

Terbit online pada laman: <http://jurnal.utu.ac.id/JTI>

Jurnal Teknologi Informasi

ISSN (Online): 2829-8934



Sistem Rekomendasi Pembelajaran dengan Menggunakan Metode *Content-Based Filtering* Berbasis Aplikasi Android

Ade Arya Anggara¹, Abdurrahman Ridho^{2*}Teknologi Informasi, Fakultas Teknik, Universitas Teuku Umar
Jl. Alue Peunyareng, Ujong Tanog Darat, Meurebo, Kabupaten Aceh Barat, Aceh, IndonesiaEmail: ¹adearyaanggara@gmail.com, ²abdurrahman.ridho@utu.ac.id

INFORMASI ARTIKEL

Riwayat Artikel:
Diterima: 25 April 2024
Revisi: 16 Mei 2024
Diterbitkan: 28 Mei 2024

Kata Kunci:
Sistem Rekomendasi
Content-based Filtering
Course

ABSTRAK

Pada dewasa saat ini, menjadi seorang *developer* atau *programmer* merupakan impian dari banyak anak muda saat ini terkhusus di Indonesia sendiri. Namun, banyak dari mereka yang merasa bingung untuk memulai dari mana terlebih dahulu dalam mewujudkan impian tersebut. Oleh karena itu, penelitian ini ditujukan kepada mereka agar dapat membantu dalam menentukan pilihan yang lebih cocok untuk mereka ikuti. Penelitian ini nantinya akan merekomendasikan *course* yang sesuai kepada pengguna. Setelah pengguna selesai mengisi jawaban terkait dengan *course-course* yang disediakan oleh sistem, sistem akan mengkalkulasikan hasil jawaban tersebut guna dijadikan sebagai rekomendasi pembelajaran atau *course* yang sesuai untuk mereka. Sistem rekomendasi ini sendiri dikembangkan dengan menerapkan algoritma atau metode *content-based filtering*. Metode ini merupakan sebuah algoritma yang sering digunakan dalam membangun sebuah sistem untuk merekomendasikan suatu hal. *Dataset* yang digunakan untuk membangun model dengan metode tersebut, diambil dengan menyebarkan kuesioner kepada sekumpulan orang secara acak. Hasil akhir penelitian ini merupakan sebuah aplikasi yang dapat merekomendasikan pembelajaran atau *course* yang dapat digunakan oleh pengguna dalam menentukan pilihannya.

Copyright © 2024 Jurnal Teknologi Informasi UTU
All rights reserved

1. Pendahuluan

Dalam era dewasa ini, banyak generasi muda yang terpesona oleh dunia teknologi, bermimpi untuk meniti karir sebagai *Programmer* atau *Developer*. Mereka terinspirasi oleh kemungkinan tak terbatas yang ditawarkan oleh ranah IT, ingin menyelami dunia *coding* dan pengembangan perangkat lunak. Seperti data dari *World Economic Forum 2023*, kebutuhan untuk menjadi seorang *Data Engineer* meningkat hingga 50% di tahun 2023 ini. Menurut Basuki dalam wawancaranya di Gedung Kominfo “Kita prioritaskan yang di pasar laku, seperti misalnya *design*, pemrograman, dan lain-lain. Itu yang mau didorong dulu karena di pasar diperlukan”. Bagi banyak anak muda, memulai perjalanan menuju keahlian dalam bidang tertentu bisa menjadi tantangan yang membingungkan. Kurangnya pemahaman dalam dunia pemrograman atau IT seringkali membuat mereka merasa kebingungan dan ragu tentang langkah awal yang harus diambil. Sebagian dari mereka mungkin bahkan tidak tahu harus memulai dari mana terlebih dahulu. Dalam keadaan seperti itu, mencari arah yang jelas dan memahami langkah-langkah awal yang diperlukan dapat menjadi kunci untuk memulai perjalanan belajar dalam bidang IT. Oleh karena itu, solusi dalam bentuk sistem rekomendasi ini dirancang dan disempurnakan untuk mengatasi kekurangan pengetahuan tersebut. Tujuannya adalah memberikan panduan yang jelas dan relevan tentang langkah-langkah pertama yang harus diambil atau kursus mana yang sesuai bagi individu yang baru memulai atau masih dalam tahap awal pembelajaran.

Fitrianti menjelaskan bahwa *recommendation system* adalah sebuah aplikasi komputer yang mampu melakukan pencarian dengan memberikan saran atau rekomendasi yang bervariasi, baik berupa informasi, barang, atau layanan, dari sejumlah besar data yang tersedia. Sistem ini bertujuan untuk membantu pengguna dalam menemukan apa yang mereka cari dengan lebih efisien, mengingat kompleksitas dan volume besar informasi yang tersedia di dunia digital saat ini [1], [2]. Biasanya, dalam sistem rekomendasi, terdapat berbagai metode pemfilteran yang digunakan untuk menyesuaikan konten yang ditampilkan kepada pengguna. Metode-metode ini tidak hanya berfungsi untuk menyaring konten yang tidak relevan, tetapi juga untuk memprediksi preferensi pengguna berdasarkan data historis dan perilaku mereka. Dengan demikian, sistem rekomendasi dapat lebih akurat dalam menentukan pilihan yang sesuai dengan kebutuhan dan minat individu pengguna [3]. Sistem ini memiliki kapabilitas untuk menghasilkan prediksi konten atau item yang sesuai dengan interaksi pengguna berdasarkan data atau informasi yang telah diberikan oleh pengguna sebelumnya. Dengan menggunakan data yang telah tersedia, sistem dapat menyajikan rekomendasi yang lebih akurat dan relevan untuk meningkatkan pengalaman pengguna secara signifikan [4].

Sistem akan menganalisis data yang telah diinputkan oleh *user* berupa jawaban atas pertanyaan yang telah ditentukan besaran nilai tiap soal yang akan dijawab oleh *user*. Salah satu cara yang biasa digunakan untuk menentukan sebuah rekomendasi adalah menggunakan metode *content-based filtering*. Metode tersebut merupakan sebuah cara umum dalam membangun *recommendation system* [5]. Metode ini nantinya akan memeriksa hasil jawaban dari *user* dan akan menentukan atau merekomendasikan pembelajaran yang paling cocok [6]. Fokus penelitian ini adalah menciptakan sebuah sistem atau aplikasi yang dapat digunakan oleh masyarakat umum atau mahasiswa dan siswa untuk menemukan alur pembelajaran yang optimal sesuai dengan kebutuhan dan preferensi mereka. Tujuan utamanya adalah memberikan solusi yang praktis bagi mereka yang ingin memperoleh pengetahuan dan keterampilan secara efektif dan efisien, tanpa perlu merasa bingung atau terbebani dengan banyaknya pilihan yang tersedia. Dengan adanya sistem ini, diharapkan bahwa proses belajar akan menjadi lebih terstruktur, menyenangkan, dan menghasilkan pencapaian yang lebih memuaskan bagi pengguna.

Sistem rekomendasi pemfilteran adalah sebuah alat yang dirancang untuk melakukan pencarian atau prediksi dengan mempertimbangkan kesamaan antara item satu dengan yang lain. Tujuannya adalah untuk memberikan rekomendasi yang relevan kepada pengguna berdasarkan preferensi mereka. Dengan menganalisis data tentang perilaku pengguna dan karakteristik *item*, sistem ini dapat menyajikan informasi yang berguna, barang, atau bahkan pilihan dari beragam sumber informasi yang tersedia. Dengan demikian, sistem ini menjadi sebuah alat yang sangat berguna dalam membantu pengguna menemukan apa yang mereka butuhkan atau sukai dalam tenggang waktu yang lebih singkat dan dengan akurasi yang lebih besar [2], [7]. Sistem rekomendasi sendiri memiliki tujuan untuk merekomendasikan suatu produk, *item*, layanan, dan lain sebagainya yang sesuai dengan preferensi dari pengguna. Sistem rekomendasi sendiri telah dikenalkan sejak pertengahan tahun 1990an perihal penyaringan yang kolaboratif [8]. Sistem rekomendasi sendiri biasa disandingkan atau dipasangkan dengan metode *content-based filtering*.

Metode *content-based filtering* bekerja dengan menggunakan beragam informasi yang terdapat dalam setiap *item* atau data, seperti metadata, deskripsi, atau atribut lainnya, untuk menentukan rekomendasi yang paling cocok untuk setiap pengguna [9]. Dengan menganalisis karakteristik dan konten dari *item-item* tersebut, sistem dapat menyusun profil pengguna dan mencocokkannya dengan *item* yang memiliki kesamaan atau relevansi dalam konteks informasi yang disediakan. Ini memungkinkan sistem untuk memberikan rekomendasi yang lebih personal dan akurat, karena didasarkan pada preferensi dan kebutuhan individual dari pengguna. Dengan demikian, *content-based filtering* menjadi salah satu pendekatan yang efektif dalam meningkatkan pengalaman pengguna dan membantu mereka menemukan konten yang paling relevan dan bermanfaat bagi mereka. Metode ini memiliki keunggulan yang signifikan karena pengguna dapat memperoleh informasi yang dianggap relevan untuk mereka. Hal ini terjadi karena konten dari setiap *item* dapat dengan jelas teridentifikasi dari representasinya. Baik atribut maupun deskripsi dari setiap *item* memainkan peran krusial dalam proses penyaringan pada metode ini, memastikan bahwa pengguna mendapatkan hasil yang sesuai dengan kebutuhan dan preferensi mereka secara efektif.

2. Metodologi Penelitian

2.1 Content-based Filtering

Content-based filtering merupakan sebuah metode yang dapat memberikan hasil rekomendasi yang bersifat independen atau hanya kepada pengguna itu saja [4]. Metode ini dapat memberikan sebuah rekomendasi dengan melihat komponen atau kegiatan yang dilakukan sebelumnya oleh *user* [8]. Pemfilteran berbasis konten memiliki ketergantungan pada pilihan pengguna sebelumnya [10]. Metode ini didasarkan oleh preferensi pengguna dan deskripsi *item*. Untuk menunjukkan preferensi pengguna, *item* diberi label dengan kata kunci yang tidak terkait dengan profil pengguna [11]. Metode ini juga memiliki keterbatasan, yaitu hanya merekomendasikan *item* yang memiliki kesamaan dengan preferensi pengguna. Hal ini menyebabkan pengguna tidak dapat menemukan *item* yang unik atau berbeda dari yang biasa mereka pilih [8]. *Cosine similarity* adalah sebuah metode untuk mengukur kesamaan antara dua vektor. Vektor tersebut mewakili garis pada matriks *user-item*. Kesamaan didefinisikan sebagai sudut antara kedua vektor tersebut, dan dapat dilihat pada rumus persamaan 1 (satu) berikut.

$$\text{Sim}(x, y) = \cos(x, y) = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \times \sqrt{\sum_{i=1}^n y_i^2}} \quad (1)$$

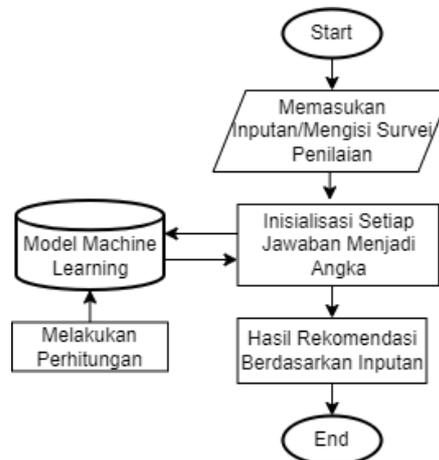
Persamaan $\text{sim}(x, y)$ mengukur kesamaan antara dua vektor x dan y . Kedua vektor tersebut memiliki dimensi n . Beberapa *item* menggunakan TF-IDF untuk pembobotan *item* yang dimasukkan pengguna. Dimana pembobotan pada *item* dapat dilihat pada persamaan 2 (dua) berikut.

$$\text{TFIDF} = \text{weight} = \text{tf} \cdot \log \frac{n}{df} \quad (2)$$

Dalam perhitungan metode *content-based filtering*, *tf* (*term frequency*) menunjukkan berapa kali suatu istilah muncul dalam satu dokumen, n merupakan total jumlah dokumen yang ada, sementara *df* (*document frequency*) merujuk pada jumlah dokumen yang mengandung istilah tertentu.

2.2 Kerja Sistem

Proses dari kerja sistem dapat dilihat pada Gambar 2.1 berikut:

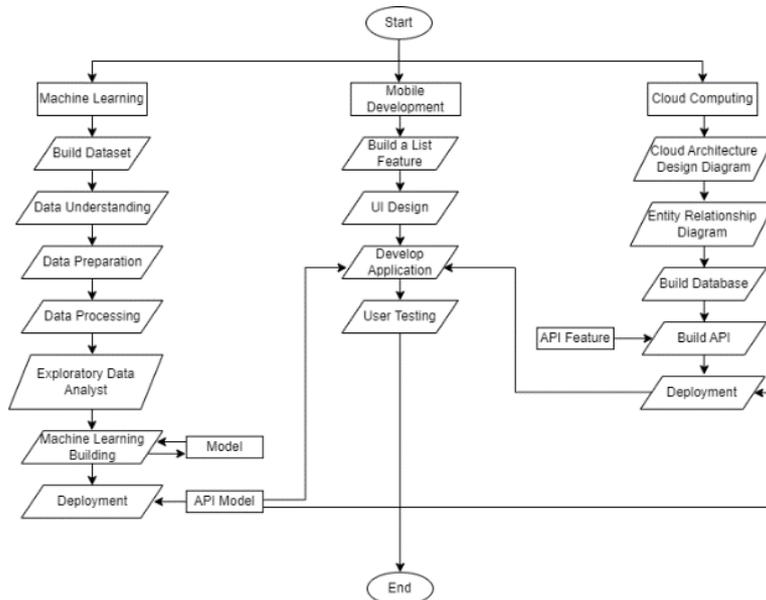


Gambar 2.1 Flowchart Kerja Sistem

Berdasarkan gambar diatas, sitem bekerja berdasarkan inputan atau informasi yang telah *user* berikan. *User* memberikan jawaban atas pertanyaan-pertanyaan yang ditujukan guna dapat diproses dalam memberikan sebuah rekomendasi yang cocok, dalam kasus ini sebuah *course* yang sesuai berdasarkan hasil inputan dari *user*. Setelah selesai mengisikan jawaban dari setiap pertanyaan, proses selanjutnya ialah penginisialisasian yang terjadi didalam model *machine learning*. Dimana setiap jawaban akan di inisialisasikan menjadi sebuah nilai agar dapat dihitung oleh model, dan nilai tertinggi akan di keluarkan menjadi hasil dari rekomendasi *course* yang sesuai untuk *user*.

2.3 Proses Pembuatan Aplikasi Sistem Rekomendasi

Proses selanjutnya ialah pembuatan aplikasi yang dilakukan oleh tiga bidang yang berbeda, dimana ada *Machine Learning*, *Mobile Development*, dan *Cloud Computing*. Dapat dilihat pada Gambar 2.2 untuk masing-masing tugas dalam pembangunan aplikasi sistem rekomendasi ini.



Gambar 2.2 Flowchart Pembuatan Aplikasi

Berdasarkan Gambar 2.2 diatas seorang praktisi machine learning memiliki tanggung jawab besar dalam menciptakan dan mengoptimalkan model-model yang memanfaatkan teknologi *machine learning*, serta mengembangkan API model untuk memfasilitasi integrasi dengan berbagai aplikasi dan sistem. Tugas utamanya melibatkan proses penelitian, pengumpulan data, pemrosesan data, pemilihan model, pelatihan model, evaluasi performa, penyetelan model, dan implementasi API model dengan memperhatikan aspek keamanan, kinerja, dan skalabilitas. Selain itu, mereka juga bertanggung jawab untuk memastikan bahwa model dan API yang dikembangkan dapat diandalkan, responsif, dan sesuai dengan kebutuhan pengguna akhir. API model yang diciptakan harus dapat diintegrasikan secara efisien dalam pengembangan aplikasi mobile dan proses komputasi awan.

Di samping itu, peran seorang ahli komputasi awan (*Cloud Computing*) juga mencakup tanggung jawab terhadap manajemen infrastruktur sistem dan *database* yang beroperasi di GCP (*Google Cloud Platform*), serta perancangan dan implementasi API untuk setiap fitur yang ada dalam aplikasi. Seorang pengembang aplikasi *mobile* bertanggung jawab untuk merancang antarmuka pengguna yang menarik, menciptakan berbagai fitur yang memperkaya pengalaman pengguna, serta mengimplementasikan semua aspek teknis dengan menyatukan berbagai antarmuka pemrograman aplikasi (API) yang telah dikembangkan sebelumnya. Selain itu, mereka juga harus memastikan bahwa aplikasi yang dibangun tidak hanya fungsional tetapi juga responsif dan dapat diandalkan di berbagai platform perangkat *mobile*.

3. Hasil dan Pembahasan

Metode *content-based filtering* dimanfaatkan sepenuhnya dalam pengembangan sistem rekomendasi pembelajaran ini, dengan *TensorFlow* digunakan sebagai alat utama dalam membangun modelnya. Dengan memanfaatkan kekuatan *content-based filtering*, sistem ini dapat memberikan rekomendasi yang lebih tepat dan relevan berdasarkan konten yang sudah ada. *TensorFlow*, sebagai *framework deep learning* yang kuat, memberikan kerangka kerja yang kokoh untuk mengimplementasikan dan melatih model rekomendasi secara efisien dan efektif. Dengan demikian, hasil akhirnya adalah sebuah sistem rekomendasi pembelajaran yang canggih dan berkinerja tinggi, yang memberikan pengalaman pembelajaran yang lebih personal dan memuaskan bagi penggunanya. Aplikasi atau sistem rekomendasi ini mampu menyajikan saran pembelajaran yang sesuai dengan preferensi pengguna berdasarkan fitur-fitur tertentu atau judul-judul yang relevan dengan respons yang diberikan oleh pengguna saat menggunakan aplikasi. Dengan menggunakan algoritma canggih, sistem

ini secara dinamis mengidentifikasi pola-pola dari jawaban pengguna untuk menyusun rekomendasi yang semakin personal dan sesuai dengan kebutuhan belajar masing-masing individu. Dengan demikian, pengguna dapat dengan lebih mudah menemukan materi-materi pembelajaran yang relevan dan menarik bagi mereka, meningkatkan efektivitas dan efisiensi dalam proses pembelajaran mereka. Hasil yang diberikan oleh sistem telah memenuhi ekspektasi yang diharapkan, di mana model mampu menghasilkan prediksi atau rekomendasi yang relevan dengan menggabungkan masukan dari pengguna serta berbagai faktor lainnya.

Cloud Computing juga sangat penting dalam pembangunan sistem ini, karena berguna sebagai penyedia infrastruktur yang *scalable* dan fleksibel sehingga memungkinkan untuk penyimpanan data yang besar dan aksesibilitas yang cepat. Selain itu, dengan menggunakan layanan *cloud*, tim pengembang sistem rekomendasi dapat menghindari investasi awal yang besar dalam infrastruktur fisik, serta memperoleh fleksibilitas dalam menyesuaikan sumber daya komputasi sesuai kebutuhan proyek pengembangan sistem rekomendasi. Dengan memanfaatkan layanan *cloud*, sistem dapat bekerja secara dinamis dan meningkatkan keandalannya dalam proses pemberian rekomendasi kepada *user*.

Sistem ini telah berhasil dalam menghasilkan *output* yang sesuai dengan harapan, menunjukkan kemampuan model dalam memberikan solusi yang tepat berdasarkan respon yang diberikan oleh pengguna. Sistem akan mengubah setiap respons dari pengguna menjadi nilai yang kemudian akan diproses secara kalkulatif untuk menemukan jawaban dengan skor tertinggi, yang selanjutnya akan dipilih sebagai rekomendasi utama. Dapat dilihat pada Gambar 3.1 bentuk dari pertanyaan yang diajukan kepada *user*.

Question Rekomendasi karier

Apa yang Lebih Seru Buat Kamu ?

- Membuat Animasi
- Bermain Dengan Logika
- Menganalisis Data

Apa yang Ingin Kamu Pelajari?

- Cara Membuat Tampilan Website Yang Menarik
- Cara Membuat Database
- Cara Mengelola Data dan Menganalisis Data

Mata Pelajaran Apa Yang Kamu Suka?

- Kesenian dan Kreativitas
- Matematika
- Statistika

Gambar 3.1 Pertanyaan yang Ditunjukkan Untuk *User*

Setelah pengguna menyelesaikan pengisian jawabannya, sistem akan melakukan pemrosesan terhadap jawaban tersebut menggunakan model yang telah disusun sebelumnya. Setelah itu, sistem akan memberikan rekomendasi kursus yang paling cocok kepada pengguna berdasarkan hasil analisis. Pada Gambar 3.2 berikut merupakan potongan *code* dari sistem, dan pada Gambar 3.3 berikut merupakan tampilan hasil rekomendasi dari sistem.

```

import pandas as pd
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder

# Membaca dataset
df = pd.read_csv('data capstone.csv')

# Membagi dataset menjadi fitur (X) dan label (y)
X = df[['desain', 'logika', 'data']]
y = df['hasil']

# Mengonversi label menjadi bentuk numerik
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(y)

# Memisahkan data menjadi data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Normalisasi fitur menggunakan StandardScaler
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Membangun model neural network sederhana
model = tf.keras.Sequential([
    tf.keras.layers.Dense(64, activation='relu', input_shape=(5,)),
    tf.keras.layers.Dense(32, activation='relu'),
    tf.keras.layers.Dense(16, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model.fit(X_train_scaled, y_train, epochs=100, validation_data=(X_test_scaled, y_test))

```

Gambar 3.2 Tampilan Potongan Code dari Sistem Rekomendasi

```

Epoch 6/20 [====] - 0s 21ms/step - loss: 0.9236 - accuracy: 0.8889 - val_loss: 0.8522 - val_accuracy: 0.8889
Epoch 7/20 [====] - 0s 20ms/step - loss: 0.8513 - accuracy: 0.8750 - val_loss: 0.7700 - val_accuracy: 0.8889
Epoch 8/20 [====] - 0s 19ms/step - loss: 0.7832 - accuracy: 0.8750 - val_loss: 0.6899 - val_accuracy: 0.8889
Epoch 9/20 [====] - 0s 22ms/step - loss: 0.7166 - accuracy: 0.8750 - val_loss: 0.6151 - val_accuracy: 0.8889
Epoch 10/20 [====] - 0s 20ms/step - loss: 0.6528 - accuracy: 0.8750 - val_loss: 0.5441 - val_accuracy: 0.8889
Epoch 11/20 [====] - 0s 21ms/step - loss: 0.6004 - accuracy: 0.8750 - val_loss: 0.4763 - val_accuracy: 0.9444
Epoch 12/20 [====] - 0s 21ms/step - loss: 0.5478 - accuracy: 0.8889 - val_loss: 0.4182 - val_accuracy: 0.9444
...
Epoch 19/20 [====] - 0s 19ms/step - loss: 0.3183 - accuracy: 0.9167 - val_loss: 0.1647 - val_accuracy: 1.0000
Epoch 20/20 [====] - 0s 38ms/step - loss: 0.1647 - accuracy: 1.0000

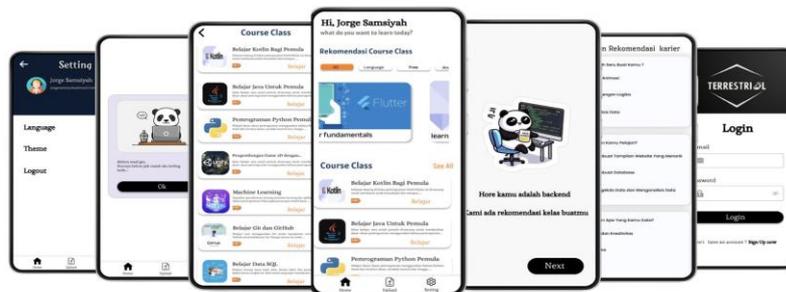
Test accuracy: 1.0
1/1 [====] - ETA: 0s
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
D:\API\TEST\python\lib\site-packages\sklearn\base.py:465: UserWarning: X does not have valid feature names, but StandardScaler was fitted with feature
warnings.warn()
1/1 [====] - 0s 123ms/step

Predicted class: Back2nd

```

Gambar 3.3 Tampilan Hasil Rekomendasi Oleh Code

Berikut penjelasan mengenai tampilan yang ada di dalam aplikasi sistem rekomendasi dapat dilihat pada Gambar 3.4, seperti berikut:

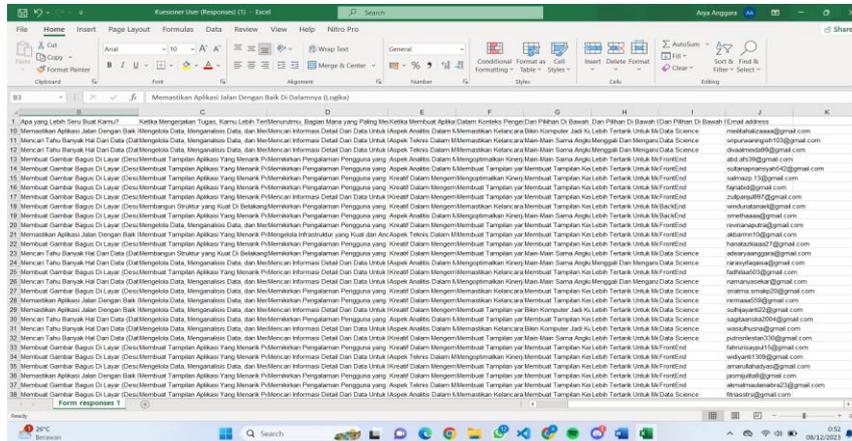


Gambar 3.4 Tampilan Design Antarmuka

Pada Gambar 3.4 diatas merupakan tampilan *design* dari aplikasi sistem rekomendasi pembelajaran. Dimana nantinya pada saat *user* pertama kali membuka aplikasi akan tertampilkan logo dari aplikasinya. Kemudian *user* diarahkan untuk *login* dan *register*, setelah itu sebelum seorang *user* masuk kehalam utama dari aplikasi ia akan diarahkan terlebih dahulu untuk mengisi jawaban dari pertanyaan-pertanyaan yang nantinya akan menjadi data rekomendasinya. Setelah selesai, *user* akan masuk ke halaman utama dimana tertampilkan hasil rekomendasi pembelajaran yang sesuai dengan hasil jawaban pada tahap sebelumnya. Pada halaman utama juga akan tampil *course class* yang mungkin *user* ingin mempelajari materi-materi yang ada pada bagian tersebut. Setelah itu, *user* dapat membuka materi yang direkomendasikan dan materi yang ada pada daftar *course class*. Selanjutnya dari tampilan *design* terakhir yaitu tampilan *setting*. Dimana *user* dapat memperbaharui data profilnya.

3.1 Dataset

Dalam proses pembuatan aplikasi sistem rekomendasi pembelajaran, *dataset* sangat diperlukan untuk dapat melatih model yang akan dibuat nantinya. *Dataset* ini sendiri diperoleh dengan membuat sebuah kuesioner yang nantinya akan ditunjukkan kepada sekumpulan orang secara acak atau random. Dengan adanya *dataset* model diharapkan akan dapat belajar dan berlatih dengan sekumpulan data yang ada untuk menemukan pola dari data tersebut. Dapat dilihat pada Gambar 3.5 tampilan *dataset* yang digunakan untuk melatih model.



Gambar 3.5 Tampilan Data Jawaban Kuesioner

Gambar 3.5 diatas merupakan tampilan dari seluruh jawaban kuesioner yang di tunjukan kepada orang secara acak atau random. Kumpulan jawaban ini masih belum dapat digunakan sebagai *dataset* dari model, karena masih belum sesuai dengan yang akan digunakan dalam pembangunan aplikasi sistem rekomendasi pembelajaran. Kumpulan jawaban ini perlu dilakukan proses *preprocessing data*, agar data sesuai dengan yang dibutuhkan. Dapat dilihat pada Gambar 3.6 tampilan data yang telah dilakukan tahapan *preprocessing data*:

1	desain	logika	data	hasil
2	7	0	0	FrontEnd
3	1	4	2	BackEnd
4	3	1	3	FrontEnd
5	2	4	1	BackEnd
6	3	2	2	Data Science
7	4	2	1	FrontEnd
8	3	4	0	BackEnd
9	4	2	1	FrontEnd
10	1	3	3	Data Science
11	0	2	5	Data Science
12	0	2	5	Data Science
13	4	0	3	FrontEnd
14	6	0	1	FrontEnd
15	6	0	1	FrontEnd
16	7	0	0	FrontEnd
17	6	0	1	FrontEnd

Gambar 3.6 Tampilan Jawaban Kuesioner Setelah Data *Preprocessing*

Pada Gambar 3.6 diatas, data yang terkumpul sudah dapat dijadikan sebagai *dataset* dari model pembuatan aplikasi sistem rekomendasi pembelajaran. Dimana setiap jawaban dari seseorang akan dikalkulasikan dan digabungkan sesuai dengan jenisnya. Sehingga jika seorang *user* mengisikan atau menjawab pertanyaan lebih banyak mengarah ke bidang *FrontEnd* ketimbang bidang yang lainnya yaitu *BackEnd*, dan *Data Science* maka model nantinya akan memberikan rekomendasi *FrontEnd* kepada *user*.

3.2 Proses Kerja Sistem

Sistem rekomendasi pembelajaran ini, akan bekerja dengan memanfaatkan hasil jawaban yang telah *user* berikan pada tahapan setelah *login* kedalam aplikasinya. Kemudian sistem akan menganalisa jawaban yang telah *user* berikan dengan mengkalkulasikan jawaban *user*, dimana sebelumnya setiap pertanyaan yang akan diisikan oleh seorang *user* telah diberikan nilai atau bobot agar sistem dapat menentukan hasil keluaran yang akan diberikan pada seorang *user*. Setelah sistem telah selesai dalam menghitung nilai dari jawaban yang *user* berikan, sistem akan menghasilkan rekomendasi pembelajaran yang paling sesuai berdasarkan hasil jawaban dari pertanyaan tersebut.

4. Kesimpulan

Dengan mengacu pada hasil penelitian yang telah diuji secara ketat, dapat disimpulkan bahwa metode *content-based filtering* pada sistem rekomendasi pembelajaran telah terbukti efektif dalam menyajikan rekomendasi yang cocok dengan keinginan dan preferensi pengguna. Ini menegaskan bahwa pendekatan tersebut bukan hanya berhasil diterapkan, tetapi juga mampu menghasilkan hasil yang relevan dan bermanfaat bagi pengguna. Dengan kehadiran aplikasi sistem rekomendasi pembelajaran tersebut, diharapkan akan mempersembahkan bantuan yang signifikan bagi pengguna, terutama generasi muda, dalam menavigasi ragam pilihan pembelajaran yang ada. Aplikasi ini mengizinkan pengguna untuk menentukan alur pembelajaran yang sesuai dengan preferensi dan kebutuhan mereka berdasarkan jawaban yang mereka berikan, menciptakan pengalaman pembelajaran yang lebih personal dan efektif.

Pada saat ini, sistem hanya akan memberikan *output* atau hasil rekomendasi pembelajaran tentang *Data Science*, *FrontEnd*, dan *BackEnd*. Dimana keterbatasan dalam penggunaan *database* yang ada di *Google Cloud Platform*. Penulis juga mengharapkan, agar semakin berkembangnya sistem rekomendasi pembelajaran ini, dengan lebih banyak lagi hasil rekomendasi yang diberikan. Dimana agar lebih dapat berguna bagi khalayak ramai terutama seseorang yang ingin berkecimpung di dunia IT.

Daftar Pustaka

- [1] B. S. Fitrianti, M. Fachurrozi, and N. Yusliani, "Sistem Rekomendasi Artikel Ilmiah Berbasis Web Menggunakan Content-based Learning dan Collaborative Filtering."
- [2] A. 'Alim, A. Solichin, and P. Painem, "Sistem Rekomendasi Keterampilan Teknologi Informasi Dengan Metode User-Based Collaborative Filtering dan Log-Likelihood Similarity," *CogITO Smart Journal*, vol. 6, no. 2, pp. 141–154, Dec. 2020, doi: 10.31154/COGITO.V6I2.234.141-154.
- [3] N. Zulkarnaim, Musyriyah, Sulfiyanti, Irfan, and Asmawati, "Sistem Rekomendasi Berbasis-Konten Untuk Pengembangan Web Smart Tourism," *Jurnal Komputer Terapan*, vol. 8, no. 1, pp. 143–150, Jun. 2022, doi: 10.35143/JKT.V8I1.5214.
- [4] E. Salim, J. Pragantha, and M. D. Lauro, "Perancangan Sistem Rekomendasi Film menggunakan metode Content-based Filtering".
- [5] S. Lestari and M. M. Ramdhani, "Sistem Rekomendasi Film Menggunakan Metode Content-Based Filtering Studi Kasus Materi Data Mining Di Smk Idn Boarding School," *Jurnal Indonesia: Manajemen Informatika dan Komunikasi*, vol. 4, no. 3, pp. 1581–1587, Sep. 2023, doi: 10.35870/jimik.v4i3.381.
- [6] M. Rizqi and A. Zayyad, "Sistem Rekomendasi Buku Menggunakan Metode Content Based Filtering."
- [7] Z. Munawar *et al.*, "Sistem Rekomendasi Hibrid Menggunakan Algoritma Apriori Mining Asosiasi," *TEMATIK*, vol. 8, no. 1, pp. 84–95, Jun. 2021, doi: 10.38204/TEMATIK.V8I1.567.
- [8] R. H. Mondy and A. Wijayanto, "Recommendation System with Content-Based Filtering Method for Culinary Tourism In Mangan Application."

- [9] A. I. Putra and R. R. Santika, "Edumatic: Jurnal Pendidikan Informatika Implementasi Machine Learning dalam Penentuan Rekomendasi Musik dengan Metode Content-Based Filtering," vol. 4, no. 1, 2020, doi: 10.29408/edumatic.v4i1.2162.
- [10] P. B. Thorat, R. M. Goudar, and S. Barve, "Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System," *Int J Comput Appl*, vol. 110, no. 4, pp. 975–8887, 2015.
- [11] S. Suriati *et al.*, "A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender System," p. 12101, 2018, doi: 10.1088/1742-6596/1000/1/012101.