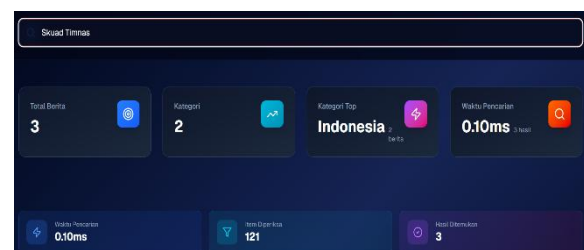
4.6. Interface

Tahap implementasi antarmuka merupakan visualisasi dari rancangan sistem yang telah dibuat. Pada bagian ini, akan dijelaskan mengenai tampilan halaman web yang dibangun untuk menampilkan data judul berita dari hasil scraping Bola.com serta fitur pencarian menggunakan metode *Sequential Search*. Antarmuka dirancang agar pengguna dapat dengan mudah berinteraksi dan mendapatkan informasi yang dicari.

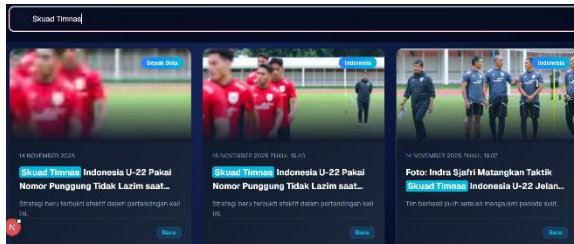
4.7. Interface Data Judul Berita

Sub-bab ini menampilkan antarmuka utama yang memuat seluruh data judul berita hasil scraping yang tersimpan dalam database JSON. Pada tampilan ini, sistem membaca array objek dari file JSON dan merender semua properti judul ke dalam bentuk daftar (list) tanpa melakukan filter apapun.

Halaman ini berfungsi sebagai indikator awal bahwa proses integrasi data antara file JSON dan antarmuka website telah berhasil berjalan. Pengguna dapat melihat daftar judul berita terkini yang diambil dari Bola.com secara berurutan sesuai indeks data. Tampilan antarmuka pencarian data judul berita tersebut dapat dilihat pada gambar 6 di bawah ini:



Gambar 6. Tampilan Antarmuka Pencarian Data

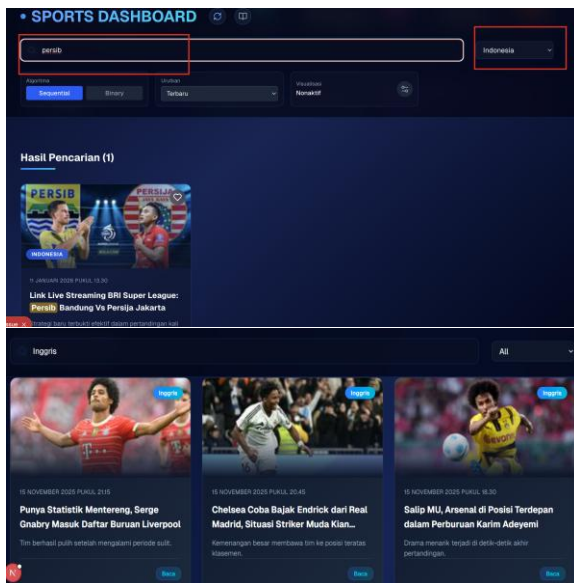


Gambar 7. Tampilan Pengujian Algoritma Sequential Search Data Judul Berita

4.8. Tampilan Data Judul Berita Berdasarkan Kategori

Selain menampilkan data secara keseluruhan, antarmuka sistem juga menyajikan informasi judul berita yang dikelompokkan berdasarkan atribut kategori yang terdapat pada struktur data JSON (misalnya: Liga Inggris, Bola Indonesia, atau Ragam).

Tampilan ini bertujuan untuk memvalidasi bahwa data scraping tidak hanya mengambil teks judul, tetapi juga berhasil mengambil metadata kategori berita. Pengguna dapat melihat label atau penanda khusus pada setiap judul yang menunjukkan klasifikasi berita tersebut. Tampilan data berita dengan label kategori dapat dilihat pada Gambar 8.



Gambar 8. Tampilan Pencarian Data Berdasarkan Kategori

Deskripsi Antarmuka: Gambar tersebut menampilkan fitur pencarian berita sepak bola dengan desain modern (tema gelap). Antarmuka terdiri dari bilah pencarian di bagian atas untuk memfilter kata kunci (seperti "Inggris") dan menu kategori (dropdown). Hasil pencarian disajikan dalam bentuk tiga kartu berita utama yang masing-masing dilengkapi dengan foto pemain, label kategori, tanggal unggah, judul berita, ringkasan singkat, dan tombol "Baca".

4.9. Pengujian

Tahap pengujian sistem merupakan fase krusial untuk memvalidasi akurasi dan efisiensi algoritma pencarian yang diterapkan. Pada penelitian ini, pengujian difokuskan pada fungsionalitas fitur pencarian judul berita yang datanya bersumber dari hasil scraping website Bola.com dan tersimpan dalam format JSON. Pengujian dilakukan untuk memastikan bahwa metode *Sequential Search* mampu menelusuri himpunan data JSON (*array of objects*) dan mengembalikan hasil yang relevan sesuai kata kunci yang diinputkan pengguna.

4.10. Proses Pengujian Algoritma

Proses pengujian algoritma Sequential Search dilaksanakan melalui tiga tahapan esensial untuk memastikan fungsi pencarian berjalan sesuai logika pemrograman. Tahapan tersebut mencakup:

- Input Kata Kunci (Keyword)** Pencarian Pengujian dimulai dengan memasukkan string atau kata kunci pada kolom pencarian yang tersedia di antarmuka website. Dalam tahap ini, input yang dimasukkan berupa teks yang relevan dengan judul berita yang tersimpan dalam dataset JSON, seperti nama klub (misalnya: "Persib", "MU"), nama pemain, atau topik liga (misalnya: "Liga 1", "Piala Dunia").
- Eksekusi Tombol Cari Data** Setelah kata kunci dimasukkan, tahap berikutnya adalah menekan tombol "Search" atau menekan tombol Enter untuk memicu fungsi pencarian. Pada tahap ini, sistem akan memanggil fungsi Sequential Search untuk membaca file JSON dan melakukan iterasi (perulangan) membandingkan kata kunci input dengan setiap properti judul berita yang ada di dalam array data.
- Penampikan Hasil dan Validasi Data** Hasil pengujian ditandai dengan munculnya daftar judul berita yang mengandung kata kunci yang dicari. Sistem akan menampilkan data judul beserta kategorinya dari file JSON ke halaman web. Selain itu, tahap ini juga dapat mencakup pengamatan terhadap waktu respons (execution time) yang dibutuhkan algoritma untuk menemukan dan menampilkan data tersebut sejak tombol ditekan.

4.11. Hasil Pembahasan

Sub-bab ini membahas hasil implementasi fitur pencarian judul berita pada prototipe website berbasis Next.js, serta mengevaluasi kinerjanya melalui pengujian fungsional dan pengukuran waktu eksekusi (execution time). Dataset yang digunakan berasal dari proses scraping judul berita Bola.com dan disimpan dalam format JSON (*array of objects*), kemudian dimuat ke memori untuk diproses secara client-side.

Berdasarkan pengujian fungsional (black-box testing) yang telah dilakukan, fitur pencarian mampu mengembalikan daftar judul berita yang relevan terhadap kata kunci yang dimasukkan pengguna. Sistem juga menampilkan informasi pendukung, seperti kategori berita, sehingga pengguna dapat

memverifikasi konteks hasil pencarian secara lebih cepat.

Dari sisi performa, karakteristik Sequential Search yang menelusuri elemen secara berurutan menyebabkan jumlah perbandingan (comparisons) bertambah seiring ukuran data (n). Dengan demikian, waktu pencarian akan cenderung meningkat secara linear ketika jumlah judul berita yang disimpan bertambah. Hal ini menjelaskan mengapa pengukuran execution time menjadi bagian penting untuk menunjukkan konsekuensi praktis dari kompleksitas waktu algoritma pada data yang semakin besar.

Meskipun demikian, pada konteks prototipe dan dataset berukuran kecil-menengah, Sequential Search tetap relevan karena implementasinya sederhana, mudah diuji, dan cukup responsif untuk kebutuhan pencarian cepat. Temuan ini selaras dengan tujuan penelitian, yaitu mengimplementasikan algoritma dasar secara praktis pada sistem pencarian judul berita.

4.12. Pengujian Kompleksitas Waktu Algoritma Sequential Search

Pengujian kompleksitas waktu dilakukan untuk mengkaji hubungan antara ukuran data (n) dan waktu eksekusi pencarian menggunakan algoritma Sequential Search. Secara teori, Sequential Search melakukan pencocokan elemen satu per satu dari indeks awal hingga data ditemukan atau sampai elemen terakhir. Konsekuensinya, kompleksitas waktu bersifat linear terhadap n .

Pada kasus terbaik (best case), data yang dicari berada pada elemen pertama sehingga proses berhenti

setelah satu kali perbandingan ($O(1)$). Pada kasus terburuk (worst case), data tidak ditemukan atau berada pada elemen terakhir sehingga sistem melakukan perbandingan terhadap seluruh elemen ($O(n)$). Pada kasus rata-rata (average case) dengan asumsi posisi target merata, jumlah perbandingan berada di sekitar $n/2$ sehingga tetap $O(n)$ [14].

Dalam implementasi berbasis JavaScript/Next.js, waktu eksekusi diukur menggunakan fungsi `performance.now()` agar memperoleh timestamp beresolusi tinggi (milidetik). Pengukuran dilakukan dengan mencatat waktu sebelum fungsi pencarian dijalankan (t_0) dan setelah fungsi selesai (t_1), kemudian waktu eksekusi dihitung dengan rumus

$$\Delta t = t_1 - t_0 \quad [15], [16].$$

```
Contoh implementasi pengukuran waktu (JavaScript):
const t0 = performance.now();
const hasil = sequentialSearch(data, keyword);
const t1 = performance.now();
const waktuMs = t1 - t0;
```

Gambar 9. pengukuran durasi eksekusi (runtime) Algoritma

Untuk meminimalkan noise pengukuran, setiap skenario pencarian diulang beberapa kali (misalnya 10-30 kali), lalu diambil nilai rata-rata. Pengujian dilakukan pada perangkat dan kondisi yang sama (browser, spesifikasi perangkat, serta beban aplikasi) agar hasil lebih konsisten.

Tabel 1. Tabel Analisis Kompleksitas Waktu Algoritma Sequential Search

Skenario	Ukuran data (n)	Kata kunci	Posisi target	Jumlah perbandingan	Waktu eksekusi (ms)
Best case	(isi n)	(isi kata kunci)	Awal	1	(isi ms)
Average case	(isi n)	(isi kata kunci)	Tengah	$\approx n/2$	(isi ms)
Worst case	(isi n)	(isi kata kunci)	Akhir	n	(isi ms)
Worst case	(isi n lebih besar)	(isi kata kunci)	Tidak ditemukan	n	(isi ms)
Worst case	(isi n lebih besar)	(isi kata kunci)	Tidak ditemukan	n	(isi ms)

Berdasarkan teori dan hasil pengukuran, tren yang diharapkan adalah waktu eksekusi meningkat seiring bertambahnya n (linear). Jika ukuran dataset dilipatgandakan, jumlah perbandingan dan waktu pencarian cenderung meningkat secara proporsional. Oleh karena itu, untuk skala data yang sangat besar, optimasi dapat dipertimbangkan, misalnya menggunakan struktur data berindeks atau metode pencarian yang lebih efisien.

5. KESIMPULAN DAN SARAN

Kesimpulan dari penelitian ini adalah sebagai berikut: Pertama, implementasi website berhasil meningkatkan efektivitas Website Bola.com sebagai

sarana informasi dan promosi. Kedua, penggunaan algoritma *sequential search* pada website tersebut berhasil diimplementasikan sebagai metode pencarian data siswa, donatur, dan kegiatan keagamaan yang efektif dan sederhana. Ketiga, hasil pengujian dan evaluasi efektivitas website menunjukkan peningkatan signifikan dalam pencarian data dan penggunaan website secara keseluruhan. Terakhir, implementasi algoritma *sequential search* telah diuji dan dievaluasi dalam konteks website bola.com, menunjukkan kecocokan algoritma ini untuk website dengan jumlah data yang relatif kecil serta efektivitasnya dalam mencari informasi yang diperlukan oleh pengguna. Saran untuk penelitian selanjutnya mencakup evaluasi

dan optimasi kecepatan pencarian data dengan mempertimbangkan teknik optimasi, seperti penggunaan indeks atau struktur data yang lebih efisien. Selain itu, dianjurkan untuk memperluas cakupan pencarian pada website Bola.com dengan menyertakan fitur pencarian konten berita atau artikel.

DAFTAR PUSTAKA

- [1] W. S. Wahyuni, S. Andryana, and B. Rahman, "Penggunaan Algoritma Sequential Searching pada Aplikasi Perpustakaan Berbasis Web," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 7, no. 2, pp. 294–302, Jun. 2022, doi: 10.29100/jipi.v7i2.2646.
- [2] W. L. Yuniar and F. Amin, "Sistem Pencarian Naskah Dinas dengan Algoritma Sequential Search," *Jurnal Manajemen Informatika dan Sistem Informasi*, vol. 4, no. 2, pp. 92–100, Oct. 2021, doi: 10.36595/misi.v4i2.359.
- [3] Y. Sonita and M. Sari, "Implementasi Algoritma Sequential Searching untuk Pencarian Nomor Surat pada Sistem Arsip Elektronik," *Pseudocode*, vol. 5, no. 1, pp. 1–9, Feb. 2018, doi: 10.33369/pseudocode.5.1.1-9.
- [4] S. Alridho, R. Mulyana, D. Saputra, and A. Arianto, "Implementasi Algoritma Sequential Search dalam Game Edukasi Penyusunan Kalimat Terstruktur," in *Seminar Nasional Sains dan Teknologi Informasi dan Komunikasi (SNASTIKOM)*, vol. 2, no. 1, pp. 90–97, 2023.
- [5] Encyclopaedia Britannica, "Algorithm." Accessed: Jan. 11, 2026. [Online]. Available: <https://www.britannica.com/science/algorithm>
- [6] E. H. S. Atmaja and E. H. Parnadi, "Aplikasi Penjadwalan Dokter Menggunakan Metode Constraint Satisfaction Problem dengan Algoritma Sequential Search dan Forward Checking," in *Seminar Nasional Teknologi Informasi dan Multimedia*, pp. 109–114, Feb. 2014.
- [7] K. A. Widodo, S. A. Wibowo, and N. Vendyansyah, "Penerapan Sequential Search untuk Pengelolaan Data Barang," *Antivirus: Jurnal Ilmiah Teknik Informatika*, vol. 15, no. 1, pp. 86–97, 2021, doi: 10.35457/antivirus.v15i1.1385.
- [8] D. Markuci and C. Prianto, "Analisis Perbandingan Penggunaan Algoritma Sequential Search dan Binary Search pada Aplikasi Surat Perjalanan Dinas," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 6, no. 1, pp. 110–119, Mar. 2022, doi: 10.36040/jati.v6i1.4569.
- [9] R. M. Jessar, F. Khasanah, and Suraji, "Implementasi Algoritma Sequential Search dalam Pencarian Data Berbasis Website," *Jurnal Teknologi Informasi Digital*, vol. 1, no. 1, pp. 12–21, 2025.
- [10] W. T. R. Ridwan, S. Sunardi, and S. Alam, "Implementasi Algoritma Sequential Search dalam Pencarian Data Cuti pada Aplikasi Cuti Pegawai," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 5, no. 1, pp. 198–206, Dec. 2024, doi: 10.57152/malcom.v5i1.1735.
- [11] A. A. Wahid, "Analisis Metode Waterfall untuk Pengembangan Sistem Informasi," *Jurnal Ilmu-Ilmu Informatika dan Manajemen STMIK*, pp. 1–5, Nov. 2020.
- [12] I. Sommerville, *Software Engineering*, 10th ed. Pearson, 2015.
- [13] W. Wibisono and F. Baskoro, "Pengujian Perangkat Lunak dengan Menggunakan Model Behaviour UML," *JUTI*, vol. 1, no. 1, pp. 43–50, Jul. 2002, doi: 10.12962/j24068535.v1i1.a95.
- [14] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 4th ed. Cambridge, MA, USA: MIT Press, 2022.
- [15] MDN Web Docs, "Performance: now() method." Accessed: Jan. 11, 2026. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/API/Performance/now>
- [16] W3C, "High Resolution Time," W3C Recommendation, Nov. 7, 2024. Accessed: Jan. 11, 2026. [Online]. Available: <https://www.w3.org/TR/hr-time-3/>