

# **PENJELASAN TUGAS**

## **Penugasan Aibeecara Path AI Engineer**

### **Sistem Rekomendasi Produk E-Commerce Amazon**

Muhammad Farrel Danendra Rachim

#### **I. DESKRIPSI DAN PERMASALAHAN**

##### **A. DESKRIPSI INDUSTRI**

Amazon adalah perusahaan *multi-tech* nasional di Amerika Serikat yang berperan dalam menjual dan menyimpan berbagai *inventory* serta mengurus berbagai hal seperti harga, pengantaran barang, *customer service*, dan pengembalian barang. Salah satu peran Amazon yang utama adalah menjual berbagai macam barang di toko *online*-nya. Terdapat berbagai macam barang yang dijual dengan bermacam-macam kategori sesuai fungsinya, misalnya *smartwatch*, kabel USB, *remote control*, dan lain-lain. Pada industri ini, berbagai barang juga dapat dilihat dan disaring berdasarkan harga, rating produk, dan banyak pengguna yang memberi rating.

Industri *e-commerce* yang didorong oleh korporasi besar seperti Amazon mengalami perkembangan pesat dalam dunia digital ini. Hal ini disebabkan oleh perubahan perilaku konsumen yang lebih memilih belanja *online* karena kemudahan akses dan variasi produk yang tersedia.

##### **B. PERMASALAHAN**

Berikut adalah beberapa permasalahan yang sering dihadapi pengguna ketika berbelanja di situs *e-commerce* seperti Amazon.

1. Pengguna seringkali kesulitan menemukan produk yang sesuai dengan kebutuhan dan preferensinya akibat banyaknya jumlah produk.

2. Banyak pengguna mengalami kesulitan dalam menemukan informasi produk yang spesifik atau dalam menavigasi situs *e-commerce* besar seperti Amazon.
3. Tanpa rekomendasi yang relevan atau bantuan interaktif, pengguna mungkin kesulitan membandingkan produk atau memahami fitur-fitur utama produk.

Oleh karena itu, sistem rekomendasi berbasis *machine learning* menjadi sistem penting untuk menyarankan produk yang sesuai dengan preferensi pengguna pada saat itu juga. Dalam kata lain, ketika pengguna memilih suatu produk tertentu, sistem rekomendasi bertugas untuk menyediakan sekumpulan produk yang sejajar dengan keinginan pengguna, baik dalam aspek nama produk dan kategori maupun dalam aspek harga dan rating produk. Dengan semakin berkembangnya sistem *Artificial Intelligence* (AI) dalam pengembangan teknologi saat ini, model sistem rekomendasi yang semakin akurat dapat dikembangkan lebih lanjut performanya.

Tidak hanya itu, sekarang ini, pengguna juga cenderung memanfaatkan *chatbot* untuk menyelesaikan permasalahan sehari-hari, termasuk mencari produk yang sesuai dengan keinginannya, karena penggunaannya yang interaktif dan efisien. Oleh karena itu, untuk meningkatkan kualitas pengalaman pengguna dalam menggunakan Amazon, saya juga mengusulkan dan mengembangkan fitur *chatbot* yang dispesialisasi untuk menjawab rekomendasi produk yang tersedia di Amazon. Jawaban yang disediakan *chatbot* ini sangat membantu pengguna dalam membandingkan produk-produk yang ada beserta justifikasi mengapa produk-produk tersebut direkomendasikan berdasarkan kualitasnya. berdasarkan kueri pengguna yang dinamis. Hal ini sangat berguna karena sudah banyak *Large Language Model* (LLM) dalam aspek pembangkit jawaban yang menjawab berbagai permasalahan pengguna dalam bentuk kalimat padat, terstruktur, dan jelas. Dengan adanya sistem rekomendasi produk dan *chatbot* ini, pengguna diharapkan memperoleh rekomendasi produk yang relevan, akurat, dan *user-friendly*.

## II. PENDEKATAN MACHINE LEARNING

## A. SISTEM REKOMENDASI

Model untuk sistem rekomendasi menggunakan pendekatan berbasis *representation learning* dengan *embedding-based retrieval*. Sistem rekomendasi tidak menggunakan model *machine learning* (ML) secara eksplisit, namun membuat representasi vektor (*embedding*) dari kueri produk yang diinginkan pengguna dan menghitung *semantic similarity* antara keduanya. Sekumpulan produk dengan selisih *embedding* antara kueri pengguna yang paling dekat akan menjadi prioritas utama dalam daftar rekomendasi.

Model yang digunakan dalam sistem ini adalah Google Gemini API dengan nama “models/text-embedding-004”. Model ini merupakan model *pre-trained* Google yang menghasilkan representasi vektor dari teks (*embedding model*). Untuk kasus ini, dua fitur dari API ini akan digunakan, yakni “retrieval\_document” untuk teks produk dan “retrieval\_query” untuk kueri pengguna.

Berikut adalah tahap-tahap *retrieval* untuk sistem rekomendasi produk Amazon.

1. Setiap produk digabung menjadi teks lengkap beserta masing-masing atributnya, seperti nama produk, deskripsi, harga, dan rating. Teks kemudian diekstraksi menjadi *embedding* oleh Google Gemini API. Akhirnya, *embedding* disimpan di database vektor ChromaDB yang mampu menyimpan vektor secara efisien dan mudah diakses.
2. Kueri pengguna dalam mencari suatu produk ("I want a product similar to {selected\_product}") juga diubah menjadi *embedding*. Kemudian, menggunakan fitur AI, dilakukan pencocokan *embedding* kueri terhadap *embedding* produk di ChromaDB untuk mencari yang paling mirip.
3. Produk-produk dengan *embedding* paling mirip dikembalikan berdasarkan nilai “k” dalam top-k (dalam hal ini top-k=11, tidak termasuk produk yang diinginkan pengguna sehingga k=10).

## B. CHATBOT

Model untuk *chatbot* menggunakan pendekatan berbasis *generative question answering* menggunakan LLM dari Google Gemini API. Tujuan utama *chatbot* ini adalah untuk memberikan respons terstruktur berupa rekomendasi dua produk terbaik berdasarkan pertanyaan pengguna, dengan alasan yang memiliki justifikasi kuat dan dalam format yang mudah dibaca.

Pertama-tama, perlu dilakukan proses *prompt engineering* terhadap LLM Gemini. *Prompt engineering* meyakinkan model bahwa model adalah sebuah sistem rekomendasi produk yang digunakan untuk menyarankan dua produk terbaik berdasarkan produk *request* pengguna. Hal ini dilakukan agar model memberikan jawaban yang terstruktur dan terspesialisasi untuk rekomendasi produk.

Berikut adalah cara kerja *chatbot* rekomendasi produk Amazon.

1. Pengguna bertanya ke *chatbot* mengenai rekomendasi produk, seperti “Apa ada rekomendasi produk mirip seperti [nama produk]?” atau “Rekomendasikan Bluetooth speaker dengan harga sekitar 1100.”
2. Sistem melakukan *retrieval* produk dengan jarak *embedding* paling dekat dengan *query* pengguna.
3. LLM Gemini membangkitkan respons yang terstruktur dan jelas berdasarkan hasil produk yang ada, kueri, dan histori *chat* pengguna.
4. Model LLM menghasilkan respons final yang terstruktur: nama produk, harga, rating, alasan membeli, link produk.

### III. IMPLEMENTASI MODEL DAN APLIKASI

#### A. IMPLEMENTASI BACK-END

Berikut adalah implementasi kode *back-end* seperti pemrosesan data serta sistem rekomendasi dalam bahasa Python.

##### 1. Dataset

[Dataset produk Amazon](#) diperoleh dari hasil *web scraping* situs web resmi Amazon, dengan lebih dari 1000 baris data. Data produk memiliki berbagai atribut yang terdiri atas nama produk, deskripsi, kategori, harga jual, harga diskon, rating, dan link resmi Amazon dari produk tersebut.

Data akan diproses terlebih dahulu sebelum diekstraksi *embedding*-nya. Data yang memiliki *missing value* atau duplikat dihapus dari data, dan kolom kategori di-*parsing* sehingga nilainya menjadi bersih.

## 2. Sistem Rekomendasi

Sistem rekomendasi memanfaatkan kelas GeminiEmbedding yang berfungsi memanggil *task document retrieval* atau *query retrieval*. Hasil pemanggilan ini adalah embedding dari model Gemini. Untuk menginisialisasi database, dokumen yang berisi fitur-fitur produk seperti nama produk, harga, dan rating, ditambahkan ke database ChromaDB untuk dikonversi ke dalam bentuk embedding. Dokumen-dokumen ditambahkan sesuai banyak data produk dengan *batch size* 80.

## 3. Chatbot

Untuk melakukan spesialisasi terhadap fungsi model *chatbot*, model akan dikasih suatu prompt khusus yang memberitahukan bahwa model chat adalah sistem rekomendasi produk yang berfungsi untuk menyarankan tepat dua produk kepada pengguna berdasarkan *request* pengguna. Prompt memastikan bahwa chat model hanya memberikan dua produk agar pengguna tidak perlu bingung membandingkan terlalu banyak pilihan produk. Prompt juga memastikan bahwa model menjawab menggunakan format yang rapi, terstruktur, dan mudah dibaca. Selain itu, rating produk yang disarankan sebaiknya di atas 4 dan alasan yang diberikan sebaiknya dapat meyakinkan pengguna untuk membeli produk tersebut, khususnya berhubungan dengan kualitas produk tersebut.

Untuk menyimpan riwayat *chat* yang dapat ditampilkan di halaman *chatbot* dan sebagai bahan pembelajaran model untuk pertanyaan-pertanyaan selanjutnya, dibuat suatu *dictionary* yang berisi pesan-pesan dari dua buah role: pengguna dan asisten AI.

Model chat dikonfigurasi sedemikian rupa sehingga instruksi sistem menerima *prompt* yang telah dibuat di awal serta memiliki  $temperature = 1.4$  dan  $top\_p = 0.6$ . Untuk pesan pengguna yang pertama, input yang dimasukkan ke model chat berisi *prompt* instruksi sistem, kueri pengguna, dan hasil rekomendasi berdasarkan API sistem rekomendasi sebelumnya. Untuk pesan-pesan selanjutnya, model hanya menerima input *chat* agar dapat mempelajari dari riwayat *chat* untuk memberi output yang sesuai.

## B. IMPLEMENTASI FRONT-END

Sebagai prototipe implementasi sistem rekomendasi, akan digunakan kaskas Streamlit untuk menjadi platform yang mudah dibuat dan digunakan oleh pengguna. Streamlit digunakan karena Streamlit memiliki berbagai fitur UI yang dapat dikustomisasi dengan mudah dan dapat di-*deploy* ke Streamlit Cloud agar dapat diakses secara *online*.

### 1. Sistem Rekomendasi

Pengguna dapat memilih kategori produk dan nama produk yang diinginkan menggunakan menu dropdown. Dari nama produk yang dipilih, akan ditampilkan rekomendasi 10 produk yang paling sesuai berdasarkan jarak *embedding*-nya, beserta harga, rating, dan link produk dalam bentuk tabel. Produk yang direkomendasikan di atas merupakan produk yang paling cocok dengan kueri pengguna berdasarkan sistem rekomendasinya.

### 2. Chatbot

Pengguna dapat berinteraksi dengan asisten AI yang sudah terspesialisasi untuk menjadi *recommender system*. Pengguna tinggal menanyakan produk yang ingin disarankan berdasarkan kategorinya atau harganya, dan asisten AI akan menjawab pertanyaan pengguna berdasarkan hasil sistem rekomendasi. Penampilan jawaban asisten AI akan disimulasikan seperti ChatGPT sehingga huruf-huruf teks muncul satu per satu.

## IV. EVALUASI MODEL DAN EFEKTIVITAS SOLUSI

### A. EVALUASI MODEL

Sistem rekomendasi dan model chatbot dievaluasi performanya menggunakan beberapa metode yang berbeda.

#### 1. Sistem Rekomendasi

Untuk melihat apakah produk yang direkomendasikan relevan dengan produk *request* pengguna atau tidak, akan dihitung *cosine similarity* antara kueri produk dan produk yang direkomendasikan. Semakin besar nilai *cosine similarity*, semakin mirip nama produk dengan kueri pengguna (namun bukan berarti makin relevan). Metrik ini menjadi metrik yang paling efektif untuk mengukur performa sistem rekomendasi tanpa adanya *ground truth* berupa produk-produk yang relevan.

Precision@k adalah metrik yang mengukur produk rekomendasi berdasarkan relevansi. Karena tidak ada *feedback* pengguna mengenai produk mana saja yang relevan, diasumsikan bahwa suatu produk bersifat relevan terhadap kueri tertentu adalah jika skor *cosine similarity*  $\geq 0.7$  dan kedua produk tergolong dalam kategori yang sama.

Setiap kali pengguna memilih suatu produk, di aplikasi Streamlit akan ditampilkan daftar produk rekomendasi yang diurutkan berdasarkan skor *cosine similarity*. Urutan produk rekomendasi berdasarkan skor *cosine similarity* tidak selalu persis sama dengan daftar produk rekomendasi awal. Selain itu, ditampilkan juga skor rata-rata dari seluruh produk yang direkomendasikan, skor minimum, skor maksimum, dan skor precision@k.

Untuk mengukur performa model sistem rekomendasi secara keseluruhan, akan dianalisis relevansi dari kueri produk-produk dalam kategori yang terdiri atas setidaknya 20 produk. Sampel hanya terdiri dari 14 kategori yang memenuhi kriteria di atas. Hal ini dilakukan karena ukuran kategori yang lebih kecil dari 20 cenderung memiliki skor yang lebih buruk dikarenakan dari 10 produk yang direkomendasikan, beberapa produk kurang berhubungan kategorinya atau namanya.

Nilai metrik setiap produk akan dikelompokkan dan dirata-ratakan berdasarkan kategorinya. Kategori “SmartWatches” memiliki rata-rata skor *cosine similarity* tertinggi sebesar 0.8293, kategori “DryIrons” memiliki rata-rata skor precision@k tertinggi sebesar 0.9667, dan kategori “SmartTelevisions” memiliki rata-rata persentase produk yang relevan tertinggi sebesar 0.9717. Rata-rata skor *cosine similarity* untuk seluruh 14 kategori adalah 0.7749 (paling rendah sebesar 0.7327) dan rata-rata precision@k adalah 0.7569.

Untuk produk dalam kategori lain, umumnya produk yang direkomendasikan memiliki nama produk yang mirip, namun berasal dari kategori berbeda. Semakin sedikit produk dalam kategori tersebut, semakin kecil juga skor *similarity* dan precision@k-nya karena nama-nama yang direkomendasikan memiliki jarak *embedding* yang lebih jauh. Contohnya, untuk satu-satunya produk di kategori “EspressoMachines”, produk-produk yang direkomendasikan berasal dari kategori berbeda yang mirip (“StovetopEspressoPots”, “DripCoffeeMachines”, “Juicers”, etc). karena produk yang dimasukkan ke dalam kueri adalah satu-satunya dari kategori ini. [Berikut adalah link yang menyimpan evaluasi untuk beberapa produk dalam kategori yang berbeda beserta observasinya.](#)

## 2. Chatbot

Evaluasi untuk *chatbot* akan dilakukan secara manual untuk memastikan output yang dihasilkan model *chat* sesuai dengan ekspektasi pengguna. Evaluasi dilakukan dengan memasukkan *prompt* “Recommend me similar products to {nama produk}”. Fitur-fitur yang dinilai dari respons model adalah:

- Apakah model menjawab tepat 2 produk: skala 0-1, 15%
- Apakah semua hal dalam instruksi sistem dimasukkan (nama, harga, rating, alasan, dan link): skala 0-4, 25%
- Apakah alasan persuasif, relevan, dan spesifik: skala 0-4, 35%
- Apakah rekomendasi produk relevan terhadap kueri berdasarkan daftar rekomendasi produk dari sistem rekomendasi awal: skala 0-4, 25%

Beberapa produk yang diambil berasal dari 10 kategori terbesar, sedangkan produk lain diambil dari kategori yang lebih kecil untuk melihat varietasnya. Skor rata-rata evaluasi dari total 22 produk adalah 95.3409. [Berikut adalah link evaluasi untuk chatbot.](#)

## B. EFEKTIVITAS SOLUSI

Solusi yang ditawarkan melalui sistem rekomendasi dan *chatbot* terbilang efektif dalam menjawab masalah pengguna dalam memperoleh rekomendasi produk yang sesuai.

### 1. Sistem Rekomendasi

Rata-rata skor *cosine similarity* dan *precision@k* yang tinggi (sekitar 0.75-0.77), beserta observasi untuk produk secara terpisah, menunjukkan bahwa sistem rekomendasi mampu menunjukkan produk-produk yang relevan berdasarkan produk *request* pengguna. Selain itu, metrik ini menunjukkan bahwa kebanyakan produk yang direkomendasikan memiliki merek yang mirip, kategori yang sama, atau rating yang tinggi. Daftar produk yang direkomendasikan juga sudah diurutkan sesuai produk yang paling cocok dengan preferensi pengguna. Hal ini menunjukkan bahwa produk yang direkomendasikan pada umumnya sudah sesuai dengan preferensi pengguna. Dengan sistem rekomendasi ini, pengguna mampu memperoleh rekomendasi yang sesuai dengan efektivitas penyelesaian masalah rekomendasi produk meningkat sebesar setidaknya 75%.



## 2. Chatbot

Berdasarkan evaluasi manual, *chatbot* sangat efektif dalam menjawab pertanyaan mengenai rekomendasi produk dari pengguna dengan jawaban yang mendetail, terstruktur, dan relevan. Dengan *chatbot* ini, pengguna akan memperoleh rekomendasi produk yang cocok dengan rating tinggi, alasan yang jelas dan meyakinkan, serta link terhadap produk yang disebutkan sehingga pengguna langsung dapat membeli produk tersebut. Selain itu, dengan merekomendasikan dua produk paling bagus, pengguna mampu membandingkan produk mana yang lebih cocok dengan preferensinya. Kemudahan pengguna dalam mendapatkan rekomendasi produk Amazon dengan UI yang *user-friendly* dan interaktif meningkatkan penyelesaian masalah interaktivitas rekomendasi produk sebesar setidaknya 85%.