Journal of Information Technology and Computer Science (INTECOMS)
Volume 8 Nomor 2, Tahun 2025
e-ISSN : 2614-1574
p-ISSN : 2621-3249

MOVIE TITLE SEARCH WITH FULL-TEXT SEARCH USING ELASTIC SEARCH

PENCARIAN JUDUL FILM DENGAN FULL-TEXT SEARCH MENGGUNAKAN ELASTIC SEARCH

Iqbal Kurniawan[1], Jati Sasongko Wibowo[2]
Universitas Stikubank Semarang[1.2]
kurniawaniqbal476rpl@gmail.com[1]

ABSTRACT

The number of movie catalogs stored becomes larger as more titles are released each year. Naturally, this poses a challenge to find fast and relevant data. Traditional searches using relational databases, such as : PostgreSQL often gives less than optimal results and takes longer time. Therefore, a more efficient data search method is needed, one of which uses full-text search in elasticsearch. This research aims to implement full-text search using elasticsearch in the movie catalog information system. Researchers used documentation method to get movie data sourced from github and used waterfall method for system development. In testing, it shows that the data search ratio with elasticsearch is 1.19 to 1.33 times faster than PostgreSQL and the resulting elasticsearch search accuracy value using precission is 70%, recall is 100% and f1 score is 82%. From these tests, it can be concluded that the implementation of full-text search using elasticsearch on the film catalog information system can improve efficiency and user experience in searching for film titles.

***Keywords**: Full-Text Search, Elasticsearch, Information System, Waterfall.*

ABSTRAK

Jumlah katalog film yang disimpan menjadi lebih besar seiring dengan bertambahnya judul yang dirilis setiap tahunnya. Tentu, hal ini menimbulkan tantangan tersendiri untuk mencari data yang cepat dan relevan. Pencarian tradisional dengan menggunakan database relasional, seperti: PostgreSQL sering kali memberikan hasil yang kurang optimal dan membutuhkan waktu yang lebih lama. Oleh karena itu, diperlukan metode pencarian data yang lebih efisien, salah satu nya menggunakan full-text search di elasticsearch. Penelitian ini bertujuan untuk mengimplementasikan full-text search menggunakan elasticsearch di sistem informasi katalog film. Peneliti menggunakan metode dokumentasi untuk mendapatkan data film yang bersumber dari github dan menggunakan metode waterfall untuk pengembangan sistem. Dalam pengujian, menunjukkan rasio pencarian data dengan elasticsearch lebih cepat 1,19 hingga 1,33 kali lebih cepat dibandingkan dengan PostgreSQL dan nilai akurasi pencarian elasticsearch yang dihasilkan dengan menggunakan precission sebanyak 70%, recall sebanyak 100% dan f1 score sebanyak 82%. Dari pengujian tersebut, dapat disimpulkan bahwa implementasi full-text search menggunakan elasticsearch pada sistem informasi katalog film dapat meningkatkan efisiensi dan pengalaman pengguna dalam mencari judul film.

Kata Kunci: Full-Text Search, Elasticsearch, Sistem Informasi, Waterfall.

PENDAHULUAN

Di Era digital saat ini, teknologi berkembang dengan pesat di berbagai industri, salah satunya di industri perfilman. Setiap tahunnya ribuan judul dirilis, hal tersebut menyebabkan bertambahnya katalog film yang disimpan menjadi lebih besar dari yang sebelumnya. Dengan begitu banyaknya pilihan, mencari judul film yang tepat sesuai keinginan masyarakat menjadi tantangan tersendiri.

Menurut survei yang telah dilakukan dengan penyebaran kuisioner ke 72 responden dari kalangan pecinta film, hasil survei menunjukkan bahwa sebesar 88,6% orang merasa kebingungan untuk menentukan film apa yang hendak ditonton, sebesar 84% orang telah mengatakan kebutuhan akan film meningkat sewaktu pandemi (Sitorus & Astrianty, 2024). Untuk mengatasi masalah ini, dibutuhkan sebuah sistem informasi katalog film yang digunakan untuk menyimpan informasi terkait film, seperti: jenis, genre, judul dan type film (Abdul Muni & Khairul Ihwan, 2021) serta dapat mempercepat proses pencarian film yang relevan bagi pengguna. Penerapan

sistem informasi perlu ditunjang dengan teknologi informasi sebagai alat untuk mempercepat pendistribusian data dan informasi agar bisa diakses oleh pengguna (Pratama,2023)

Namun, sistem informasi katalog film dengan pencarian tradisional, sering kali tidak mampu memberikan hasil yang optimal dan membutuhkan waktu yang lebih lama, karena pencarian tradisional menggunakan database relasional seperti : MySql atau PostgreSQL. Database relasional dioptimalkan untuk operasi dasar seperti Create, Read, Update, Delete (CRUD) dan kurang efektif dalam menangani pencarian teks yang kompleks. Salah satu metode yang paling banyak digunakan untuk mencari data di kumpulan data yang besar yaitu dengan Full-Text Search (DeniZ et al., 2023), metode pencarian **Full-Text Search (FTS)** menjadi solusi yang lebih efisien dan efektif dibandingkan pencarian tradisional yang hanya mengandalkan pencocokan kata kunci sederhana (Shrestha, 2022). Studi yang dilakukan oleh (Chaitanya et al., 2019) menunjukkan bahwa pencarian pada dataset yang lebih besar, lebih efektif jika menggunakan full-text search index di MyISAM dibandingkan dengan penggunaan klausa LIKE atau Regular Expression di PostgreSQL.

Salah satu teknologi FTS yang banyak digunakan dan memiliki peforma yang paling optimal dalam menangani pencarian berbasis text, yaitu dengan elasticsearch (Fotopoulos et al., 2023). Elasticsearch merupakan basis data Non-Relasional (NoSql) yang model datanya menggunakan (Javascript Object Notation) bukan model data SQL (Scripted Query Language (Eddy Tungadi et al., 2019). Pencarian data dengan NoSql 0,1 sampai 0,8 detik lebih cepat dibandingkan SQL (Abdi et al., 2021). Penelitian yang dilakukan oleh (Asyhari & Mauludin, 2019), menunjukkan bahwa penggunaan **elasticsearch** pada sistem informasi perpustakaan menghasilkan pencarian yang 15,59 kali lebih cepat dibandingkan menggunakan SQL.

Penelitian lainnya yang dilakukan oleh (Olsson, 2019) menunjukkan bahwa **elasticsearch** memiliki keunggulan dalam menangani pencarian teks besar dan kompleks, dengan waktu pencarian yang cepat, bahkan untuk data tidak terstruktur seperti log audit keuangan. Penelitian lain yang dilakukan mengenai pengembangan sistem pencarian berbasis elasticsearch, yaitu : mengembangkan sistem pencarian referensi Islam berbasis web menggunakan ExpressJS dan VueJS dengan mengimplementasikan elasticsearch yang dilakukan oleh (Hakim et al., 2024), penelitian oleh (Muharam & Fauziah, 2020) mengenai pengembangan sistem pencarian judul skripsi menggunakan Elasticsearch, yang dapat menentukan skor similarity untuk mencegah plagiarisme dalam penulisan skripsi. Dengan sistem ini, mahasiswa dapat mencari judul yang tidak terlalu mirip dengan skripsi yang sudah ada, sehingga menghindari pelanggaran hak cipta atau duplikasi penelitian.

Penelitian lain juga pernah dilakukan mengenai sistem informasi pencarian, yaitu: penelitian yang dilakukan oleh (Hartawan & Yanti, 2022) dengan judul *Sistem Informasi Pencarian Judul Skripsi Teknik Informatika Berbasis Web* membuktikan bahwa sistem pencarian berbasis web dapat membantu mahasiswa dalam mencari tema skripsi atau tugas akhir dengan lebih cepat dan akurat. Dengan adanya sistem informasi ini, mahasiswa dapat melakukan pencarian dan pengecekan judul-judul skripsi yang sudah ada, sehingga dapat menghindari duplikasi topik penelitian. Selanjutnya, penelitian yang dilakukan oleh (Hafiz, 2022) dengan judul *Perancangan dan Implementasi Sistem Informasi Pencarian Judul Lagu Menggunakan Metode Progressive Web Apps menunjukkan* bahwa penerapan *Progressive Web Apps* (PWA) dalam sistem informasi pencarian lagu dapat membantu pengguna, terutama di daerah

dengan jaringan internet yang kurang stabil, untuk menemukan informasi tentang lagu-lagu lama tanpa kendala kualitas jaringan internet.

Penelitian ini bertujuan untuk mengimplementasikan Full-Text Search dengan menggunakan Elasticsearch pada sistem informasi katalog film. Dengan adanya hal tersebut diharapkan dapat memberi pengalaman bagi pengguna dalam mencari data informasi film secara cepat dan akurat.

METODE

Perumusan Obyek Penelitian

Perumusan Obyek Penelitian ini berfokus kepada:

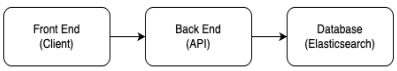
1. Implementasi pencarian full-text dengan elasticsearch.
2. Perbandingan kecepatan pencarian antara elasticsearch dengan database relasional.
3. Pengembangan antarmuka sistem katalog film.

Metode Pengumpulan Data

1. Metode dokumentasi dengan metode ini peneliti mengumpulkan data dari dokumen yang sudah ada. Data tersebut sebelumnya sudah dikumpulkan oleh pihak lain dan sudah dipublish ke publik. Data yang digunakan peneliti bersumber dari github, berikut link nya :
   https://raw.githubusercontent.com/prust/wikipedia-movie-data/master/movies.json

Metode Pengembangan Sistem

1. Waterfall
   Metode waterfall merupakan metode yang pengerjaannya bersifat berurutan (Kurniawati, Mohammad Badrul, 2021). Berikut tahapan pengembangan sistem menggunakan metode waterfall, diantaranya: analisis kebutuhan, desain sistem atau perancangan, implementasi dan pengujian.
2. Arsitektur Sistem
   Arsitektur sistem yang digunakan yaitu: berbasis client-server, arsitektur tersebut memiliki komponen utama, yaitu :
   - Client (Front End)
     Pada komponen ini, merupakan antarmuka yang digunakan oleh pengguna untuk melakukan pencarian informasi film dengan cara memasukkan data berupa text.
   - API (Back End)
     Untuk komponen selanjutnya, yaitu komponen Back End. Komponen ini bertanggung jawab untuk mengelola inputan dari sisi client, berkomunikasi dengan elasticsearch dan menampilkan respons data.
   - Database (Elasticsearch)
     Elasticsearch merupakan database NoSql sekaligus engine pencarian yang digunakan untuk melakukan pencarian full-text terhadap data yang sudah diindeks kedalam elasticsearch.



Gambar 1. Arsitektur Sistem

Lingkungan Pengembangan

Berikut lingkungan pengembangan yang diperlukan, yaitu :

- Database : Elasticsearch
- Container : Docker
- Browser : Google Chrome
- Code Editor : Visual Studio Code
- Pengujian Api : Postman

HASIL DAN PEMBAHASAN

Hasil dari penelitian ini, peneliti berhasil mengembangkan sistem informasi katalog film dengan mengimplementasikan full-text search menggunakan elasticsearch.

Implementasi

Saat proses mengimplementasikan metode full-text search menggunakan

elasticsearch pada sistem informasi katalog film, peneliti melalui beberapa tahapan, yaitu :

1. Import Dataset Film ke Elasticsearch
   a. Pembuatan index
      Peneliti membuat index yang bernama movies, indexs tersebut nantinya digunakan untuk menyimpan data film.

```
let createIndex = await
        esDb.indices.create({
        index: "movies" });
```

Gambar 2. Membuat index elasticsearch

   b. Mengambil data
      Proses pengambilan data menggunakan library axios.

```
const response = await axios.get(
"https://raw.githubusercontent.co
m/prust/wikipedia-movie-
data/master/movies.json" );
```

Gambar 3. Ambil data dengan axios

   c. Menyimpan data
      Setelah proses pengambilan data, data tersebut dimasukkan satu persatu kedalam elasticsearch menggunakan looping.

```
for (let x in data) {
    let insert_data = {
      index: "movies",
      id: id++,
      document: data[x],
    };
    console.log("processing
input data....", insert_data);
    await
esDb.index(insert_data);
 }
```

Gambar 4. Insert Data kedalam Elasticsearch

2. Pencarian Data
   Untuk melakukan pencarian pencarian full-text, peneliti menggunakan fitur, seperti: term, multi match, fuzzy dan bool query. Bool query yang digunakan, yaitu: must.

```
const axios = require("axios");
const esDb =
require("../../config/connection");
class Controller {
 static async search(req, res) {
  try {
   let arr_query = [];

   if (req.query.year) {
    arr_query.push({
     term: { year: req.query.year
},
    });
   }

   if (req.query.cari) {
    arr_query.push({
     multi_match: {
      query: req.query.cari,
      fields: ["title", "cast",
"genres", "extract"],
      fuzziness: "AUTO",
     },
    });
   }

   let data = await esDb.search({
    index: "movies",
    body: {
     query: {
      bool: {
       must: arr_query,
      },
     },
    },
   });
   res.status(200).json({ status:
200, message: "sukses", data: data
});
  } catch (err) {
   res.status(500).json({ status:
500, message:    "gagal", data: err
});
  }
 }
}
module.exports = Controller;
```

Gambar 5. Pencarian data menggunakan elasticsearch

Sebagai pembanding, penelitian ini juga mengimplementasikan pencarian data menggunakan database relasional, database relasional yang digunakan, yaitu

menggunakan PostgreSQL. Berikut tahapan yang dilakukan:

1. Import Dataset ke PostgreSQL
   Pada tahap ini, dilakukan proses memasukkan dataset film kedalam tabel.

```
const response = await axios.get(
    "https://raw.githubusercontent.c
    om/prust/wikipedia-movie-
    data/master/movies.json" );
let data = response.data;
let id = 1;
for (let x in data) {
    let insert_data = {
            index: "movies",
            id: id++,
            document: data[x],
    }
     await esDb.index(insert_data);
}
```

Gambar 6. Import data kedalam PostgreSQL

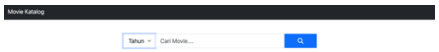
2. Pencarian Data
   Pencarian **data menggunakan query SELECT dan operator LIKE untuk menemukan data berdasarkan kata kunci pencarian.**

```
let where = "";
   if (req.query.year != "" &&
  req.query.cari != "") {
     where = `year =
  ${req.query.year} AND (title
  LIKE '%${req.query.cari}%'
  OR "cast" LIKE
  '%${req.query.cari}%' OR
  genres LIKE
  '%${req.query.cari}%' OR
  extract LIKE
  '%${req.query.cari}%')`;
   } else if (req.query.year != ""
  && req.query.cari == "") {
     where += `year =
  ${req.query.year}`;
   } else if (req.query.cari != ""
  && req.query.year == "") {
     where += `(title LIKE
  '%${req.query.cari}%' OR
  "cast" LIKE
  '%${req.query.cari}%' OR
  genres LIKE
  '%${req.query.cari}%' OR
  extract LIKE
  '%${req.query.cari}%')`;
   }
   let data = await
  sq.query(`SELECT * FROM
  movies WHERE ${where}`);
   res.status(200).json({ status:
  200, message: "sukses", data:
  data });
```

Gambar 7. Pencarian data menggunakan PostgreSQL

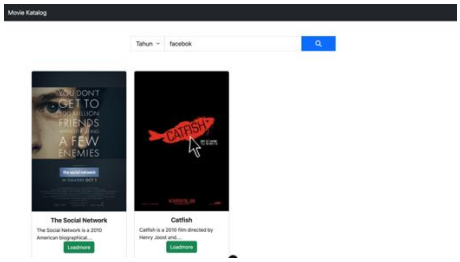
Tampilan User Interface

1. Halaman Awal Pencarian



Gambar 8. Tampilan Awal Pencarian

2. Halaman Daftar Film



Gambar 9. Tampilan Daftar Film

Pengujian Sistem

Pengujian sistem dilakukan untuk membandingkan kinerja pencarian antara elasticsearch dan PostgreSQL. Kinerja yang diuji yaitu: kecepatan pencarian dan kemampuan pencarian dengan toleransi kesalahan *(fuzzy matching)*.

Pengujian Kecepatan Pencarian

Untuk menguji kecepatan pencarian, peneliti menggunakan 2 skenario pengujian, yaitu : pencarian dengan parameter berdasarkan judul dan pencarian berdasarkan parameter judul dan tahun. Keyword yang digunakan, yaitu: passengers untuk judul dan 2016 untuk

tahun. Masing-masing skenario dilakukan sebanyak 5 kali.

Table 1. Hasil Pengujian Kecepatan

Parameter Pencarian	Elasticsearch (ms)	PostgreSQL (ms)	Selisih (ms)
Judul (passengers)	183,2	243,2	60
Tahun dan Judul (2016, passengers)	79	93,8	14,8

Berdasarkan hasil pengujian kecepatan pada table 1, menunjukan bahwa Elasticsearch memiliki waktu pencarian lebih cepat dibandingkan dengan PostgreSQL.

Pengujian Fuzzy Matching

Pengujian ini dilakukan dengan dengan cara memasukkan keyword pencarian yang memiliki kesalahan ejaan, seperti str wrs untuk mencari Star Wars. Berikut hasilnya.

Table 2. Hasil Pengujian Fuzzy Matching

Kata Kunci	Elasticsearch	PostgreSQL
Str wrs	7 Relevan	Tidak Ada

Untuk mengetahui jumlah data yang relevan atau tidak, peneliti menggunakan acuan *confusion matrix.* Acuan tersebut digunakan untuk mengetahui berapa banyak data yang diklasifikasi atau diprediksi benar terhadap data actual *(ground truth).* Berikut confussion matrix yang dihasilkan.

Table 3. *Confusion Matrix* Elasticsearch

Actual	Predicted	
	Relevan	Tidak Relevan
Relevan	TP = 7	FN = 0
Tidak Relevan	FP = 3	TN = 0

- Presisi : 0,7 (70%)
- Recall : 1,0 (100%)
- Skor F1 : 0.82 (82%)

SIMPULAN

Dalam penelitian ini, peneliti berhasil mengimplementasikan elasticsearch kedalam sistem informasi katalog film. Berdasarkan hasil pengujian, pencarian full text yang diimplementasikan kedalam sistem informasi katalog film menggunakan elasticsearch mampu mencari data dengan 1,19 hingga 1,33 kali lebih cepat dibandingkan dengan PostgreSQL, dengan peningkatan kecepatan mulai dari 15,8% hingga 24,7%.. Nilai akurasi pencarian elasticsearch yang dihasilkan dengan menggunakan precission sebanyak 70%, recall sebanyak 100% dan f1 score sebanyak 82%, berdasarkan nilai akurasi tersebut tentu elasticsearch memiliki akurasi yang sangat baik jika dibandingkan dengan PostgreSQL, hal ini karena elasticsearch mampu menangani kesalahan pengetikkan dengan menggunakan fitur fuzzy matching sedangkan PostgreSQL tidak. Dengan adanya penelitian ini, diharapkan dapat meningkatkan pengalaman pengguna dalam mencari informasi film dengan lebih cepat dan relevan.

DAFTAR PUSTAKA

Abdi, M. F., Susanto, A., & Kusnawi, K. (2021). Perbandingan Kecepatan Pencarian Data SQL dan NOSQL. *Jurnal Teknologi Informasi*, *5*(1), 7–11. https://doi.org/10.36294/jurti.v5i1.1696

Abdul Muni & Khairul Ihwan. (2021). Perangcangan Sistem Informasi Film Berbasis WEB. *JUTI UNISI*, *5*(2), 28–33. https://doi.org/10.32520/juti.v5i2.1809

Asyhari, I., & Mauludin, M. S. (2019). IMPLEMENTASI FULL TEXT SEARCH PADA SISTEM INFORMASI PERPUSTAKAAN MENGGUNAKAN LARAVEL. *Jurnal Informatika dan Rekayasa Perangkat Lunak*, *1*(1). https://doi.org/10.36499/jinrpl.v1i1.2759

Chaitanya, B. S. S. K., Reddy, D. A. K., Chandra, B. P. S. E., Krishna, A. B., & Menon, R. R. K. (2019). Full-text Search Using Database Index. *2019 5th International Conference On*

Computing, Communication, Control And Automation (ICCUBEA), 1–5. https://doi.org/10.1109/ICCUBEA47591.2019.9128683

DeniZ, A., Elömer, M. M., & Aydin, A. A. (2023). A comparison of Apache Solr and Elasticsearch technologies in support of large-scale data analysis. *Gümüşhane Üniversitesi Fen Bilimleri Enstitüsü Dergisi*. https://doi.org/10.17714/gumusfenbil.1213317

Eddy Tungadi, M. O., Suwesti Akbar, Baysal, O., Holmes, R., & Godfrey, M. W. (2019). Mining modern repositories with elasticsearch. *Proceedings of the 11th Working Conference on Mining Software Repositories*, 328–331. https://doi.org/10.1145/2597073.2597091

Fotopoulos, G., Koloveas, P., Raftopoulou, P., & Tryfonopoulos, C. (2023). Elasticsearch dalam sistem pencarian referensi Islam berbasis web menggunakan ExpressJS dan VueJS. *Proceedings of the 12th International Conference on Data Science, Technology and Applications*, 406–413. https://doi.org/10.5220/0012089200003541

Hafiz, R. (2022). *Perancangan dan Implementasi Sistem Pencarian Judul Lagu Menggunakan Metode Progressive Web Apps*.

Hakim, F., Harahap, N. S., Handayani, L., & Afriyanti, L. (2024). *IMPLEMENTASI WEB SERVICE REST API UNTUK PENCARIAN RUJUKAN ISLAM MENGGUNAKAN ELASTICSEARCH*. *7*.

Hartawan, S., & Yanti, F. (2022). *Perancangan dan Implementasi Sistem Pencarian Judul Lagu Menggunakan Metode Progressive Web Apps menunjukkan bahwa penerapan Progressive Web Apps (PWA)*. *1*(10).

Kurniawati, Mohammad Badrul. (2021). Penerapan Metode waterfall untuk Perancangan Sistem Informasi Inventory Pada Toko Keramik Bintang Terang. *PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer*, *8*(2), 57–52. https://doi.org/10.30656/prosisko.v8i2.3852

Olsson, J. (2019). *Using Elasticsearch for full-text searches on unstructured data*.

Pratama, A. (2023). Analisis Dan Perancangan Sistem Informasi Persediaan Barang Berbasis Web. *JURNAL TEKNOLOGI DAN SISTEM INFORMASI*, *4*(2).

Shrestha, A. (2022). *Full-Text Search Using Elasticsearch*.

Sitorus, T. A., & Astrianty, L. E. (2024). *Implementasi The Movie DataBase API Untuk Sistem Informasi Film Berbasis Mobile*. *13*(3).