

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Tes minat dan bakat merupakan metode evaluasi yang digunakan untuk mengidentifikasi potensi, kecenderungan, serta kompetensi individu secara objektif. Salah satu institusi pendidikan yang menerapkan metode ini adalah Multimedia Nusantara Polytechnic (MNP), sebuah perguruan tinggi vokasi yang berfokus pada pengembangan kompetensi berbasis teknologi dan industri kreatif [1]. Secara konvensional, pelaksanaan tes minat dan bakat dilakukan secara tatap muka melalui sesi evaluasi, peserta mengerjakan serangkaian soal yang kemudian dianalisis secara manual untuk menentukan profil minat dan bakat. Namun, pelaksanaan tes secara konvensional melalui sesi tatap muka cenderung kompleks dan memakan waktu karena membutuhkan ruang fisik serta analisis manual terhadap lembar jawaban yang beragam. Keterbatasan tersebut menghambat efisiensi operasional dan memperbesar risiko kesalahan penilaian, sehingga diperlukan solusi digital yang mampu meningkatkan kecepatan pelaksanaan dan analisis hasil tes. Meskipun platform seperti Google Form dapat digunakan, keterbatasan fleksibilitas dalam penyesuaian antarmuka pengguna (*custom UI*), alur logika soal, serta integrasi hasil analisis membuat platform tersebut belum memadai bagi kebutuhan institusi, sehingga diperlukan pengembangan sistem tes minat dan bakat mandiri yang mampu memenuhi kriteria teknis serta fungsional pihak MNP.

Oleh karena itu, Multimedia Nusantara Polytechnic (MNP) mengembangkan sebuah solusi digital untuk menggantikan metode tes minat dan bakat konvensional, yaitu melalui aplikasi bernama Talent Discovery. Aplikasi Talent Discovery dirancang sebagai sistem tes minat dan bakat berbasis web yang mampu menjawab kebutuhan efisiensi dan fleksibilitas. Melalui sistem ini, proses pengerjaan tes tidak lagi terbatas oleh ruang dan waktu, karena calon mahasiswa dapat mengikuti tes di mana saja dan kapan

saja menggunakan perangkat gawai atau komputer yang terhubung dengan internet.

Target utama pengguna Talent Discovery adalah lulusan SMA/SMK sederajat di seluruh Indonesia. Berdasarkan data statistik dari Kementerian Pendidikan Dasar dan Menengah (Kemendikdasmen), terdapat sekitar 1.676.244 lulusan SMA dan 1.607.959 lulusan SMK pada tahun ajaran 2024/2025 [2][3]. Jumlah ini menggambarkan besarnya potensi calon pendaftar yang dapat menggunakan sistem setiap tahun. Dengan volume lulusan yang terus bertambah dan terakumulasi dari tahun ke tahun, sistem memerlukan kemampuan skalabilitas yang kuat agar performa aplikasi tetap stabil dan responsif meskipun jumlah pengguna meningkat secara signifikan.

Talent Discovery memiliki dua panel utama, yaitu panel user atau peserta dan panel admin. Panel user atau peserta digunakan untuk mengakses serta mengerjakan paket tes yang diambil, sekaligus melihat hasil dari tes yang telah diselesaikan. Sementara itu, panel admin merupakan bagian yang sangat krusial karena digunakan untuk melakukan proses CRUD (*Create, Read, Update, Delete*) terhadap berbagai data penting dalam Talent Discovery. Salah satu komponen terpenting dalam panel admin adalah halaman daftar pengguna atau user, karena halaman ini memuat data pengguna yang bersifat sensitif dan sangat penting bagi pihak MNP.

Namun, setelah Talent Discovery mulai berjalan secara operasional, muncul permasalahan yang berkaitan dengan aspek skalabilitas, khususnya pada panel admin. Permasalahan utama terjadi pada halaman daftar pengguna yang memiliki waktu muat (*load time*) relatif lama dalam menampilkan data pengguna. Terdapat beberapa aspek yang dapat mempengaruhi hasil waktu muat data pada halaman data pengguna seperti, internet yang kurang memadai menyebabkan latensi saat pengiriman data, konfigurasi server yang salah menyebabkan *resource* yang digunakan oleh aplikasi menjadi tidak optimal dalam melayani permintaan data oleh klien, terakhir spesifikasi server yang tidak memadai dalam menjalankan aplikasi sehingga menyebabkan error atau latensi hebat dalam mengirim data. Beberapa aspek tersebut memang dapat

memperbaiki hasil latensi pengiriman data, namun bersifat sementara dan mahal. Karena penyebab utama ada di aspek arsitektur bagaimana data dikirimkan, mekanisme sistem eksisting yang mengambil seluruh data pengguna dari database setiap kali halaman daftar pengguna dibuka, kemudian memuat dan menyimpannya secara penuh di sisi klien (*frontend*).

Penyimpanan seluruh data pengguna di sisi *frontend* dilakukan untuk mendukung fitur pencarian yang dijalankan pada sisi klien menggunakan metode penyaringan sederhana *substring matching* berdasarkan input nama pengguna. Pendekatan pencarian ini mengharuskan seluruh data pengguna tersedia di sisi klien dan hanya mampu mencocokkan sebagian teks yang sama, sehingga memiliki toleransi terbatas dalam menangani variasi penulisan maupun kesalahan pengetikan (*typo*). Keterbatasan tersebut menyebabkan pencarian berpotensi gagal menemukan data meskipun data pengguna sebenarnya tersedia di dalam sistem. Selain kesalahan pengetikan, *substring matching* juga memiliki waktu komputasi yang cukup lambat. Hal ini disebabkan oleh sifatnya yang melakukan pengecekan per-karakter input, lama waktu komputasi ini juga menyebabkan pencarian menjadi patah-patah dan lambat.

Berdasarkan pengujian performa, waktu pemuatan (*load time*) halaman pengguna menunjukkan peningkatan seiring dengan bertambahnya jumlah data. Pada kondisi 1.000 pengguna, halaman masih dapat dimuat dalam waktu 512 ms (*millisecond*) atau sekitar setengah detik, namun ketika jumlah pengguna meningkat menjadi 5.000, waktu muat melonjak hingga 1.970 ms (*millisecond*) atau hampir dua detik. Kenaikan ini menunjukkan keterbatasan sistem dalam menangani kebutuhan pada skala yang lebih besar. Kondisi tersebut berkaitan dengan mekanisme penyajian data eksisting yang memuat seluruh data pengguna ke sisi klien, termasuk untuk mendukung proses pencarian. Pendekatan ini menyebabkan beban pemuatan meningkat seiring pertumbuhan data dan menjadikan proses pencarian tidak lagi efisien serta tidak kompatibel dengan kebutuhan sistem berskala besar.

Waktu muat halaman yang cepat menjadi hal yang krusial bagi admin, waktu tunggu sampai dua detik dapat meningkatkan waktu tunggu sebelum admin dapat melanjutkan aktivitas lain, seperti validasi data mahasiswa yang pastinya dilakukan berulang-ulang untuk setiap mahasiswa. Yang dimana ketika kondisi ini terjadi secara berulang dalam satu siklus kerja, waktu tunggu yang terakumulasi berpotensi menghambat kelancaran dan kecepatan proses operasional yang bergantung pada sistem Talent Discovery. Seiring bertambahnya jumlah data pengguna, pendekatan pemuatan data secara keseluruhan ini menjadi semakin tidak efisien dan menunjukkan keterbatasan sistem dalam mendukung kebutuhan operasional pada skala yang lebih besar.

Selain waktu muat, meningkatkan keterbatasan toleransi kesalahan dan waktu komputasi pencarian juga menjadi permasalahan pada sistem eksisting. Mekanisme pencarian yang digunakan masih mengandalkan *substring matching*, sehingga setiap kueri harus diproses terhadap seluruh data yang tersedia. Kondisi ini menyebabkan pencarian menjadi cukup kaku dan terasa lambat, khususnya ketika kueri yang dimasukkan tidak memiliki kecocokan dengan data di dalam database. Akibatnya, admin harus menunggu proses pencarian selesai meskipun hasil yang diperoleh bersifat nihil, yang menimbulkan waktu tunggu lebih lagi.

Perancangan ulang arsitektur merupakan langkah umum dalam optimasi *software*, optimasi pada tingkat arsitektur berperan penting dalam mengurangi beban pemrosesan dan meningkatkan performa sistem secara keseluruhan [4]. Maka dari itu, penelitian ini berfokus pada perancangan ulang mekanisme penyajian data dan fitur pencarian pada aplikasi Talent Discovery untuk meningkatkan performa pemuatan halaman tanpa menghilangkan fungsi pencarian. Solusi yang diusulkan adalah pengembangan arsitektur baru yang meliputi penerapan *pagination* untuk membatasi jumlah data yang dikirim ke sisi klien dalam satu kali permintaan, sehingga sistem tidak perlu memuat seluruh data pengguna secara sekaligus. Pendekatan ini umum digunakan untuk meningkatkan performa pemuatan data dan menjaga responsivitas sistem pada kondisi jumlah data yang besar. Serta, diusulkan pula mekanisme

pencarian baru dengan memindahkan mekanisme pencarian dari sisi *frontend* ke *backend* agar lebih kompatibel dengan sistem berbasis *pagination*.

Untuk memperoleh relevansi antara input pencarian dan data yang tersimpan dalam database, penelitian ini mengadopsi konsep *similarity detection* dalam *information retrieval*, di mana pengukuran kemiripan (*similarity measure*) atau jarak antar entitas informasi merupakan kebutuhan inti dalam proses penemuan informasi. Pendekatan ini berperan dalam meningkatkan kualitas seleksi informasi sekaligus mengurangi waktu dan biaya pemrosesan [5]. Dengan menerapkan konsep *similarity detection*, sistem pencarian tidak hanya bergantung pada pencocokan teks secara eksak, tetapi juga mampu mengukur tingkat kemiripan antar string sehingga hasil pencarian tetap relevan meskipun terdapat kesalahan pengetikan atau variasi penulisan.

Terdapat beberapa algoritma yang mampu mendeteksi kecocokan (*similarity detection*) antar string, seperti *Winnowing* yang mendeteksi kemiripan teks dengan memanfaatkan konsep *fingerprint* dokumen [6]. Selain itu ada juga algoritma *Trigram* yang digunakan untuk merepresentasikan teks ke dalam potongan tiga karakter (*n-grams*) yang saling tumpang tindih guna menangkap kemiripan lokal [7], dan algoritma Jaro-Winkler diterapkan karena mampu menghitung kesamaan antara string dan memiliki sensitivitas tinggi terhadap kesamaan awalan string yang relevan untuk pencarian pengguna [8].

Dalam arsitektur baru tersebut, penelitian ini menerapkan pendekatan pencarian berbasis *similarity detection* untuk mengatasi keterbatasan toleransi kesalahan (*fault tolerance*) pada pencarian *substring* akibat kesalahan pengetikan dan variasi input pengguna. Dua pendekatan algoritmik yang dievaluasi adalah metode *Winnowing* serta kombinasi *Trigram* dan Jaro-Winkler, kedua pendekatan ini dipilih untuk mengetahui algoritma mana yang paling efektif dalam mendukung pencarian data pengguna. Apakah teknik *fingerprinting* (*Winnowing*) atau *N-grams* (Kombinasi *Trigram* dan Jaro-Winkler).

Mengingat proses pencarian yang memiliki waktu komputasi yang relatif tinggi, penelitian ini mengintegrasikan Bloom Filter sebagai struktur data probabilistik yang digunakan untuk melakukan penyaringan awal (*early rejection*) terhadap kandidat data yang tidak mungkin ada pada database. Dengan pendekatan ini, hanya data yang berpotensi relevan yang diproses lebih lanjut, sehingga beban komputasi dapat diminimalisir dan waktu respons sistem tetap terjaga.

Untuk memvalidasi efektivitas solusi yang diusulkan, penelitian ini melakukan analisis kuantitatif dengan membandingkan implementasi eksisting dan arsitektur baru menggunakan tiga metrik evaluasi, yaitu *load time*, *computation time*, dan *tolerance level*. Metrik *load time* digunakan untuk mengukur waktu respons sistem terhadap permintaan klien, khususnya untuk mengevaluasi dampak penerapan mekanisme *pagination* pada pemuatan penyajian data. Pengujian ini dilakukan menggunakan Grafana k6 pada lingkungan terkontrol. Sedangkan *computation time* digunakan untuk mengukur waktu komputasi pencarian, khususnya untuk mengevaluasi penggunaan mekanisme *early rejection* menggunakan Bloom Filter pada proses pencarian, terutama pada skenario input yang tidak memiliki kecocokan di dalam database. Selain itu, metrik *tolerance level* digunakan untuk menilai kemampuan toleransi kesalahan (*fault tolerance*) fitur pencarian dalam menangani variasi ejaan dan kesalahan pengetikan dengan membandingkan tiga pendekatan pencarian, yaitu *substring matching*, algoritma *Winnowing*, serta kombinasi *Trigram* dan Jaro–Winkler, di mana hasil pencarian dibandingkan terhadap *ground truth* untuk menentukan keberhasilan kemunculan target dalam daftar hasil pencarian pada batas kandidat tertentu (*Top-10*), kemudian dianalisis berdasarkan tingkat kesalahan input maksimum yang masih dapat ditoleransi oleh masing-masing pendekatan.

1.2 Pertanyaan Penelitian

Berdasarkan uraian latar belakang yang telah disampaikan, perumusan masalah dalam penelitian ini dapat dirangkum ke dalam beberapa poin berikut:

- 1.2.1 **RQ1:** Sejauh mana arsitektur baru yang diusulkan dapat mengoptimalkan *load time* pada proses penyajian data pengguna, berdasarkan hasil pengujian performa menggunakan Grafana k6?
- 1.2.2 **RQ2:** Pendekatan pencarian mana yang memiliki tingkat *fault tolerance* tertinggi dalam menangani variasi ejaan dan kesalahan pengetikan (*typo*) pada input pencarian, jika dibandingkan antara mekanisme pencarian *substring* pada sistem eksisting, metode *Winnowing*, dan kombinasi *Trigram* dan Jaro-Winkler?
- 1.2.3 **RQ3:** Sejauh mana penerapan mekanisme *early rejection* Bloom Filter mampu meningkatkan kecepatan waktu komputasi (*computation time*) pencarian terhadap input pencarian yang tidak terdapat dalam database?

1.3 Batasan Penelitian

Berdasarkan pertanyaan penelitian yang sudah disebutkan, berikut merupakan batasan masalah dari penelitian ini:

- 1.3.1 Penelitian ini hanya berfokus pada perbaikan penyajian data pengguna dan fitur pencarian pada halaman daftar user di panel admin aplikasi Talent Discovery. Fitur lain seperti manajemen tes, soal, paket, atau modul user panel tidak termasuk dalam lingkup penelitian.
- 1.3.2 Pengujian performa difokuskan pada dua aspek utama, yaitu *load time* dan *tolerance level* pencarian. Parameter lain seperti penggunaan memori, CPU server, atau

konsumsi bandwidth tidak menjadi objek analisis utama.

- 1.3.3 Pengujian dilakukan menggunakan Grafana k6 dengan konfigurasi *hardware* dan *software* yang tetap antara *host* dan *client*. Merepresentasikan kondisi trafik nyata, variasi konfigurasi server tidak menjadi fokus penelitian.
- 1.3.4 Dataset yang digunakan berupa data pengguna hasil simulasi atau *dummy* data yang merepresentasikan skala ribuan pengguna. Penelitian tidak mengevaluasi dampak dari data *real-time*, data dinamis, atau perubahan frekuensi *update* data di lingkungan operasional.
- 1.3.5 Penelitian tidak mencakup aspek keamanan (*security*), autentikasi, otorisasi, maupun enkripsi data. Seluruh proses pengujian difokuskan pada performa pencarian dan pemuatan data, bukan pada aspek keamanan aplikasi.
- 1.3.6 Penelitian hanya menilai performa dari sisi *backend* dan batas tertentu dari *frontend* terkait *load time* data pengguna. Aspek desain antarmuka pengguna (UI/UX), pengalaman pengguna (*user experience*), atau respon visual tidak menjadi fokus.
- 1.3.7 Penelitian hanya menguji tiga pencarian, yaitu *substring matching*, *Winnowing*, kombinasi *Trigram* dan Jaro-Winkler. Dan Bloom Filter sebagai mekanisme *early rejection*. Pendekatan algoritma pencarian lain seperti Levenshtein Distance, Damerau-Levenshtein atau pendekatan lainnya tidak dibahas atau dibandingkan.

1.4 Tujuan Penelitian

Berikut beberapa tujuan dari penelitian ini, yaitu:

- 1.4.1 Mengembangkan arsitektur penyajian data pengguna yang lebih efisien pada halaman daftar pengguna di panel admin

Talent Discovery melalui penerapan mekanisme-mekanisme yang relevan.

- 1.4.2 Menentukan pendekatan mana yang paling optimal dalam menangani *fault tolerance* pada proses pencarian pengguna, dengan membandingkan kinerja pencarian *substring matching*, *Winnowing*, dan kombinasi *Trigram* dan Jaro-Winkler.
- 1.4.3 Menerapkan Bloom Filter untuk mempercepat proses *early rejection* terhadap input pencarian yang tidak terdapat dalam database, sehingga mengurangi beban dan waktu pemrosesan pada *backend*.
- 1.4.4 Mengevaluasi performa keseluruhan dari arsitektur baru yang mengintegrasikan *pagination*, *backend search*, *Winnowing*, kombinasi *Trigram* dan Jaro-Winkler, dan Bloom Filter. Serta membandingkannya dengan implementasi eksisting berdasarkan indikator waktu muat (*load time*) dan *tolerance level*.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1.5.1 Manfaat Akademis

- Memberikan kontribusi wawasan terhadap metode pencarian yang toleran terhadap kesalahan ejaan (*typo*) melalui implementasi pencarian *Winnowing*, dan pencarian kombinasi *Trigram* dan Jaro-Winkler.
- Memperluas pemahaman mengenai penggunaan Bloom Filter sebagai mekanisme *early rejection* untuk mengurangi waktu dan beban proses pencarian pada sistem database, sehingga menambah referensi akademik terkait struktur data probabilistik dalam konteks *search optimization*.

- Menyediakan model evaluasi performa untuk mengukur hubungan antara pendekatan algoritmik dan metrik performa aplikasi (*load time*), yang dapat menjadi dasar bagi penelitian selanjutnya.

1.5.2 Manfaat Praktis

- Meningkatkan kecepatan pencarian pengguna pada aplikasi internal MNP dengan menurunkan *load time* secara signifikan melalui optimasi algoritmik.
- Memberikan alternatif metode optimasi yang dapat diterapkan pada aplikasi serupa yang mengalami pertumbuhan jumlah data, tanpa harus melakukan perubahan arsitektur besar.
- Menjadi landasan implementasi fitur pencarian yang lebih toleran terhadap *typo*, meningkatkan kenyamanan pengguna dalam skenario *real-world*.
- Mengurangi proses komputasi yang tidak perlu berkat mekanisme *early rejection* menggunakan Bloom Filter, sehingga meningkatkan responsivitas dan efisiensi sistem pada lingkungan pengembangan maupun produksi.

1.6 Sistematika Penulisan

1. BAB I: Pendahuluan

Pada bab ini, penulis membahas, latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

2. BAB II: Landasan Teori

Pada bab ini, penulis membahas penelitian terdahulu dan tinjauan solusi yang sudah dilakukan pada penelitian sebelumnya yang nantinya dipakai penulis sebagai acuan, dalam mengembangkan sistem.

3. BAB III: Metodologi Perancangan

Pada bab ini, penulis membahas perancangan sistem secara detail, dan metode pengujian untuk menguji sistem.

4. BAB IV: Hasil dan Analisa

Pada bab ini, penulis membahas hasil dan melakukan analisa terhadap kinerja sistem yang telah dibuat.

5. BAB V: Kesimpulan dan Saran

Pada bab ini, akan berisi kesimpulan dan saran dari penulis untuk mengembangkan sistem yang telah dibuat agar dapat bermanfaat bagi civitas akademika lainnya untuk mengembangkan penelitian akademika lain.

