



Penerapan Algoritma BM25 dalam Pencarian Lowongan Pekerjaan pada Website Job Portal

Tek Kheng^{1,*}, Jefry Sunupurwa Asri¹, Sawali Wahyu¹, Yulhendri Yulhendri²

¹ Fakultas Ilmu Komputer, Teknik Informatika, Universitas Esa Unggul, Jakarta, Indonesia

² Fakultas Ilmu Komputer, Sistem Informasi, Universitas Esa Unggul, Jakarta, Indonesia

Email: ^{1,*}tekkheng22@student.esaunggul.ac.id, ²jefry.sunupurwa@esaunggul.ac.id, ³sawaliwahyu@esaunggul.ac.id,

⁴yulhendri@esaunggul.ac.id,

Email Penulis Korespondensi: tekkheng22@student.esaunggul.ac.id

Abstrak—Perkembangan era digital telah berkembang dengan cepat setiap waktunya yang mengubah proses pencarian kerja bagi para pelamar kerja secara signifikan, membuat Job Portal Online menjadi salah satu tempat utama dalam kegiatan rekrutmen sumber daya manusia, walaupun begitu, efektivitas Job Portal Pencarian kerja masih memiliki kelemahan yang mendasar seperti teknologi pencarian kerja yang digunakan masih menggunakan pencocokan string sederhana yang dapat menyebabkan hasil pencarian kurang relevan dan menurunkan kualitas pengalaman pengguna dalam melamar kerja. Penelitian ini dilakukan untuk meningkatkan kualitas dari hasil pencarian lowongan pekerjaan pada Job Portal A Career dengan menerapkan algoritma Okapi BM25. Metode Penelitian ini menggunakan pendekatan pengembangan Rapid Application Development (RAD), seperti perancangan arsitektur client server dengan Next.js sebagai frontednya, ASP.NET Core sebagai backend dan PostgreSQL sebagai database utamanya. Algoritma BM25 ini diintegrasikan langsung ke dalam database menggunakan ekstensi VectorChord BM25 untuk menghitung skor relevansi pencarian dengan kueri yang diinput pengguna. Pada pengujian dengan kueri “accelist the quality support career IT need”, sistem menampilkan 800 dari 1.011 dokumen (79,13%) dengan skor relevansi non-nol. Selain itu, evaluasi melalui User Acceptance Testing (UAT) menunjukkan tingkat kepuasan pengguna sebesar 91,2%, yang menegaskan bahwa BM25 mampu menampilkan hasil paling relevan di peringkat teratas dan mendukung efektivitas sistem pencarian. Dari hasil penelitian ini dapat disimpulkan bahwa algoritma BM25 menjadi solusi pencarian yang lebih efektif, efisien dan memiliki potensi skalabilitas yang tinggi untuk diterapkan pada sistem pencarian lowongan pekerjaan berbasis web lainnya.

Kata Kunci: BM25; Algoritma; Pencarian Kerja; Information Retrieval; Aplikasi Berbasis Web

Abstract—The development of the digital era has grown rapidly all the time which has significantly changed the job search process for job applicants, making Online Job Portals one of the main places in human resource recruitment activities, however, the effectiveness of Job Portals Job search still has fundamental weaknesses such as the job search technology used still uses simple string matching which can cause less relevant search results and reduce the quality of user experience in applying for jobs. This study was conducted to improve the quality of job vacancy search results on Job Portal A Career by applying the Okapi BM25 algorithm. This research method uses a Rapid Application Development (RAD) development approach, such as designing a client server architecture with Next.js as the frontend, ASP.NET Core as the backend and PostgreSQL as the main database. The BM25 algorithm is integrated directly into the database using the VectorChord BM25 extension to calculate the search relevance score with the user inputted query. In testing with the query “accelist the quality support career IT need”, the system displays 800 of 1,011 documents (79.13%) with a non-zero relevance score. Furthermore, evaluation through User Acceptance Testing (UAT) showed a user satisfaction rate of 91.2%, confirming that BM25 is capable of displaying the most relevant results at the top of the rankings and supporting the effectiveness of the search system. The results of this study can be concluded that the BM25 algorithm is a more effective and efficient search solution with high scalability potential for application to other web-based job search systems.

Keywords: BM25; Algorithm; Job Search; Information Retrieval; Web-Based Application

1. PENDAHULUAN

Kemajuan pesat teknologi digital telah berdampak pada kehidupan manusia, termasuk proses rekrutmen dan cara individu mencari peluang karier. Teknologi informasi kini memainkan peran krusial dalam dunia bisnis modern, di mana *portal* lowongan kerja daring telah menjadi jembatan utama yang menghubungkan perusahaan dengan para pencari kerja. Urgensi untuk memanfaatkan teknologi ini menjadi semakin signifikan di tengah tantangan ketenagakerjaan yang dihadapi Indonesia. Menurut data dari Badan Pusat Statistik (BPS), tingkat pengangguran di Indonesia pada Februari 2024 masih mencapai 7,19 juta orang. Meskipun angka ini menunjukkan tren menurun dari tahun-tahun sebelumnya, hal ini tetap menjadi isu kritis yang memerlukan solusi inovatif. Salah satu solusi tersebut adalah platform pencarian kerja yang efektif. Namun, efektivitas platform-platform digital ini sering kali terhambat oleh teknologi pencarian yang usang dan sederhana. Lebih lanjut, upaya untuk memastikan visibilitas dan peringkat yang optimal dalam platform semacam itu sering kali berkaitan dengan prinsip-prinsip optimisasi mesin pencari (SEO) [1].

Masalah utama penelitian yang diidentifikasi dalam studi kasus *portal* A Career adalah ketergantungannya pada metode pencocokan string literal yang dieksekusi sepenuhnya di sisi frontend. Pendekatan ini, di mana semua data lowongan pertama-tama dimuat ke dalam browser pengguna dan kemudian disaring menggunakan fungsi JavaScript sederhana seperti `.includes()`, memiliki kelemahan mendasar. Pertama, proses ini sangat tidak efisien dan membebani browser klien, terutama pada perangkat dengan performa rendah. Kedua, dan yang paling kritis, metode ini gagal memahami relevansi kontekstual. Hasil pencarian sering kali tidak akurat; misalnya, pencarian untuk “*Fullstack Developer*” mungkin menampilkan lowongan “*Backend Developer*” atau “*Intern*” yang hanya mengandung salah satu kata dari kueri. Ketiga, arsitektur ini memiliki skalabilitas yang sangat rendah; performa menurun drastis seiring dengan bertambahnya *volume* data lowongan, sehingga tidak sesuai untuk pertumbuhan jangka panjang platform.



Dalam bidang Temu Kembali Informasi (*Information Retrieval*), berbagai model telah dikembangkan untuk mengatasi masalah relevansi [2]. Sebagai contoh, sebuah studi membandingkan Model *Boolean* dan *Vector Space Model* (VSM) [3], dan menyimpulkan bahwa VSM lebih unggul dalam mengukur similaritas. Namun, Model *Boolean* tidak dapat memberikan hasil yang terperingkat, dan kinerja VSM menurun pada himpunan data yang besar. Sebuah studi yang lebih baru mengusulkan modifikasi *Vector Space Model* yang bertujuan untuk meningkatkan temu kembali informasi semantik [4]. Penelitian ini berupaya mengatasi keterbatasan tersebut dengan menggunakan model probabilistik yang lebih canggih.

Studi lain [5] menerapkan algoritma TF-IDF yang dikombinasikan dengan *Cosine Similarity* untuk sistem rekomendasi lowongan pekerjaan. Metode ini populer dan cukup efektif untuk mencocokkan dokumen dengan struktur serupa, seperti CV dengan deskripsi pekerjaan. Namun, pendekatan TF-IDF standar tidak secara eksplisit menyertakan faktor normalisasi panjang dokumen, yang dapat menyebabkan bias di mana dokumen yang lebih panjang cenderung mendapatkan skor lebih tinggi meskipun kurang relevan. Penelitian ini mengatasi keterbatasan tersebut dengan menggunakan algoritma yang memasukkan panjang dokumen sebagai faktor kunci dalam perhitungannya.

Algoritma Okapi BM25 sendiri telah terbukti efektif di berbagai domain. Studi terbaru [6] berhasil menerapkannya pada aplikasi pencarian berita Polresta Pekanbaru berbasis Android, menghasilkan presisi antara 74%–92% dan *recall* 100%, yang menunjukkan kinerja tinggi dalam konteks berita berbahasa Indonesia. Lebih lanjut, penelitian ini mengusulkan algoritma BM25 yang disempurnakan untuk dukungan keputusan klinis [7]. BM25 juga telah ditingkatkan dengan teknik *stemming* Bahasa Indonesia untuk meningkatkan kinerja pencarian dokumen tugas akhir [8]. Penelitian lain menyoroti penggunaan BM25 dalam sistem rekomendasi kerja berbasis deep learning dengan pendekatan *hyper-personalization* [9]. Perbedaan utama penelitian ini dari penelitian lainnya adalah pada fokusnya terhadap implementasi menyeluruh (*end-to-end*) algoritma BM25 dalam arsitektur tiga lapis (*three-tier*) yang spesifik dan modern (Next.js, ASP.NET Core, PostgreSQL) untuk domain portal kerja [10], [11], [12]. Penelitian ini tidak hanya menguji algoritma, tetapi juga merancang dan mengevaluasi arsitektur pendukungnya untuk menyelesaikan masalah bisnis di dunia nyata [13].

Untuk mengatasi tantangan teknis itu, maka dilakukan perancangan dan implementasi algoritma Okapi BM25 (Best Matching 25), yang merupakan metode canggih (*state-of-the-art*) dalam pemeringkatan probabilistik, dan telah digunakan dalam berbagai penelitian terkini di Indonesia untuk pencarian dokumen skripsi [14]. BM25 dipilih karena kemampuannya yang unggul dalam menghitung skor relevansi dengan mempertimbangkan tiga faktor krusial: frekuensi istilah (*term frequency*), frekuensi dokumen terbalik (*inverse document frequency*), dan normalisasi berdasarkan panjang dokumen. Penelitian ini bertujuan untuk membuat rancangan, mengimplementasikan, lalu mengevaluasi algoritma BM25 dalam arsitektur yang telah disebutkan sebelumnya pada *portal web* lowongan kerja ACareer.

Pernyataan kontribusi dari penelitian ini adalah untuk menyajikan sebuah studi kasus implementasi yang komprehensif yang membuktikan bahwa migrasi dari sistem pencarian berbasis frontend yang tidak efisien ke arsitektur yang berpusat pada backend dengan algoritma BM25 dapat secara signifikan meningkatkan akurasi, efisiensi, dan skalabilitas sebuah portal kerja. Hasil penelitian ini menyediakan cetak biru arsitektur yang dapat direplikasi dan telah divalidasi serta menunjukkan dampak positif langsungnya terhadap pengalaman pengguna dan efektivitas platform.

2. METODOLOGI PENELITIAN

2.1 Objek Penelitian

Objek penelitian adalah portal lowongan kerja ACareer yang dikembangkan oleh PT. Accelist Lentera Indonesia. Fokus penelitian ini adalah untuk meningkatkan kualitas pencarian lowongan pekerjaan pada platform tersebut, sehingga pengguna dapat menemukan informasi lowongan pekerjaan yang paling relevan dengan kebutuhan mereka. Data penelitian berupa koleksi 1.011 lowongan pekerjaan yang tersimpan di basis data PostgreSQL. Setiap entri lowongan memiliki beberapa atribut utama berupa *judul pekerjaan*, *deskripsi singkat*, serta *informasi tambahan* (unit bisnis, level pengalaman, dan lokasi). Seluruh data tersebut digunakan sebagai korpus untuk menguji dan menerapkan algoritma BM25 dalam proses pencarian. Algoritma BM25 ini digunakan untuk menghitung tingkat relevansi antara kata kunci (kueri) yang dimasukkan pengguna dengan data lowongan yang ada, sehingga diharapkan dapat memberikan hasil pencarian yang lebih tepat dan sesuai.

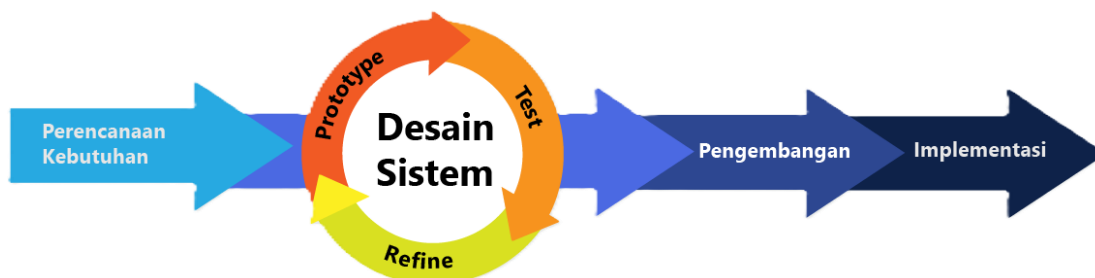
2.2 Metodologi Pengembangan Aplikasi (*Rapid Application Development*)

Metode penelitian yang dipakai adalah *Rapid Application Development* (RAD), metode ini dipilih karena kemampuannya mengembangkan aplikasi secara cepat melalui pembuatan prototipe secara iteratif. Metode Dengan pendekatan RAD, proses pengembangan dapat lebih fleksibel dan cepat, karena setiap tahap pengembangan dapat langsung diuji dan diperbaiki berdasarkan masukan yang diperoleh [15]. Pemilihan metode RAD dianggap paling sesuai untuk penelitian ini, mengingat fokus penelitian bukan hanya pada pengembangan fitur baru, tetapi juga pada penerapan dan pengujian algoritma BM25 dalam sistem pencarian. Dengan RAD, tim dapat langsung memvalidasi efektivitas algoritma tersebut terhadap data lowongan yang ada, kemudian melakukan perbaikan jika ditemukan hasil yang kurang optimal.



2.2.1 Tahapan Penelitian

Tahapan penelitian ini mengikuti langkah-langkah yang terdapat dalam metode Rapid Application Development (RAD). Proses penelitian dibagi menjadi empat tahap utama, seperti diilustrasikan pada Gambar 1:



Gambar 1. Tahapan Penelitian Metode Rapid Application Development (RAD)

- Perencanaan Kebutuhan (*Requirement Planning*)**
Penelitian ini dimulai dengan mengidentifikasi masalah utama pada sistem sekarang, yaitu rendahnya relevansi hasil pencarian dan kecepatan performa platform ACareer yang masih perlu ditingkatkan untuk jangka panjang. Dari hasil analisa sistem berjalan saat ini, ditemukan bahwa fungsi pencarian lowongan pekerjaan saat ini masih sepenuhnya dilakukan di sisi *frontend*, dengan menggunakan pencocokan string sederhana yang kurang efisien dan tidak cocok untuk skala besar. Oleh karena itu dilakukan Tinjauan pustaka tentang *Information Retrieval* (IR) dan algoritma BM25 untuk membangun dasar teori yang kuat sebagai fondasi pengembangan solusi yang akan diimplementasikan.
- Desain Sistem (*User Design & Prototyping*)**
Setelah dilakukan analisa masalah sistem sebelumnya, selanjutnya dirancang solusinya yaitu dengan mengimplementasikan algoritma BM25 ini untuk meningkatkan hasil pencarian lowongan pekerjaan menggunakan ekstensi *vchord_bm25* di PostgreSQL. Sistem akan dirancang dengan arsitektur tiga tingkat yaitu: *Frontend* yang menggunakan Next.js dan TypeScript, *Backend* yang menggunakan ASP.NET Core dan menggunakan PostgreSQL sebagai *Database*.
- Pengembangan (*Construction*)**
Algoritma BM25 ini akan diimplementasikan ke *Database*, dimulai dengan memasang ekstensi *pg_tokenizer* dan *vchord_bm25* pada PostgreSQL, kemudian menambahkan kolom *bm25_vector* pada tabel *Vacancy*, dan membuat indeks BM25 untuk mengoptimalkan kinerja pencarian. Tahapan Prapemrosesan data dimulai dari normalisasi, tokenisasi, *stopword removal*, dan *stemming* akan dilakukan secara otomatis oleh ekstensi *bm25nya*. Selanjutnya membuat *endpoint* API baru di sisi *backend* ASP.NET Core untuk menerima inputan kueri pencarian dari user dan mengeksekusi kueri SQL mentah menggunakan *Dapper*, yang menghitung skor relevansi pencarian untuk setiap lowongan pekerjaan dan akan mengembalikan hasilnya yang sudah diurutkan.
- Implementasi dan Pengujian (*Cutover*)**
Terakhir melakukan pengujian sistem untuk mengecek solusi sistem bekerja sesuai yang diinginkan, pengecekan memakai metode *black-box* dan *User Acceptance Testing* (UAT) untuk memastikan fungsionalitas, efektivitas dan relevansi pencarian sudah sesuai. Pengujian difokuskan pada evaluasi sistem dari hasil feedback yang diberikan user, memastikan pencarian yang relevan dan bekerja dengan baik serta integritas sistem secara keseluruhan.

2.3 Desain Arsitektur dan Metode Penyelesaian

Penyelesaian masalahnya adalah dengan merancang sistem baru memakai arsitektur tiga lapis (*three-tier*) dan mengimplementasikan algoritma Okapi BM25 dalam mesin pencarian lowongan kerja.

2.3.1 Arsitektur Sistem

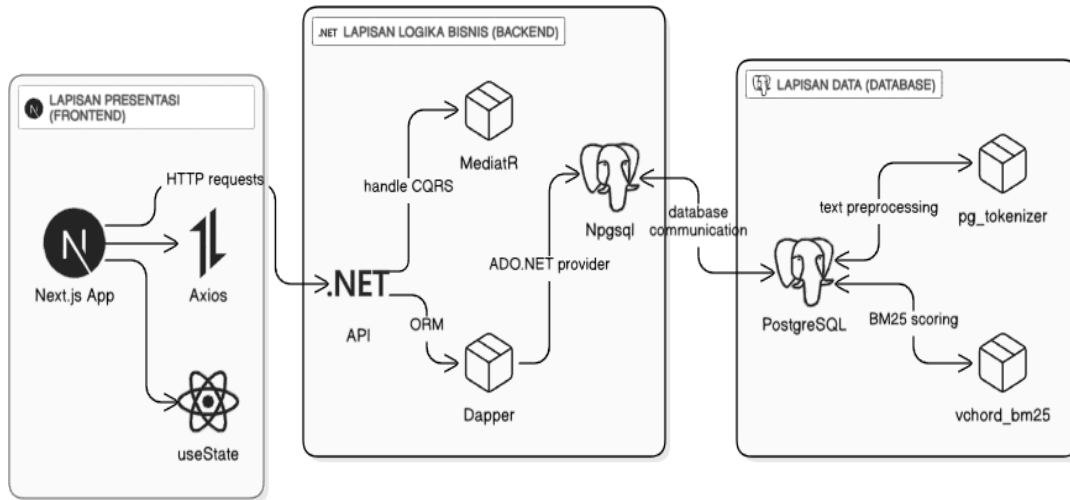
Untuk mengatasi kelemahan arsitektur lama, dirancang arsitektur baru dengan sistem tiga lapis, seperti yang digambarkan pada Gambar 2:

- Lapisan Presentasi (*Frontend*)**
Lapisan ini dibangun menggunakan Next.js dan TypeScript. Perannya adalah sebagai antarmuka pengguna (*user interface*) yang menampilkan halaman pencarian dan hasil pencarian lowongan pekerjaan. Di lapisan ini, pengguna dapat memasukkan kata kunci pencarian, memfilter lowongan berdasarkan kategori tertentu, serta melihat detail informasi lowongan [16].
- Lapisan Logika Bisnis (*Backend*)**
Lapisan ini dikembangkan menggunakan ASP.NET Core. Fungsinya adalah menangani seluruh proses logika aplikasi, termasuk menerima permintaan pencarian dari frontend, memproses kata kunci, serta menjalankan algoritma Okapi BM25 untuk menghitung tingkat relevansi setiap lowongan pekerjaan terhadap kueri pencarian pengguna. Prinsip desain web application pada ASP.NET juga diterapkan untuk menjaga modularitas dan maintainability sistem [17]. Arsitektur backend modern, seperti *microservices* dengan *RESTful API Gateway* dan *backend-for-frontend pattern*, menjadi pendekatan yang kuat untuk memisahkan logika bisnis dari presentasi dan data [18].



c. Lapisan Data (*Database*)

Lapisan ini menggunakan PostgreSQL sebagai sistem manajemen basis data utama. PostgreSQL dilengkapi dengan ekstensi `pg_tokenizer` untuk melakukan pra-pemrosesan teks (seperti pemisahan kata dan normalisasi), serta `vchord_bm25` untuk melakukan komputasi skor relevansi BM25 secara langsung di dalam database. Dengan cara ini, proses perhitungan skor dapat dilakukan lebih cepat tanpa perlu memindahkan seluruh data ke lapisan *backend*, sehingga menghemat waktu eksekusi dan sumber daya. Pemilihan PostgreSQL juga didukung oleh studi perbandingan performa dengan MySQL yang menunjukkan PostgreSQL lebih unggul dalam skenario data besar dan kompleks [19].



Gambar 2. Arsitektur Sistem

2.3.2 Implementasi Algoritma BM25

Implementasi algoritma BM25 difokuskan pada integrasi dengan database PostgreSQL untuk efisiensi maksimal. Prosesnya dilakukan melalui lima tahapan utama sebagai berikut:

- Instalasi Ekstensi PostgreSQL. Langkah pertama adalah menginstal ekstensi `pg_tokenizer` dan `vchord_bm25`. Ekstensi ini menyediakan fungsi-fungsi inti yang diperlukan untuk tokenisasi teks dan perhitungan skor algoritma BM25 langsung di level database.
- Pra-pemrosesan Data Otomatis. Setelah ekstensi terpasang, proses pra-pemrosesan data seperti normalisasi (mengubah teks menjadi huruf kecil), tokenisasi (menggunakan model BERT), penghapusan stopwords, dan *stemming* dapat dilakukan secara otomatis oleh fungsi yang disediakan ekstensi.
- Menambahkan Kolom `bm25_vector`. Sebuah kolom baru (`bm25_vector`) ditambahkan ke tabel *Vacancy*. Kolom ini berfungsi khusus untuk menyimpan representasi vektor dari token yang dihasilkan dari judul lowongan.
- Melakukan Tokenisasi Data. Kolom `bm25_vector` diisi dengan hasil tokenisasi dari kolom Title (judul lowongan). Proses ini mengubah setiap judul lowongan menjadi format vektor yang dapat diproses oleh algoritma BM25.
- Pembuatan Indeks BM25. Langkah terakhir dan paling krusial adalah membuat indeks BM25 pada kolom `bm25_vector`. Indeks ini berfungsi untuk mengoptimalkan kinerja pencarian secara drastis, memungkinkan sistem menemukan hasil relevan dengan sangat cepat.

2.4 Teknik Pengujian

Untuk memastikan solusi sistem bekerja sesuai yang diinginkan, dilakukan dua jenis pengujian utama:

- Pengujian *Black-Box*: Pengujian fungsionalitas dilakukan untuk memverifikasi bahwa setiap fitur bekerja sesuai spesifikasi, terutama pada fitur pencarian untuk memastikan sistem memberikan hasil yang relevan dan terurut dengan benar.
- User Acceptance Testing* (UAT): Pengujian ini melibatkan pengguna akhir untuk mengevaluasi efektivitas, efisiensi, dan kepuasan mereka terhadap sistem baru, yang menjadi validasi utama keberhasilan penelitian.

3. HASIL DAN PEMBAHASAN

Hasil utama penelitian ini yaitu berhasilnya implementasi algoritma BM25 ke sistem baru yang secara fundamental mengubah arsitektur lama dan fungsionalitas pencarian di *Job Portal Career*. Pemindahan proses pencarian dari *frontend* ke *backend* berhasil meningkatkan secara signifikan pada efisiensi, relevansi pencarian dan skalabilitas pada sistem. Pada bagian ini akan menjelaskan hasil teknis dari implementasi BM25 ini dan pembahasan mendalam mengenai dampak perubahannya.



3.1 Implementasi Teknis Algoritma BM25

3.1.1 Arsitektur Sistem

Arsitektur sistem yang diimplementasikan adalah tiga lapis (three-tier architecture), terdiri atas:

- Lapisan Presentasi (*Frontend*)
Menggunakan *framework* Next.js dan TypeScript, berfungsi untuk menampilkan antarmuka pencarian. *Frontend* hanya bertugas mengirimkan *query* pencarian dan menampilkan hasil dalam bentuk daftar pekerjaan yang sudah diperingkat berdasarkan relevansi.
- Lapisan Logika Bisnis (*Backend*)
Dibangun dengan ASP.NET Core, yang berfungsi sebagai penghubung antara *frontend* dengan database. *Backend* juga bertugas membangun *query* SQL dinamis sesuai *input* pengguna, kemudian mengeksekusinya ke database, dan mengembalikan hasil ke *frontend* dalam format JSON.
- Lapisan Data (*Database*)
Menggunakan PostgreSQL yang diperkuat dengan dua ekstensi utama: `pg_tokenizer` untuk pemrosesan teks, dan `vchord_bm25` untuk perhitungan skor BM25. Seluruh proses penilaian relevansi dilakukan langsung di database, sehingga mengurangi beban *backend* dan *frontend*.

3.1.2 Persiapan Database dan Ekstensi

Proses implementasi algoritma BM25 dimulai dengan pengaturan pada level *database*. Langkah-langkah yang dilakukan adalah:

- Aktivasi ekstensi:
Ekstensi `pg_tokenizer` dan `vchord_bm25` diaktifkan agar PostgreSQL mampu melakukan tokenisasi teks dan menghitung skor BM25 secara langsung.
- Penambahan kolom BM25:
Kolom `bm25_vector` ditambahkan pada tabel `Vacancies`. Kolom ini menyimpan representasi vektor hasil tokenisasi dari judul lowongan kerja.

```
ALTER TABLE "Vacancies"  
ADD COLUMN bm25_vector bm25;
```
- Pengisian tokenisasi:
Judul lowongan kerja ditokenisasi dengan model BERT, yang mampu memahami konteks bahasa lebih baik daripada tokenisasi konvensional.

```
UPDATE "Vacancies"  
SET bm25_vector = tokenize(coalesce("Title", ''), 'bert');
```
- Pembuatan indeks BM25:
Indeks khusus dibuat untuk mempercepat pencarian. Dengan indeks ini, PostgreSQL tidak perlu melakukan full table scan, tetapi dapat langsung mencari berdasarkan struktur BM25.

```
CREATE INDEX IF NOT EXISTS idx_vacancies_bm25_vector  
ON "Vacancies" USING bm25 (bm25_vector bm25_ops);
```

3.1.3 Pengembangan *Backend* dan Kueri Pencarian BM25

Backend pada sistem ini dikembangkan menggunakan ASP.NET Core sebagai kerangka kerja utama, dengan bantuan Dapper sebagai micro-ORM. Pemilihan Dapper didasarkan pada kemampuannya mengeksekusi *query* SQL mentah dengan *overhead* yang sangat rendah, sehingga kinerja tetap optimal meskipun harus menangani proses pencarian yang intensif. Kecepatan eksekusi menjadi faktor penting dalam penelitian ini, karena hasil pencarian diharapkan muncul secara cepat meskipun jumlah data lowongan cukup besar. Di bawah ini merupakan kueri inti BM25 yang digunakan dalam sistem pencarian lowongan:

-- Struktur Inti Kueri Pencarian BM25

```
SELECT  
  v."Id",  
  v."Title",  
  -- Kolom lain yang relevan  
  (v.bm25_vector <&> to_bm25query('idx_vacancies_bm25_vector',  
    tokenize(@keyword, 'bert'))) AS BM25Score  
FROM  
  "Vacancies" v  
WHERE  
  v.bm25_vector IS NOT NULL  
  AND v.bm25_vector <&> to_bm25query('idx_vacancies_bm25_vector',  
    tokenize(@keyword, 'bert')) > 0  
ORDER BY  
  BM25Score DESC;
```

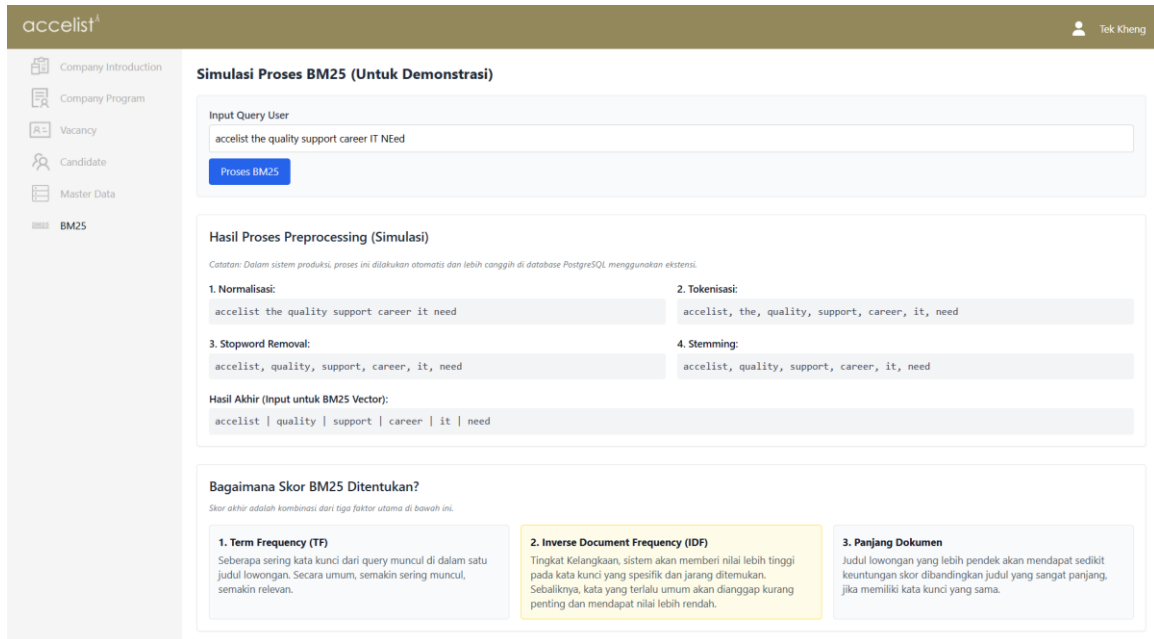


Fungsi `tokenize(@keyword, 'bert')` memastikan bahwa input dari pengguna diproses dengan cara yang sama seperti dokumen, sehingga konsistensi terjaga. Operator `<&>` adalah inti dari BM25 yang menghitung skor relevansi antara *query* pengguna dengan dokumen yang ada.

3.2 Hasil dan Analisis Fungsional Sistem

3.2.1 Tampilan Hasil Pencarian

Bukti peningkatan efisiensi dan efektivitas dari implementasi ini dapat diamati melalui fitur simulasi proses BM25 di sisi admin serta hasil yang berjalan sesuai harapan di tampilan pengguna akhir. Gambar 3 menunjukkan simulasi proses BM25 dari *input query* pencarian hingga hasil proses preprocessing oleh ekstensi BM25 dan penjelasan bagaimana skor BM25 ditentukan.



Gambar 3. Halaman Langkah-Langkah Praproses BM25

Setelah kueri diproses oleh *backend* dan skor dihitung, sistem akan menampilkan hasil akhir kepada pengguna. Gambar 4 dibawah ini memperlihatkan daftar lowongan pekerjaan yang sudah diperingkat berdasarkan skor relevansi. Hasilnya diurutkan secara eksplisit berdasarkan BM25Score dalam urutan menurun, memastikan bahwa lowongan dengan skor tertinggi dan paling relevan muncul di bagian paling atas. Hal ini menunjukkan bahwa fungsi utama dari algoritma BM25 telah berhasil diimplementasikan.

Judul Lowongan	Token yang Cocok (Simulasi)	Skor BM25
IT Support Accelist Career Need	acelist support career it need	12.881733894348145
Support Specialist Accelist Technology Need	acelist support need	8.550719261169434
IT Support Accelist Career	acelist support career it	7.5876545906066895
IT Support Accelist Career Phone	acelist support career it	7.1782660484313965
IT Support Accelist Career Stand	acelist support career it	7.1782660484313965
IT Support Accelist Career Should	acelist support career it	7.1782660484313965
IT Support Accelist Career Sand	acelist support career it	7.1782660484313965

Gambar 4. Hasil Pencarian Berperingkat Ditampilkan dengan Skor BM25 yang Sesuai

3.2.2 Analisis Detail Perhitungan Skor BM25

Keberhasilan implementasi divalidasi lebih lanjut melalui analisis detail skor. Dilakukan pengujian dengan kueri "acelist the quality support career IT need". Pada tabel 1 memperlihatkan detail skor untuk hasil peringkat pertama, "IT Support Accelist Career Need" (total skor 12.886), pada korpus data dengan 1011 dokumen.

Tabel 1. Detail Perhitungan Skor BM25 untuk Peringkat 1

Query	Tf	Df	IDF	TF Component	Final Score
Accelist	1	708	-0.848	1.053	-0.893
Support	1	197	1.417	1.053	1.492
Career	1	154	1.714	1.053	1.805
It	1	108	2.120	1.053	2.232
Need	1	4	5.411	1.053	8.250



Analisis mendalam pada Tabel 1 secara jelas menunjukkan kecerdasan dan kemampuan dari algoritma BM25. *Term need*, yang hanya muncul di 4 dari 1011 dokumen (sangat langka), memiliki nilai *IDF* yang sangat tinggi (5.411) dan memberikan skor terbesar (8.250). Ini menunjukkan relevansi yang sangat kuat. Sebaliknya, kata *accelist* yang sering muncul di banyak dokumen (tercatat ada di 708 dokumen), menghasilkan nilai *IDF* yang rendah bahkan negatif, sehingga sedikit mengurangi total skor relevansi dokumen tersebut. Ini membuktikan bahwa sistem secara efektif memprioritaskan kata kunci yang lebih spesifik dan diskriminatif, yang merupakan inti dari pencarian relevan dan menjadi keunggulan signifikan dibandingkan metode pencocokan string sederhana. Perbandingan dengan hasil peringkat kedua, "*Support Specialist Accelist Technology Need*" (skor 8.553), semakin memperjelas logika ini. Meskipun dokumen peringkat kedua juga mengandung *term need* yang berbobot tinggi, tapi tidak mengandung *term career* dan *it*. Walaupun nilainya tidak sebesar *need*, tidak munculnya kedua kata ini tetap mengurangi tingkat kecocokan dokumen secara keseluruhan. Ini menunjukkan kemampuan BM25 dalam menyeimbangkan antara adanya *term* yang sangat langka dengan kelengkapan *term* dari kueri asli. Sistem tidak hanya mengandalkan satu kata kunci kuat, melainkan menghitung kontribusi dari setiap kata yang cocok, lalu menyesuaikan hasilnya berdasarkan panjang masing-masing dokumen. Demonstrasi perhitungan ini dihalaman admin, tidak hanya berfungsi sebagai alat validasi teknis tetapi juga meningkatkan kepercayaan pengguna terhadap kecerdasan sistem.

3.3 Pengujian Kinerja dan Dampak Bisnis

3.3.1 Evaluasi Kinerja Sistem

Perubahan arsitektur sistem dari yang berfokus pada *frontend* menjadi berfokus pada *backend* memberikan dampak positif terhadap peningkatan kinerja dan skalabilitas. Pada sistem lama, seluruh data lowongan pekerjaan dikirim ke *frontend*, sehingga membebani sisi *frontend* dan jaringan, yang pada akhirnya menurunkan pengalaman pengguna terutama pada perangkat dengan spesifikasi rendah. Pada sistem baru, masalah tersebut berhasil diatasi sepenuhnya. Proses komputasi yang berat ditangani oleh server *backend* dan *database* yang telah dioptimalkan, sehingga aplikasi di sisi pengguna menjadi lebih ringan dan responsif. Dari sisi skalabilitas, pendekatan arsitektur baru ini jauh lebih baik. PostgreSQL digunakan sebagai basis data yang dipadukan dengan algoritma BM25 serta diperkuat dengan indeks yang dirancang untuk menangani data dalam jumlah besar secara efisien. Hal ini memastikan performa pencarian tetap cepat meskipun jumlah data lowongan pekerjaan terus bertambah, sehingga platform *Job Portal ACareer* lebih aman untuk digunakan dalam jangka panjang.

3.3.2 Dampak pada Proses Bisnis

Peningkatan arsitektur sistem juga memberikan dampak yang signifikan pada sisi bisnis. Secara operasional, tim rekrutmen dapat menyaring kandidat dengan lebih cepat dan akurat, sehingga proses seleksi menjadi lebih efisien, karena waktu yang sebelumnya dihabiskan untuk meninjau lamaran yang tidak sesuai kini dapat dialihkan untuk fokus pada kandidat yang benar-benar memenuhi kriteria. Dengan platform *ACareer* yang telah dioptimalkan, perusahaan dipandang lebih siap secara teknologi, yang pada gilirannya dapat meningkatkan minat pencari kerja potensial untuk bergabung. Hal ini tidak hanya berdampak pada peningkatan jumlah pelamar, tetapi juga pada kualitas kandidat yang masuk. Kemampuan platform untuk memberikan hasil pencarian yang relevan secara instan menjadi salah satu nilai tambah yang membedakan *ACareer* di tengah persaingan ketat antar portal lowongan kerja. Tabel 2 membandingkan operasi bisnis sebelum dan sesudah algoritma BM25 diimplementasikan, dan menunjukkan peningkatan penting di sejumlah area strategis.

Tabel 2. Perbandingan Sistem Sebelum dan Sesudah Implementasi BM25

Aspek	Sebelum BM25(Frontend Search)	Sesudah BM25 (Backend Search)
Lokasi Proses	Frontend	Backend
Efisiensi	Tidak Efisien. Performa menurun drastis seiring bertambahnya data	Jauh lebih efisien. Hanya data hasil yang dikirim ke <i>frontend</i> .
Akurasi	Rendah. Bergantung pada pencocokan string sederhana (.includes()).	Tinggi. Peringkat akurat berdasarkan skor relevansi BM25.
Skalabilitas	Buruk. Tidak dirancang untuk pertumbuhan data jangka panjang.	Sangat baik. Siap menangani volume data dan lalu lintas yang besar.

3.4 Hasil Pengujian Penerimaan Pengguna (UAT)

Untuk memastikan bahwa perbaikan teknis ini benar-benar memberikan nilai bagi pengguna, dilakukan *User Acceptance Testing* (UAT) yang melibatkan 6 partisipan representatif (3 mahasiswa sebagai calon pelamar dan 3 staf HR sebagai pengguna internal). Partisipan diminta untuk menjalankan serangkaian skenario uji yang dirancang untuk mensimulasikan penggunaan di dunia nyata dan kemudian memberikan umpan balik melalui kuesioner. Hasil kuantitatif dari kuesioner, yang dirangkum dalam Tabel 3, menunjukkan penerimaan yang sangat positif. Tingkat kepuasan rata-rata keseluruhan mencapai 91.2%, secara signifikan melampaui ambang batas keberhasilan yang ditetapkan (90%). Skor tertinggi pada aspek "Efisiensi & Kecepatan" (94%) dan "Relevansi Hasil" (92%), dari data hasil pengujian UAT ini secara langsung memvalidasi keberhasilan dari implementasi arsitektur yang berfokus pada *backend* dan algoritma BM25.

**Tabel 3.** Ringkasan Hasil Kuantitatif Kuesioner UAT

Aspek yang Dinilai	Tf	Df
Relevansi Hasil	4.6	92%
Efisiensi & Kecepatan	4.7	94%
Kemudahan Penggunaan	4.5	90%
Kepuasan Keseluruhan	4.6	92%

Hasil kuantitatif yang kuat diperkuat oleh umpan balik pengguna yang positif. Seorang mahasiswa menyebut bahwa pencarian kini jauh lebih akurat kata kunci "*Full Stack*" benar-benar menampilkan lowongan yang sesuai, tidak lagi hasil yang tidak relevan. Dari sisi internal, staf HR menyoroti kecepatan pencarian yang sangat membantu pekerjaannya. Permintaan fitur tambahan seperti sinonim dan filter lokasi menunjukkan kepuasan pengguna terhadap sistem yang ada dan keinginan untuk pengembangan lebih lanjut.

3.5 Diskusi dan Kontribusi Penelitian

Keberhasilan teknis dari algoritma atau sistem baru akan lebih berarti jika berdampak langsung pada manfaat nyata bagi bisnis. Penelitian ini memberikan kontribusi yang bisa dilihat dari dua sisi utama: manfaat operasional dan manfaat strategis.

3.5.1 Kontribusi Operasional

Kontribusi operasional dari penelitian ini terlihat jelas dalam peningkatan efisiensi kerja tim rekrutmen. Sistem yang baru mengalihkan beban komputasi dari *frontend* ke *backend*, sehingga portal jadi lebih cepat diakses dan alur kerja jadi lebih lancar. Tim HR kini bisa meninjau lamaran tanpa hambatan seperti sebelumnya. Algoritma BM25 juga sangat membantu karena mampu menyusun hasil pencarian berdasarkan relevansi yang tinggi. Jadi, tim tidak perlu lagi memilah hasil yang tidak nyambung seperti saat masih pakai metode pencocokan sederhana. Ini bikin proses seleksi awal jadi lebih cepat dan akurat. Selain itu, pencari kerja juga merasakan manfaatnya karena pencarian jadi lebih responsif dan akurat, yang secara tidak langsung memperkuat kesan bahwa PT. Accelist Lentera Indonesia adalah perusahaan yang serius dalam teknologi, sekaligus membuat lebih banyak calon pelamar merasa tertarik untuk bergabung.

3.5.2 Kontribusi Strategis

Kontribusi strategis mengacu pada dampak jangka panjang dari penelitian ini terhadap pertumbuhan, daya saing, dan kemampuan adaptasi perusahaan di masa depan. Perancangan ulang arsitektur sistem menjadi model tiga lapis merupakan sebuah investasi strategis. Platform ACareer kini siap untuk menangani pertumbuhan volume data lowongan dan lalu lintas pengguna yang signifikan di masa depan tanpa mengalami degradasi kinerja. Di pasar portal kerja yang sangat kompetitif, pengalaman pengguna yang superior adalah pembeda utama. Dengan memiliki sistem pencarian berkinerja tinggi yang didukung oleh algoritma canggih seperti BM25, PT. Accelist Lentera Indonesia memiliki keunggulan teknologi yang dapat menarik baik pelamar berkualitas maupun perusahaan lain yang mungkin ingin menggunakan platform tersebut. Selain itu, sistem *backend* yang baru, dengan kemampuan pencatatan (*logging*) yang terstruktur, meletakkan dasar untuk analitik data yang lebih mendalam. Perusahaan kini memiliki kemampuan untuk menganalisis tren kata kunci pencarian dan mengoptimalkan strategi rekrutmen berdasarkan data empiris, membuka jalan bagi pengambilan keputusan yang lebih cerdas dan proaktif.

3.5.3 Perbandingan dengan Penelitian Lain

Supaya kontribusi penelitian ini bisa lebih terlihat jelas, penting untuk membandingkannya dengan metode-metode lain yang sudah pernah digunakan dalam penelitian sebelumnya. Pada Tabel 4 ditampilkan perbandingan antara pendekatan yang digunakan dalam studi ini dengan beberapa metode populer yang biasa digunakan dalam sistem pencarian dan rekomendasi. Perbandingan ini tidak hanya mencakup aspek algoritma yang dipakai, tetapi juga mempertimbangkan performa, kompleksitas implementasi, serta kesesuaian metode dengan kebutuhan pengguna. Dengan adanya perbandingan ini, dapat dilihat posisi penelitian ini dalam konteks kajian sebelumnya, sekaligus memperlihatkan keunggulan utama pendekatan yang diusulkan. Selain itu, analisis ini juga memberikan dasar argumentasi yang lebih kuat bahwa pemilihan algoritma BM25 dalam arsitektur tiga lapis yang diusulkan bukan hanya relevan secara teoritis, tetapi juga efektif secara praktis dalam mendukung kebutuhan aplikasi portal pekerjaan modern.

Tabel 4. Perbandingan Metode Pencarian

Penelitian	Metode	Kelebihan	Kekurangan
Hersianty & Amalia [3]	TF-IDF + Cosine Similarity	Implementasi relatif sederhana; Cocok untuk pencocokan dokumen pendek.	Kurang efektif untuk dokumen panjang; Tidak mempertimbangkan posisi kata.
Susanti et al. [2]	Boolean & Vector Space Model	Boolean: sangat cepat; VSM: dapat membedakan tingkat kesamaan.	Boolean: tidak mendukung ranking; VSM: performa menurun pada skala besar.



Penelitian	Metode	Kelebihan	Kekurangan
Penelitian ini	BM25 (PostgreSQL Extension)	Akurasi relevansi tinggi; Mendukung pembobotan cerdas; Performa tinggi dengan <i>indexing</i> .	Membutuhkan pra-pemrosesan teks; Parameter perlu dikalibrasi.

Berdasarkan hasil perbandingan pada Tabel 4, metode BM25 yang diterapkan dalam penelitian ini menunjukkan keunggulan signifikan, terutama dalam hal keakuratan hasil pencarian yang relevan. Hal ini menjadikan BM25 lebih cocok diterapkan pada platform pencarian lowongan kerja seperti ACareer, yang terus berkembang dengan jumlah data lowongan yang terus meningkat. Beberapa kelebihan BM25 dibandingkan metode sebelumnya, seperti TF-IDF + Cosine Similarity atau kombinasi *Boolean* dan *Vector Space Model*, terletak pada kemampuannya memberikan skor relevansi yang lebih akurat, melakukan pembobotan kata secara lebih cerdas dibanding metode konvensional, serta menghasilkan performa tinggi berkat dukungan sistem *indexing* yang efisien. Selain unggul di domain lowongan kerja, BM25 juga terbukti serbaguna karena telah diadopsi dalam domain akademik, misalnya untuk rekomendasi dosen pembimbing tugas akhir [20].

3.5.4 Keterbatasan Penelitian

Meskipun BM25 terbukti efektif, masih terdapat beberapa keterbatasan yang perlu diperhatikan. Algoritma ini tetap bekerja berdasarkan statistik kata seperti frekuensi dan distribusi, tanpa benar-benar memahami makna kalimat secara mendalam. Hal ini membuat sistem belum mampu memahami konteks atau sinonim dari kata yang digunakan pengguna. Salah satu tantangan besar adalah proses *stemming* untuk Bahasa Indonesia. Walaupun BM25 sudah mendukung fitur *stemming*, hasilnya masih belum sepenuhnya optimal karena struktur Bahasa Indonesia yang kompleks. Sistem terkadang kesulitan mengenali bentuk turunan atau variasi kata, sehingga relevansi hasil pencarian bisa sedikit menurun. Namun, keterbatasan ini tidak secara langsung mengurangi kualitas keseluruhan sistem. Sebaliknya, hal ini membuka peluang penelitian lanjutan, khususnya pada aspek pemrosesan bahasa alami (NLP) untuk Bahasa Indonesia agar sistem dapat memahami konteks dengan lebih baik di masa depan.

4. KESIMPULAN

Berdasarkan keseluruhan proses yang sudah dilakukan pada penelitian ini, kesimpulannya yaitu keberhasilan dalam merancang, membangun, dan menguji sistem pencarian lowongan kerja yang lebih baik dengan mengimplementasikan algoritma Okapi BM25 pada platform web job portal ACareer. Pemindehan proses pencarian dari frontend ke backend berhasil mengatasi masalah performa dan meningkatkan akurasi pencarian lowongan pekerjaan secara signifikan. Penerapan algoritma BM25 yang terintegrasi dengan ekstensi PostgreSQL menunjukkan performa yang baik dalam menyusun peringkat hasil pencarian berdasarkan konteks, dengan mempertimbangkan frekuensi kata, panjang dokumen, serta keunikan kata kunci yang digunakan. Hasil akhir sistem juga telah divalidasi melalui pengujian User Acceptance Testing (UAT), yang menunjukkan bahwa para pengguna memberikan tanggapan positif terhadap sistem baru ini, dengan tingkat kepuasan mencapai 91.2%. Mereka menilai sistem ini lebih cepat, lebih mudah digunakan, dan hasil pencariannya lebih sesuai dengan kebutuhan dibandingkan sistem sebelumnya. Dapat disimpulkan bahwa penerapan algoritma BM25 berhasil dilakukan dan mampu mengatasi permasalahan terkait relevansi hasil pencarian serta meningkatkan efisiensi performa sistem. Sistem ini juga memberikan dampak yang bermanfaat bagi operasional PT. Accelist Lentera Indonesia, khususnya dalam mendukung efisiensi proses rekrutmen serta memperkuat fondasi teknologi perusahaan untuk menghadapi persaingan di masa mendatang. Ke depannya, pengembangan sistem menuju pencarian berbasis semantik layak untuk dipertimbangkan sebagai langkah lanjutan. Salah satu pendekatan yang potensial adalah dengan memanfaatkan model embedding seperti SBERT agar sistem mampu memahami sinonim dan makna kata yang lebih dalam. Selain itu, memperluas cakupan pencarian hingga ke bagian deskripsi pekerjaan serta menambahkan fitur filter seperti lokasi, berdasarkan masukan dari pengguna, dapat menjadi fokus penelitian selanjutnya agar sistem semakin komprehensif dan mampu merespons kebutuhan pengguna dengan lebih baik.

REFERENCES

- [1] O. Khatri and K. Muley, "Search Engine Optimization: An Overview," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 2, no. 9, pp. 605–608, 2022, doi: 10.48175/ijarsct-5401.
- [2] K. A. Hambarde and H. Proenca, "Information Retrieval: Recent Advances and Beyond," *IEEE Access*, vol. 11, no. January, pp. 76581–76604, 2023, doi: 10.1109/ACCESS.2023.3295776.
- [3] S. Susanti, M. Azmi, E. Ali, R. Rahmaddeni, and Y. Saputra Wijaya, "Perbandingan Boolean Model Dan Vector Space Model Dalam Pencarian Dokumen Teks," *Digit. Zo. J. Teknol. Inf. dan Komun.*, vol. 11, no. 2, pp. 268–277, 2020, doi: 10.31849/digitalzone.v11i2.4168.
- [4] C. I. Nakpih, "A modified Vector Space Model for semantic information retrieval," *Nat. Lang. Process. J.*, vol. 8, no. June, p. 100081, 2024, doi: 10.1016/j.nlp.2024.100081.
- [5] V. Meida Hersianty, E. Larasati Amalia, D. Puspitasari, and D. Wahyu Wibowo, "Penerapan Algoritma Tf-Idf Dan Cosine Similarity Dalam Sistem Rekomendasi Lowongan Pekerjaan," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 9, no. 1, pp. 1619–1625, 2025, doi: 10.36040/jati.v9i1.12406.
- [6] P. P. Indah and Irwan, "Implementasi Metode Okapi BM 25 Dalam Aplikasi Pencarian Berita Polresta Pekanbaru Berbasis



- Android,” *Mhs. Apl. Teknol. Komput. dan Inf.*, vol. 5, no. 1, pp. 26–31, 2023, doi: 10.35145/jmapteksi.v5i1.
- [7] Z. Zhang, “An Improved BM25 Algorithm for Clinical Decision Support in Precision Medicine Based on Co-Word Analysis and Cuckoo Search,” *BMC Med. Inform. Decis. Mak.*, vol. 21, no. 1, pp. 1–16, 2021, doi: 10.1186/s12911-021-01454-5.
- [8] M. Widiasri, E. Tjandra, and L. M. Chandra, “Peningkatan Kinerja Pencarian Dokumen Tugas Akhir Menggunakan Porter Stemmer Bahasa Indonesia dan Fungsi Peringkat Okapi BM25,” *Teknika*, vol. 6, no. 1, pp. 54–60, 2017, doi: 10.34148/teknika.v6i1.65.
- [9] Q. Q. Abuein, M. Q. Shatnawi, and N. Alqudah, “Improving Job Matching with Deep Learning-Based Hyper-Personalization,” *IAES Int. J. Artif. Intell.*, vol. 13, no. 2, pp. 1711–1722, 2024, doi: 10.11591/ijai.v13.i2.pp1711-1722.
- [10] A. Makris, K. Tserpes, G. Spiliopoulos, D. Zissis, and D. Anagnostopoulos, “Correction to: MongoDB Vs PostgreSQL: A Comparative Study on Performance Aspects (GeoInformatica, (2020), 10.1007/s10707-020-00407-w),” *Geoinformatica*, vol. 25, no. 1, pp. 241–242, 2021, doi: 10.1007/s10707-020-00424-9.
- [11] Harrison Oke Ekpobimi, “Building High-Performance Web Applications with NextJS,” *Comput. Sci. IT Res. J.*, vol. 5, no. 8, pp. 1963–1977, 2024, doi: 10.51594/csitrj.v5i8.1459.
- [12] M. A. Othman and M. R. Girgis, “Solution- Architecture in ASP. Net Core,” *Int. J. Comput. Appl.*, vol. 177, no. 42, pp. 15–24, 2020, doi: 10.5120/ijca2020919923.
- [13] Sapardi, W. H. Kristanto, and N. T. Kurniadi, “Implementasi Pengembangan Aplikasi Sistem Manajemen Aset Berbasis Web Menggunakan Metode Waterfall Untuk Mengoptimalkan Penggunaan Aset Pada PT. Utama Karya (Persero),” *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 5, no. 4, pp. 401–408, 2023, doi: 10.47233/jteksis.v5i4.948.
- [14] R. A. Satria, I. Indriati, and S. Sutrisno, “Pengelompokan Hasil Pencarian Skripsi Berbahasa Indonesia Menggunakan Metode DBSCAN dengan Pembobotan BM25,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 4, pp. 781–790, 2023, doi: 10.25126/jtiik.2024106899.
- [15] Lukman Santoso and Juni Amanullah, “Pengembangan Sistem Informasi Akademik Berbasis Website Menggunakan Metode Rapid Application Development (Rad),” *Elkom J. Elektron. dan Komput.*, vol. 15, no. 2, pp. 250–259, 2022, doi: 10.51903/elkom.v15i2.943.
- [16] M. Fikri, I. M. Drehem, and A. R. Adriansyah, “Penerapan Micro Frontend dengan Next.js dan Module Federation pada Aplikasi Cash Management,” *DBESTI J. Digit. Bus. Technol. Innov.*, vol. 2, no. 1, pp. 35–44, 2025, doi: 10.54914/dbesti.v2i1.1631.
- [17] A. Talekattu, R. Katiyar, and S. B., “Design of Web Applications in ASP.NET,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 12, no. 6, pp. 536–541, 2024, doi: 10.22214/ijraset.2024.63128.
- [18] F. X. Senduk, X. B. N. Najooan, and S. R. U. A. Sompie, “Pengembangan Arsitektur Microservices dengan RESTful API Gateway menggunakan Backend-for-frontend Pattern pada Portal Akademik Perguruan Tinggi,” *J. Tek. Inform.*, vol. 18, no. 1, pp. 315–324, 2023, doi: 10.35793/jti.v18i1.50402.
- [19] A. D. Praba and M. Safitri, “Studi Perbandingan Performansi Antara Mysql Dan Postgresql,” *J. Khatulistiwa Inform.*, vol. 8, no. 2, pp. 88–93, 2020, doi: 10.31294/jki.v8i2.8851.
- [20] S. Safril, L. O. M. Golok Jaya, and J. Nangi, “Penerapan Metode Bm25 Pada Sistem Rekomendasi Dosen Pembimbing Dan Penguji Tugas Akhir Mahasiswa Berbasis Web,” *semanTIK*, vol. 7, no. 1, p. 51, 2021, doi: 10.55679/semantik.v7i1.12562.