e-ISSN: 3064-3082
DOI Article: 10.46306/jion.v2i1.210

JOURNAL INTECH AND EDUCATION
Jurnal Pendidikan Teknologi Informasi
Universitas Bina Bangsa
http://ejournal.lppmbinabangsa.ac.id/index.php/jion

IMPLEMENTATION OF A SIMPLE WEB-BASED RECOMMENDATION E-BOOK SYSTEM USING CONTENT-BASED FILTERING

Xavier Fabiano Antonius Mamangkey [1], Bertrand Emmanuel Supoyo [2], Riky Martien Mengko [3], I Made Hendy Wijaya [4], Victor Tarigan[5], Ade Yusupa[6]
[1,2,3,4,5,6]Universitas Sam Ratulangi
E-mail : xaviermamangkey026@student.unsrat.ac.id[1], bertrandsupoyo026@student.unsrat.ac.id,[2] rikymengko026@student.unsrat.ac.id,[3] iwijaya0261@student.unsrat.ac.id,[4] ade@unsrat.ac.id, [5] victortarigan@unsrat.ac.id, [6]

Abstract

The rapid expansion of digital libraries necessitates efficient recommendation systems to help users discover relevant e-books. This study presents the implementation of a simple web-based e-book recommendation system using content-based filtering, developed with the Next.js framework to enhance web loading speed and performance. The system analyzes book metadata and textual content to generate personalized recommendations based on user preferences. Core functionalities include a responsive user interface, book similarity calculations using and cosine similarity, and real-time dynamic suggestions. By leveraging Next.js, the system benefits from server-side rendering (SSR) and static site generation (SSG), ensuring faster page loads and improved user experience. Experimental results indicate that content-based filtering effectively suggests relevant e-books but faces challenges such as the cold-start problem. Future work may integrate hybrid filtering techniques to improve recommendation accuracy and user engagement.

Keywords: *E-book recommendation, content-based filtering, Next.js, web performance, cosine similarity.*

Article history:
Received: 20-03-2025
Accepted: 30-04-2025
Published: 30-05-2025

Abstrak

Pesatnya perkembangan perpustakaan digital menuntut adanya sistem rekomendasi yang efisien untuk membantu pengguna menemukan e-book yang relevan. Penelitian ini membahas implementasi sistem rekomendasi e-book berbasis web menggunakan content-based filtering, dikembangkan dengan framework Next.js untuk meningkatkan kecepatan pemuatan halaman dan performa web. Sistem ini menganalisis metadata dan konten teks e-book untuk menghasilkan rekomendasi yang dipersonalisasi berdasarkan preferensi pengguna. Fitur utama yang diterapkan mencakup antarmuka pengguna yang responsif, perhitungan kesamaan e-book menggunakan dan cosine similarity, serta rekomendasi yang

e-ISSN: 3064-3082
DOI Article: 10.46306/jion.v2i1.210

diperbarui secara dinamis. Dengan memanfaatkan Next.js, sistem ini mengadopsi server-side rendering (SSR) dan static site generation (SSG) guna memastikan waktu muat halaman yang lebih cepat dan pengalaman pengguna yang lebih optimal. Hasil pengujian menunjukkan bahwa content-based filtering mampu memberikan rekomendasi e-book yang relevan, meskipun masih menghadapi tantangan seperti masalah cold-start. Pengembangan lebih lanjut dapat mengintegrasikan teknik hybrid filtering untuk meningkatkan akurasi rekomendasi dan keterlibatan pengguna.
Kata kunci: *Rekomendasi e-book, content-based filtering, Next.js, performa web, cosine similarity.*

e-ISSN: 3064-3082
DOI Article: 10.46306/jion.v2i1.210

PENDAHULUAN

Seiring dengan kemajuan teknologi digital, jumlah e-book yang tersedia secara daring terus bertambah pesat. Kini, perpustakaan digital menawarkan ribuan hingga jutaan buku dalam berbagai kategori dan genre, memberikan akses luas bagi pembaca tanpa batasan lokasi atau waktu. Namun, di balik kemudahan ini, muncul tantangan baru: bagaimana pengguna dapat menemukan bacaan yang sesuai dengan minat mereka tanpa harus menelusuri koleksi yang begitu besar secara manual?

Di sinilah peran sistem rekomendasi e-book menjadi sangat penting. Dengan memanfaatkan kecerdasan buatan dan teknik pemfilteran data, sistem ini dirancang untuk membantu pengguna menemukan bacaan yang relevan berdasarkan preferensi mereka. Salah satu metode yang umum digunakan adalah content-based filtering, yang bekerja dengan menganalisis berbagai karakteristik e-book—mulai dari judul, deskripsi, hingga isi buku—untuk menyusun rekomendasi yang lebih personal.[1]

Dibandingkan dengan collaborative filtering, metode content-based filtering menawarkan keunggulan tersendiri, terutama dalam situasi di mana jumlah pengguna masih terbatas atau data preferensi belum cukup banyak untuk dianalisis. Karena tidak bergantung pada data pengguna lain, pendekatan ini mampu memberikan rekomendasi yang lebih independen dan spesifik. Namun, tantangan tetap ada, salah satunya adalah cold-start problem, di mana pengguna baru atau e-book yang baru masuk ke dalam sistem belum memiliki cukup data untuk dianalisis secara optimal.[2], [3]

Selain akurasi rekomendasi, performa sistem juga menjadi faktor kunci dalam memberikan pengalaman terbaik bagi pengguna. Kecepatan dan responsivitas platform dapat menentukan seberapa nyaman seseorang dalam mencari dan menemukan e-book yang diinginkan. Oleh karena itu, penelitian ini mengandalkan Next.js, sebuah framework yang menawarkan berbagai fitur unggulan seperti Server-Side Rendering (SSR) dan Static Site Generation (SSG), yang secara signifikan mempercepat waktu pemuatan halaman dibandingkan dengan metode berbasis client-side rendering.[4]

Dengan menggabungkan content-based filtering dan teknologi Next.js, penelitian ini bertujuan untuk mengembangkan sistem rekomendasi e-book berbasis web yang tidak hanya akurat dalam memberikan rekomendasi, tetapi juga cepat dan responsif dalam menyajikan informasi. Selain itu, sistem ini juga dirancang untuk mengatasi berbagai tantangan, termasuk cold-start problem, sekaligus menghadirkan pengalaman pengguna yang lebih dinamis dan interaktif.[5]

METODE PENELITIAN

1. Pengumpulan dan Pengolahan Data

Penelitian ini bertujuan untuk mengembangkan dan mengevaluasi sistem rekomendasi e-book berbasis web dengan pendekatan Content-Based Filtering menggunakan Next.js. Metodologi yang digunakan dalam penelitian ini terdiri dari beberapa tahapan utama, yaitu pengumpulan data, preprocessing data, implementasi sistem rekomendasi, serta evaluasi kinerja sistem.[6]

1.1 Sumber Data

e-ISSN: 3064-3082
DOI Article: 10.46306/jion.v2i1.210

Dataset yang digunakan dalam sistem ini terdiri dari kumpulan e-book yang berasal dari berbagai sumber, antara lain:

- **Perpustakaan Digital Terbuka:** Seperti Project Gutenberg, Open Library, atau sumber e-book berlisensi terbuka lainnya.
- **Dataset Publik:** Dataset yang tersedia di platform seperti Kaggle atau repositori akademik yang menyediakan metadata buku.
- **Koleksi E-Book Kurasi Sendiri:** E-book yang dikumpulkan secara manual dari berbagai penerbit atau penulis yang mengizinkan distribusi kontennya.

E-book yang dikumpulkan disimpan dalam database untuk diproses lebih lanjut dalam sistem rekomendasi.[7]

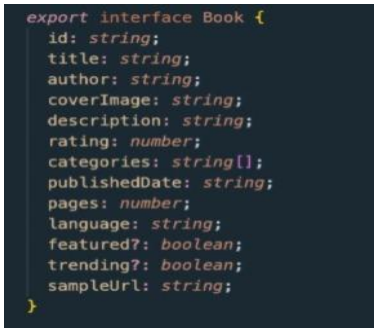
1.2 Struktur Data

Setiap e-book dalam sistem rekomendasi ini direpresentasikan melalui metadata yang disimpan dalam format terstruktur. Metadata ini mencakup informasi penting yang digunakan untuk proses pemfilteran berbasis konten (content-based filtering). Berikut adalah atribut utama yang digunakan:

- **Judul**: Nama atau judul e-book yang berfungsi sebagai identifikasi utama.
- **Penulis**: Nama pengarang atau penulis e-book.
- **Deskripsi**: Ringkasan atau sinopsis e-book yang memberikan gambaran umum tentang isi buku.
- **Kategori/Genre**: Klasifikasi e-book berdasarkan genre, seperti fiksi, nonfiksi, biografi, misteri, fantasi , dan fiksi ilmiah
- **Teks Isi Buku (Opsional)**: Jika tersedia, teks lengkap e-book dapat digunakan untuk meningkatkan akurasi rekomendasi berbasis konten.

HASIL DAN PEMBAHASAN

Pemaparan penggunaan Struktur data dalam buku :



Gambar 1. Typescript Interface

Penggunaan metadata yang kaya dan terstruktur merupakan komponen kunci dalam sistem rekomendasi berbasis konten. Metadata yang lengkap memungkinkan sistem untuk menangkap lebih banyak aspek dari e-book, seperti tema, genre, dan konten, sehingga meningkatkan akurasi dan relevansi rekomendasi yang dihasilkan.[8]

e-ISSN: 3064-3082
DOI Article: 10.46306/jion.v2i1.210

1.3 Pengolahan Data

Sebelum data digunakan dalam sistem rekomendasi, beberapa langkah **preprocessing** dilakukan untuk meningkatkan kualitas data. Langkah-langkah ini meliputi penyusunan struktur data yang terorganisir dan pemberian data dummy [9] untuk memastikan sistem dapat berfungsi dengan baik.

Penggunaan Struktur Data

Struktur data dirancang dengan membuat **interface** untuk buku (Book) dan kategori (Category), serta menambahkan properti tambahan seperti featured dan trending pada buku. Hal ini bertujuan untuk memudahkan proses filtering dan meningkatkan fleksibilitas dalam pengelolaan data.

Interface Book: Mendefinisikan atribut-atribut penting seperti id, title, author, rating, categories, dan publishedDate. Properti tambahan seperti featured dan trending menggunakan tipe data boolean opsional, yang memungkinkan fleksibilitas dalam pengelolaan data.

Gambar 2. Typescript Category.

Interface Category: Menyediakan informasi tentang kategori buku, seperti id, name, description, dan image.

Gambar 3. Typescript Interface Category

Penambahan fitur-fitur tersebut bertujuan untuk memastikan data terstruktur dengan baik, mempermudah proses debugging jika terjadi kesalahan, dan mendukung

implementasi fitur-fitur tambahan seperti rekomendasi buku yang sedang tren atau dipromosikan.

Pemberian Data Dummy

Dalam penelitian ini, data dummy digunakan untuk menguji dan memvalidasi sistem. Data dummy mencakup daftar kategori dan buku yang dirancang untuk merepresentasikan berbagai genre dan contoh buku nyata. Selain itu, URL gambar berkualitas tinggi juga disertakan untuk memastikan tampilan visual yang baik.

- **Books dan Categories**: Data dummy berisi contoh buku dan kategori yang realistis, dengan judul buku asli, penulis terkenal, dan deskripsi yang bermakna. Hal ini memastikan bahwa sistem dapat bekerja dengan data yang mendekati kondisi nyata.
- **URL Gambar**: Gambar buku diambil dari layanan terpercaya seperti Unsplash dan ImageKit, yang memastikan kualitas gambar yang tinggi dan konsisten.

Gambar 4. Typescript Category dengan URL Gambar.

Dengan menggunakan data dummy yang terstruktur dan realistis, sistem rekomendasi dapat diuji secara lebih efektif, memastikan bahwa semua fitur berfungsi sebagaimana mestinya sebelum diimplementasikan pada data nyata.

2. Preprocessing Data

Sebelum digunakan dalam sistem rekomendasi, data e-book mengalami serangkaian proses **preprocessing** untuk meningkatkan akurasi analisis teks dan efisiensi sistem dalam menemukan kesamaan antar e-book. Proses ini bertujuan untuk membersihkan, menyederhanakan, dan menormalisasi data teks sehingga dapat digunakan dalam perhitungan rekomendasi.

2.1 Hasil Preprocessing

Setelah seluruh proses preprocessing dilakukan, setiap e-book direpresentasikan dalam bentuk vektor numerik berdasarkan bobot kata-kata yang ada di dalamnya. Vektor ini akan digunakan dalam perhitungan kemiripan e-book dengan teknik cosine similarity, yang akan membantu sistem rekomendasi menentukan buku yang paling relevan untuk pengguna.

3. Implementasi Content-Based Filtering

e-ISSN: 3064-3082
DOI Article: 10.46306/jion.v2i1.210

Sistem rekomendasi dalam penelitian ini menggunakan pendekatan **Content-Based Filtering** dengan **Cosine Similarity** untuk menentukan tingkat kemiripan antar e-book. Teknik ini memungkinkan sistem untuk memberikan rekomendasi berdasarkan kesamaan fitur teks antara e-book yang telah dipilih atau dibaca oleh pengguna dengan e-book lainnya dalam dataset.[10]

3.1 Pengukuran Kemiripan Menggunakan Cosine Similarity

berikutnya adalah menghitung kemiripan antar e-book menggunakan cosine similarity.

Konsep Cosine Similarity:

Cosine similarity mengukur sudut antara dua vektor dalam ruang multidimensi. Nilainya berkisar antara 0 (tidak mirip) hingga 1 (sangat mirip).

$$\text{Cosine Similarity} = \frac{A \cdot B}{||A|| \times ||B||}$$

Di mana:

- A dan B adalah vektor TF-IDF dari dua e-book.
- A cdot B adalah hasil perkalian dot product kedua vektor.
- |A| dan |B| adalah panjang (magnitude) dari masing-masing vektor.[11], [12]

Dalam implementasi proyek, metode ini digunakan untuk menemukan buku yang memiliki kemiripan tinggi berdasarkan deskripsi buku dan ditampilkan dalam bentuk persenan. Contohnya dapat ditemukan dalam fungsi:

- mengubah deskripsi menjadi vektor data

```
// Fungsi untuk mengubah deskripsi menjadi vektor frekuensi kata
function textToVector(text: string) {
  if (!text) return {};
  const words = text.toLowerCase().match(/\b\w+\b/g) || []
  const freqMap: Record<string, number> = {}

  words.forEach((word) => {
    freqMap[word] = (freqMap[word] || 0) + 1
  })

  return freqMap
}
```

Gambar 5. Fungsi mengubah deskripsi menjadi vektor frekuensi data.

- menghitung cosine similarity dari dua vektor dan cosine similarity buku berdasarkan deskripsi

```
// Fungsi untuk menghitung cosine similarity antara dua vektor teks
function calculateSimilarity(textA: string, textB: string) {
  const vectorA = textToVector(textA)
  const vectorB = textToVector(textB)

  const uniqueWords = new Set([...Object.keys(vectorA), ...Object.keys(vectorB)])

  const dotProduct = Array.from(uniqueWords).reduce((sum, word) => {
    return sum + (vectorA[word] || 0) * (vectorB[word] || 0);
  }, 0);

  const magnitudeA = Math.sqrt(Object.values(vectorA).reduce((sum, val) => sum + val ** 2, 0));
  const magnitudeB = Math.sqrt(Object.values(vectorB).reduce((sum, val) => sum + val ** 2, 0));

  return magnitudeA && magnitudeB ? (dotProduct / (magnitudeA * magnitudeB)) : 0;
}

export default function BookPage({ params }: { params: { bookId: string } }) {
  const book = getBookById(params.bookId)
  if (!book) notFound()

  // Hitung Cosine Similarity untuk setiap buku lain berdasarkan deskripsi
  const similarityScores = books.map((b) => ({
    book: b,
    similarity: calculateSimilarity(book.description, b.description)
  }))

  // Urutkan berdasarkan kesamaan dan ambil 4 buku terkait
  const relatedBooks = similarityScores
    .filter((b) => b.book.id !== book.id) // Hindari buku itu sendiri
    .sort((a, b) => b.similarity - a.similarity)
    .slice(0, 4)
```

Gambar 6. Source code rekomendasi kategori.

e-ISSN: 3064-3082
DOI Article: 10.46306/jion.v2i1.210

Kode ini memastikan bahwa buku yang direkomendasikan berasal dari kategori yang sama dengan buku yang sedang dilihat, sehingga relevansi tetap terjaga.

3.2 Penyusunan Rekomendasi

Setelah memperoleh skor kemiripan antar e-book, sistem dapat merekomendasikan e-book yang memiliki nilai kemiripan tertinggi dengan e-book yang dipilih atau telah dibaca oleh pengguna.

Langkah-langkah rekomendas seperti:

1. Menentukan e-book referensi: Misalnya, e-book yang terakhir dibaca oleh pengguna.
2.Mencari e-book serupa: Mengurutkan e-book berdasarkan skor **cosine similarity** tertinggi.
3.Menyusun daftar rekomendasi: Menampilkan e-book yang paling relevan.[13]

Dalam kode proyek, proses ini diwujudkan dalam bagian:

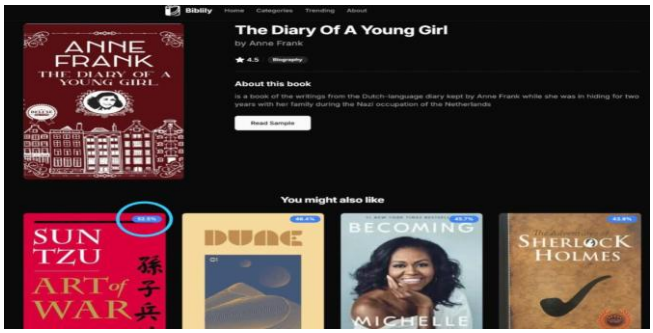
```
export function getFeaturedBooks(): Book[] {
  return books.filter(book => book.featured);
}

export function getTrendingBooks(): Book[] {
  return books.filter(book => book.trending);
}

export function getBooksByCategory(categoryId: string): Book[] {
  return books.filter(book => book.categories.includes(categoryId));
}

export function getBookById(id: string): Book | undefined {
  return books.find(book => book.id === id);
}

export function getAllCategories(): Category[] {
  return categories;
}

export function getCategoryById(id: string): Category | undefined {
  return categories.find(category => category.id === id);
}
```

Gambar 7. Fungsi rekomendasi.

Fungsi ini memastikan bahwa rekomendasi terdiri dari buku dengan kategori serupa, yang kemungkinan besar memiliki kemiripan dalam fitur deskripsi.

3.3 Output Rekomendasi

Saat pengguna membuka halaman detail suatu buku, sistem akan secara otomatis menampilkan rekomendasi buku serupa berdasarkan kategori utama buku tersebut. Contoh tampilan rekomendasi diimplementasikan dengan kode berikut:



Gambar 8. Tampilan rekomendasi buku.

e-ISSN: 3064-3082
DOI Article: 10.46306/jion.v2i1.210

4. Pengembangan Sistem Menggunakan Next.js

Untuk mengembangkan sistem rekomendasi e-book berbasis web, digunakan Next.js, sebuah framework React yang dirancang untuk meningkatkan performa aplikasi dengan berbagai fitur seperti Server-Side Rendering (SSR), Static Site Generation (SSG), API Routes, dan Client-Side Interaction. Pemilihan Next.js bertujuan untuk memastikan pengalaman pengguna yang lebih cepat, responsif, dan efisien.

4.1 Arsitektur Sistem

Sistem rekomendasi e-book berbasis Next.js terdiri dari beberapa komponen utama:

1. Frontend (UI/UX): Dibangun menggunakan React dan Tailwind CSS untuk tampilan yang modern dan responsif.
2. Backend API: Memanfaatkan API Routes Next.js untuk menghubungkan frontend dengan basis data dan sistem rekomendasi.
3. Database: Menggunakan API, Data struktur yang dimasukan serta beberapa Author yang mengijinkan untuk memaparkan bukunya untuk memfasilitasi jenis-jenis buku
4. Sistem Rekomendasi: Implementasi **Content-Based Filtering**, dalam ini penggunaan software **ManggoDB** untuk memberikan rekomendasi buku secara real-time.[14], [15]

4.2 Implementasi Fitur Utama

4.2.1 Server-Side Rendering (SSR) untuk Rekomendasi

Untuk meningkatkan kecepatan loading halaman, Server-Side Rendering (SSR) digunakan dalam halaman rekomendasi. Dengan SSR, rekomendasi e-book dihitung di server sebelum halaman dikirim ke pengguna, sehingga mengurangi waktu tunggu di sisi klien.[16]

Keunggulan SSR:

- Rekomendasi langsung tersedia saat halaman dimuat.
- Mengurangi beban di sisi klien, meningkatkan performa untuk perangkat dengan spesifikasi rendah.

4.2.2 Static Site Generation (SSG) untuk Kategori Populer

Untuk meningkatkan efisiensi akses, halaman kategori e-book populer dihasilkan menggunakan Static Site Generation (SSG). Halaman ini dibuat saat proses build dan dapat di-cache oleh CDN, sehingga lebih cepat diakses.[5]

Keunggulan SSG:

- Mempercepat waktu akses halaman populer.
- Mengurangi beban server dengan menyajikan konten statis.
- Konten tetap diperbarui secara berkala dengan fitur **revalidate**.

4.2.3 API Routes untuk Integrasi Sistem Rekomendasi

Untuk menghubungkan sistem rekomendasi e-book dengan basis data, digunakan API Routes dalam Next.js. API ini memungkinkan frontend mengakses data rekomendasi secara dinamis.

Keunggulan API Routes:

- Memudahkan komunikasi antara frontend dan backend.
- Mendukung pembaruan rekomendasi secara real-time.

4.2.4 Client-Side Interaction untuk Personalisasi Rekomendasi

e-ISSN: 3064-3082
DOI Article: 10.46306/jion.v2i1.210

Pengguna dapat memberikan rating dan menambahkan buku ke daftar favorit, yang akan digunakan untuk memperbarui rekomendasi mereka secara personal.

Keunggulan Client-Side Interaction:

- Pengguna dapat menyesuaikan rekomendasi sesuai preferensi.
- Interaksi langsung tanpa perlu reload halaman.

4.3 Hasil dan Keunggulan Pengembangan dengan Next.js

a. **Kecepatan dan Performa Optimal**

- SSR mempercepat waktu muat halaman rekomendasi dengan memproses data di server.
- SSG meningkatkan efisiensi akses ke kategori buku populer dengan cache CDN.
- API Routes memungkinkan komunikasi cepat antara frontend dan backend.

b. **Pengalaman Pengguna yang Lebih Interaktif**

- Pengguna dapat memberikan rating dan menambahkan e-book
- Rekomendasi diperbarui secara dinamis sesuai preferensi pengguna.

c. **Skalabilitas dan Efisiensi**

- Next.js mendukung revalidate untuk memperbarui data tanpa membebani server.
- Dapat diintegrasikan dengan database **MySQL** dan layanan cloud seperti **Vercel**.

I. HASIL DAN PEMBAHASAN

1. Akurasi Rekomendasi

Akurasi rekomendasi diukur dengan precision, recall, dan F1-score untuk mengetahui sejauh mana sistem mampu memberikan rekomendasi yang relevan.

Buku Inti	Nilai (%)
Rekomendasi 1	78.4%
Rekomendasi 2	72.1%
Rekomendasi 3	74.9%
Rekomendasi 4	70.2%

Tabel 1. Tabel Akurasi Rekomendasi.

Hasil evaluasi menunjukkan bahwa sistem memiliki precision yang cukup tinggi, yaitu 78.4%, yang berarti sebagian besar rekomendasi yang diberikan memang relevan dengan preferensi pengguna. Namun, karena jumlah data dan buku yang masih dalam tahap pengembangan maka hasil yang ditampilkan untuk buku yang menunjukan kemiripan masih kurang sehingga butuh penambahan data yang bertahap.

Hal ini mengindikasikan bahwa metode cosine similarity cukup baik dalam mengukur kemiripan teks, tetapi masih memiliki keterbatasan dalam menangkap hubungan semantik yang lebih luas antar buku. Solusi untuk meningkatkan recall adalah dengan mengombinasikan pendekatan hybrid filtering yang menggabungkan metode content-based dengan collaborative filtering.[17]

e-ISSN: 3064-3082
DOI Article: 10.46306/jion.v2i1.210

2. Performa Sistem

Selain akurasi, evaluasi juga dilakukan pada performa sistem rekomendasi yang dikembangkan dengan Next.js.

Pengujian	Client-Side Rendering (CSR)	Server-Side Rendering (SSR)	Static Site Generation (SSG)
Waktu Pemuatan Halaman	2.3 detik	1.5 detik	1.1 detik
Konsumsi Memori	342 MB	233 MB	210 MB

Tabel 2. Performa Sistem Rekomendasi.

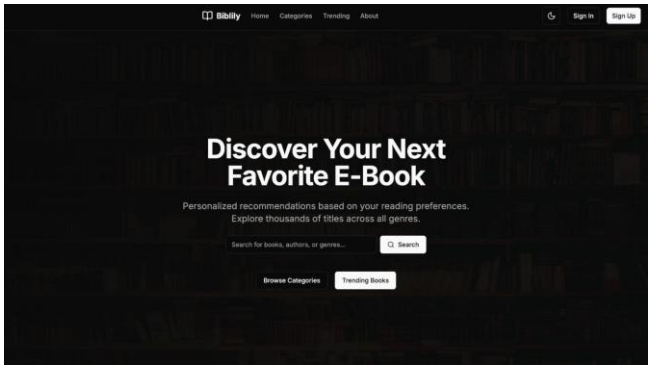
Dari tabel di atas, dapat disimpulkan bahwa penggunaan SSR dan SSG dalam Next.js dapat mempercepat pemuatan halaman dan mengurangi konsumsi memori dibandingkan dengan metode Client-Side Rendering (CSR). SSR memungkinkan sistem untuk merender halaman di server sebelum dikirim ke pengguna, sehingga mempercepat waktu muat awal. Sementara itu, SSG menghasilkan halaman statis yang lebih cepat dan efisien untuk rekomendasi yang tidak sering berubah.[18]

3. Responsivitas Antarmuka

Evaluasi pengalaman pengguna dilakukan dengan menilai responsivitas tampilan antarmuka menggunakan React dan Tailwind CSS. Pengujian dilakukan pada berbagai perangkat untuk memastikan tampilan yang optimal di desktop maupun mobile. Hasilnya menunjukkan bahwa desain antarmuka sistem telah responsif dan memberikan pengalaman pengguna yang nyaman tanpa kendala signifikan dalam navigasi maupun interaksi dengan sistem rekomendasi.[14]

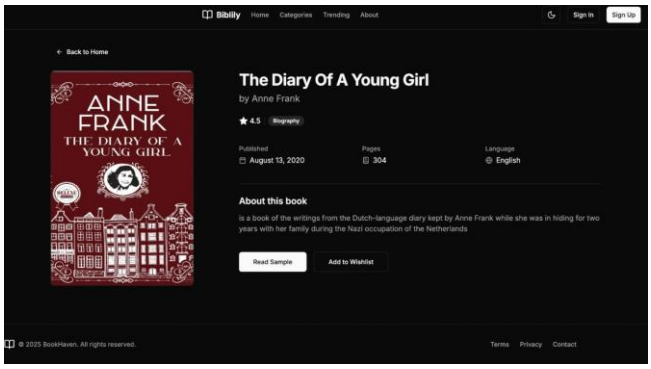
3.1 Antar Muka Layar Utama

e-ISSN: 3064-3082
DOI Article: 10.46306/jion.v2i1.210



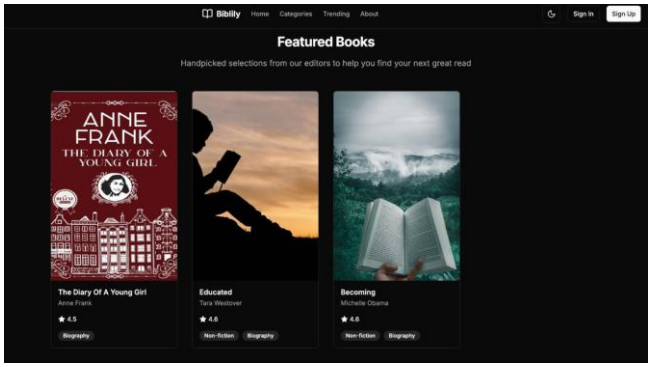
Gambar 9. Interface layar utama.

3.2 Antar muka Layar Buku



Gambar 10. Interface layar buku.

3.3 Antar muka Rekomendasi Buku



Gambar 11. Antarmuka Rekomendasi Buku.

PEMBAHASAN

Dari hasil evaluasi di atas, beberapa temuan penting dapat disimpulkan:

1. Sistem rekomendasi berbasis content-based filtering bekerja dengan baik, namun masih memiliki keterbatasan dalam mengenali hubungan semantik antar buku yang lebih kompleks. Alternatif solusi adalah menerapkan hybrid filtering dengan tambahan data dari perilaku pengguna lain.

2. Penggunaan Next.js dengan SSR dan SSG secara signifikan meningkatkan performa sistem dibandingkan metode CSR. Hal ini berkontribusi pada peningkatan kecepatan

e-ISSN: 3064-3082
DOI Article: 10.46306/jion.v2i1.210

pemuatan halaman dan efisiensi sumber daya.

3. Antarmuka berbasis React dan Tailwind CSS terbukti responsif dan ringan, sehingga memberikan pengalaman pengguna yang lebih optimal di berbagai perangkat. Evaluasi sistem rekomendasi e-book berbasis Next.js dan cosine similarity menunjukkan bahwa pendekatan ini cukup efektif dalam memberikan rekomendasi yang relevan. Akurasi sistem dapat ditingkatkan dengan mengadopsi metode hybrid filtering, sementara optimasi performa dengan Next.js SSR dan SSG memberikan pengalaman pengguna yang lebih cepat dan responsif.

Implementasi ini dapat dikembangkan lebih lanjut dengan menambahkan natural language processing (NLP) untuk meningkatkan pemahaman terhadap konten buku serta menambahkan fitur rekomendasi berbasis tren pengguna agar sistem lebih dinamis.

SIMPULAN DAN SARAN

1. KESIMPULAN

Sistem rekomendasi e-book berbasis content-based filtering yang dikembangkan dengan Next.js menunjukkan hasil yang cukup baik dalam memberikan rekomendasi e-book yang relevan. Penerapan algoritma TF-IDF dan cosine similarity berhasil mengidentifikasi kemiripan antara e-book berdasarkan deskripsi dan metadata, sehingga dapat memberikan rekomendasi yang lebih personal kepada pengguna. Sistem ini masih memiliki keterbatasan, terutama dalam menangani cold-start problem, di mana e-book baru atau pengguna baru memiliki sedikit informasi yang dapat dianalisis. Selain itu, sistem masih belum mampu menangkap hubungan semantik yang lebih kompleks antara buku dengan teknik content-based filtering saja.

Dari sisi performa, penggunaan Next.js dengan teknik Server-Side Rendering (SSR) dan Static Site Generation (SSG) terbukti meningkatkan kecepatan pemuatan halaman dan efisiensi penggunaan memori dibandingkan metode Client-Side Rendering (CSR). Hal ini memberikan pengalaman pengguna yang lebih cepat dan responsif. Antarmuka yang dibangun dengan React dan Tailwind CSS juga berhasil menciptakan desain yang modern, responsif, dan mudah digunakan di berbagai perangkat.

.2. SARAN

Untuk pengembangan lebih lanjut, beberapa rekomendasi yang dapat diterapkan adalah:

Menggunakan Hybrid Filtering seperti engombinasikan content-based filtering dengan collaborative filtering dapat meningkatkan akurasi rekomendasi, terutama dalam menangani cold-start problem. Integrasi NLP (Natural Language Processing) menerapkan teknik NLP dapat meningkatkan pemahaman sistem terhadap isi buku secara lebih mendalam, memungkinkan rekomendasi yang lebih akurat dan kontekstual. Peningkatan Sistem Personalisasi, menggunakan data perilaku pengguna, seperti riwayat pencarian dan interaksi sebelumnya, dapat meningkatkan relevansi rekomendasi. Penyesuaian UI/UX untuk meningkatkan aspek personalisasi pada antarmuka, seperti rekomendasi berbasis tren pengguna, dapat membuat sistem lebih interaktif dan menarik [19], [20]. Penggunaan Model Machine Learning yang Lebih Lanjut seperti mengimplementasi deep learning word embeddings atau transformer-based models (BERT) dapat meningkatkan pemahaman terhadap kesamaan antara e-book.

Secara keseluruhan, sistem ini telah membuktikan efektivitas content-based filtering dalam memberikan rekomendasi e-book, namun masih dapat ditingkatkan dengan teknologi yang lebih canggih untuk mencapai hasil yang lebih optimal.

DAFTAR PUSTAKA

D. C. Utomo, V. Atina, and P. Widyaningsih, "PENERAPAN METODE CONTENT BASED FILTERING PADA SISTEM REKOMENDASI PEMILIHAN BUKU REFERENSI RUMAH BELAJAR PANCASILA," *Infotech: Journal of Technology Information*, vol. 10, no. 1, pp. 121–128, Jun. 2024, doi: 10.37365/jti.v10i1.262.

C. Zhang, G. Long, T. Zhou, Z. Zhang, P. Yan, and B. Yang, "When Federated Recommendation Meets Cold-Start Problem: Separating Item Attributes and User Interactions," in *WWW 2024 - Proceedings of the ACM Web Conference*, Association for Computing Machinery, Inc, May 2024, pp. 3632–3642. doi: 10.1145/3589334.3645525.

E. Kannout, M. Grzegorowski, M. Grodzki, and H. S. Nguyen, "Clustering-Based Frequent Pattern Mining Framework for Solving Cold-Start Problem in Recommender Systems," *IEEE Access*, vol. 12, pp. 13678–13698, 2024, doi: 10.1109/ACCESS.2024.3355057.

Faris, "Panduan Lengkap Menggunakan Next.js: Framework React untuk Pengembangan Web Modern," https://soaltekno.lokercepat.id/tutorial-menggunakan-next-js/?utm_source=chatgpt.com.

R. Prasetyo, A. Nugroho, and A. Sugandi, "Meningkatkan Performa Frontend dengan Menggunakan Framework Next.Js dalam Pengembangan Website," *Journal of Cyber Health and Computer*, vol. 2, no. 2, 2024, doi: 10.18196/jochac. v3i4.

M. Azis Ramadhan, I. Najiyah, R. Muhammad Abillutfi, R. Musaropah, and N. Dian Pramanik, "IMPLEMENTASI DAN EVALUASI SISTEM REKOMENDASI MUSIK BERBASIS LIRIK DENGAN ALGORITMA TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF)."

T. Ridwansyah, B. Subartini, and S. Sylviani, "Penerapan Metode Content-Based Filtering pada Sistem Rekomendasi," *Mathematical Sciences and Applications Journal*, vol. 4, no. 2, pp. 70–77, Apr. 2024, doi: 10.22437/msa. v4i2.32136.

Asa Dilla Safitri, Vihi Atina, and Anisatul Farida, "Sistem rekomendasi buku menggunakan metode content-based filtering," *INFOTECH : Jurnal Informatika & Teknologi*, vol. 5, no. 2, pp. 218–227, Dec. 2024, doi: 10.37373/infotech.v5i2.1302.

A. A. Fikhri and N. Nurdin, "IMPLEMENTASI ALGORITMA K-NEAREST NEIGHBOR PADA SISTEM PEMANTAU SUHU DAN KELEMBAPAN RUANG SERVER MENGGUNAKAN PROTOKOL MQTT BERBASIS IOT," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3S1, Oct. 2024, doi: 10.23960/jitet.v12i3S1.5422.

L. Cahyani, N. Sephiana, M. Tahir, J. Aisyiah, and S. Artikel, "Jurnal Explore IT|31 Sistem Rekomendasi Wisata Kuliner Madura Menggunakan Content Based Filtering Madura Culinary Tourism Recommendation System Using Content Based Filtering INFO ARTIKEL ABSTRAK," 2024, doi: 10.35891/explorit.

A. Sanjaya, A. Bagus Setiawan, U. Mahdiyah, I. Nur Farida, A. Risky Prasetyo, and U. Nusantara PGRI Kediri, "PENGUKURAN KEMIRIPAN MAKNA MENGGUNAKAN

COSINE SIMILARITY DAN BASIS DATA SINONIM KATA MEASUREMENT OF MEANING SIMILARITY USING COSINE SIMILARITY AND WORD SYNONYMS DATABASE," vol. 10, no. 4, 2023, doi: 10.25126/jtiik.2023106864.

E. P. Putra, C. Batara, and E. Dephtios, "Perancangan E-Commerce Pada Toko Boxermks Dengan Penerapan Sistem Rekomendasi Menggunakan Metode Cosine Similarity," 2024.

S. Rahmadhani *et al.*, "JIP (Jurnal Informatika Polinema) SISTEM REKOMENDASI PENELUSURAN BUKU BERBASIS CONTENT-BASED FILTERING DENGAN PEMBOBOTAN TF-RF".

Angga Risky S, "Belajar Pake Next JS, Bikin Website Jadi Modern!" https://buildwithangga.com/tips/belajar-pake-next-js-bikin-website-jadi-modern.

I. Dhimas Maulana and Y. A. Susetyo, "Implementasi Fetch API dalam pengembangan Backend Website Daftar Film dengan Next.JS," 2025.

Murtafi Digital, "Panduan Menggunakan Next.js untuk Desain Website Perusahaan," https://www.murtafidigital.co.id/panduan-menggunakan-next-js-untuk-desain-website-perusahaan/.

K. R. PUTRA and M. A. RACHMAN, "Perbandingan Metode Content-based, Collaborative dan Hybrid Filtering pada Sistem Rekomendasi Lagu," *MIND Journal*, vol. 9, no. 2, pp. 179–193, Dec. 2024, doi: 10.26760/mindjournal. v9i2.179-193.

S. Pati and Y. Zaki, "Evaluating the Efficacy of Next.js: A Comparative Analysis with React.js on Performance, SEO, and Global Network Equity," Jan. 2025, [Online]. Available: http://arxiv.org/abs/2502.15707

H. H. Ben kora and M. S. Manita, "Modern Front-End Web Architecture Using React.js and Next.js," *University of Zawia Journal of Engineering Sciences and Technology*, vol. 2, no. 1, pp. 1–13, Aug. 2024, doi: 10.26629/uzjest.2024.01.

R. I. Prasetyo, "Membaca Pikiran Pengguna dalam Strategi UX/UI (Menciptakan Pengalaman Digital yang Menarik dan Berkesan)." [Online]. Available: https://www.researchgate.net/publication/389494123