

**LAPORAN TUGAS KELOMPOK  
MATA KULIAH TEMU BALIK INFORMASI**

**Pengembangan Mesin Pencari Game  
Roblox dengan Elasticsearch dan LLM**



**Disusun oleh:**

**Kelompok Ian Matil**

Muhammad Irfan Firmansyah : 2206816102  
Muhammad Raihan Akbar : 2206827674

**FAKULTAS ILMU KOMPUTER  
UNIVERSITAS INDONESIA  
DEPOK  
2025**

# 1 Pendahuluan

## 1.1 Latar Belakang & Tujuan Proyek

Roblox, sebagai platform game online terkemuka dengan jutaan pengalaman interaktif yang dibuat pengguna, menghadapi tantangan signifikan dalam sistem pencarian internalnya untuk menyajikan hasil yang relevan dan tepat sasaran. Proyek "RoFind - Roblox Search Engine" dirancang untuk mengatasi keterbatasan tersebut dengan mengembangkan mesin pencari eksternal yang komprehensif menggunakan teknologi canggih Elasticsearch dan Large Language Model (LLM). Tujuan utama proyek ini adalah:

- Mengumpulkan data game Roblox secara menyeluruh dan terstruktur melalui API publik
- Mengimplementasikan Elasticsearch sebagai backbone untuk indeksasi dan pencarian yang efisien dan skalabel
- Mengintegrasikan LLM untuk memperkaya deskripsi, meningkatkan pemahaman kontekstual kueri, dan personalisasi hasil
- Membangun arsitektur backend API berbasis FastAPI yang responsif dengan frontend yang intuitif dan ramah pengguna

## 2 Kemajuan Proyek yang Telah Dicapai (Estimasi 50%)

### 2.1 Pengumpulan Data (Scraping)

Pengumpulan data game Roblox dilakukan oleh skrip Python (`RobloxScraperBFS.py` dan `RobloxScraperDFS.py`). Proses dimulai dari satu atau beberapa game awal yang dipilih manual. Skrip kemudian melakukan traversal menggunakan algoritma BFS dan DFS untuk menemukan game-game lain yang terhubung.

- **Penemuan Tetangga & Traversal:** Untuk setiap game, API rekomendasi Roblox (`games.roblox.com/v1/games/{gameId}/recommendations`) digunakan untuk mendapatkan daftar game tetangga. Algoritma BFS menjelajahi game-game terdekat, sementara DFS menjelajahi lebih dalam melalui satu rantai rekomendasi.
- **Ekstraksi Metadata:** Detail setiap game unik yang ditemui (nama, deskripsi, genre, kreator, statistik, rating, tanggal) diekstraksi menggunakan API `games.roblox.com/v1/games`.
- **Kontrol Proses:** Mekanisme pengelolaan duplikasi (game yang sudah dikunjungi) dan batasan kedalaman/jumlah game diimplementasikan.
- **Output:** Data game terstruktur disimpan dalam format JSON (misalnya, `roblox_data.json`) untuk indeksasi.

### 2.2 Implementasi Elasticsearch

Infrastruktur Elasticsearch (versi 8.x) telah berhasil di-deploy menggunakan Docker dengan konfigurasi yang dioptimalkan untuk kasus penggunaan mesin pencari game:

- Perancangan skema indeks `roblox_games` dengan mapping yang telah disesuaikan untuk pencarian teks yang lebih akurat dan efisien
- Implementasi analyzer khusus untuk mengoptimalkan indeksasi dan pencarian dalam konteks terminologi game
- Pengembangan mekanisme pencarian dasar dengan dukungan untuk query multifaset dan filter dinamis
- Modul agregasi yang komprehensif untuk memungkinkan filterisasi dan analitik pada metadata game melalui `elasticsearch_utils.py`

## 2.3 Integrasi Large Language Model

Implementasi model AI generatif `meta-llama/Llama-3-8B-Instruct` melalui Hugging Face API telah dikembangkan dengan fokus pada peningkatan pengalaman pencarian:

- Sistem transformasi deskripsi game yang mengkonversi deskripsi teknis singkat menjadi narasi yang lebih informatif, menarik, dan kontekstual
- Pengembangan framework analisis semantik untuk memahami dan memperluas kueri pengguna dengan konteks yang lebih kaya
- Implementasi mekanisme rekomendasi cerdas yang menyarankan kueri alternatif berdasarkan maksud pengguna
- Sistem fallback dan caching untuk menjaga performa dan ketahanan aplikasi saat menghadapi keterbatasan API

## 2.4 Pengembangan Backend API

Backend aplikasi berbasis FastAPI telah dikembangkan dengan arsitektur modular dalam `main.py` yang menyediakan:

- Endpoint GET `/api/search` yang komprehensif untuk pencarian game dengan dukungan parameter kueri, filter, dan pagination
- Endpoint GET `/api/aggregations` untuk memperoleh statistik dan metadata agregat yang mendukung pengembangan filter dinamis
- Endpoint POST `/api/enhance-description` yang mengintegrasikan kapabilitas LLM untuk meningkatkan kualitas deskripsi game
- Endpoint POST `/api/initialize-data` untuk manajemen data dan administrasi indeks Elasticsearch
- Dokumentasi API terintegrasi dengan Swagger UI untuk memudahkan pengujian dan pengembangan frontend

## 3 Rencana Pengembangan Selanjutnya

- **Algoritma Peringkat Lanjutan:** Pengembangan sistem skor multifaktor yang mengintegrasikan relevansi semantik dengan berbagai metrik popularitas dan engagement game untuk meningkatkan akurasi hasil pencarian
- **Sistem Data Dinamis:** Implementasi mekanisme scraping terjadwal berbasis cron untuk memastikan dataset selalu mencerminkan katalog game Roblox terkini
- **Optimasi Integrasi LLM:** Evaluasi performa berbagai model alternatif dan implementasi strategi caching untuk meningkatkan efisiensi dan mengurangi latensi permintaan API
- **Pengembangan Frontend:** Membangun antarmuka pengguna interaktif dan responsif dengan React.js yang terhubung seamless dengan backend API
- **Strategi Deployment:** Penerapan infrastruktur berbasis cloud dengan konfigurasi container untuk memastikan skalabilitas dan ketersediaan tinggi

## 4 Kesimpulan

Proyek "RoFind - Roblox Search Engine" telah mencapai sekitar 50% dari target pengembangan dengan fondasi teknologi yang kokoh meliputi sistem pengumpulan data yang komprehensif, backend dengan Elasticsearch yang dioptimasi, dan integrasi awal LLM yang fungsional. Tahap pengembangan selanjutnya akan berfokus pada penyempurnaan algoritma pencarian, implementasi fitur dinamis, pengembangan antarmuka frontend yang intuitif, dan strategi deployment yang matang untuk menghasilkan mesin pencari game Roblox yang tidak hanya fungsional tetapi juga memberikan nilai tambah signifikan bagi komunitas pengguna platform.