

BAB III

LANDASAN TEORI

3.1. Sistem Rekomendasi

Sistem rekomendasi adalah sebuah alat dan teknik perangkat lunak yang bisa memberikan saran-saran untuk item yang sekiranya bermanfaat bagi pengguna [1]. Penerapan rekomendasi di dalam sebuah sistem biasanya melakukan prediksi suatu item, seperti rekomendasi film, music, buku, berita dan lain sebagainya yang menarik user. Sistem ini berjalan dengan mengumpulkan data dari user secara langsung maupun tidak [6]. Rekomendasi yang diberikan diharapkan dapat membantu pengguna dalam proses pengambilan keputusan, seperti barang apa yang akan dibeli, buku apa yang akan dibaca, atau musik apa yang akan didengar, dan lainnya [1]. Terdapat beberapa metode yang digunakan dalam membangun rekomendasi diantaranya :

a. *Content Based Recommender System*

Metode *content based* membuat rekomendasi dengan menganalisa deskripsi setiap *item* untuk mengidentifikasi *item* mana yang mempunyai hubungan ketertarikan khusus dari seorang *user*. Deskripsi ketertarikan *user* diperoleh dari profil *user* yang didasarkan atas penilaian menarik atau tidaknya suatu *item* yang ada pada *recommender system* ini [7].

b. *Collaborative Filtering Recommender System*

Pendekatan *collaborative filtering* merekomendasikan *item* kepada *user* berdasarkan korelasi antara *rating user* yang berbeda terhadap *item* atau transaksi *user* di dalam sistem. Sementara *user* merating *item*, sistem akan menggunakan data *rating user* tertentu untuk melakukan perhitungan prediksi dan memberikan rekomendasi *item* yang belum pernah dirating oleh *user* tersebut [8].

c. *Hybrid Recommender System*

Secara umum pendekatan *hybrid recommendations* adalah dengan menggabungkan lebih dari satu metode yang ada pada *recommender system*, kombinasi yang ada pada teknik ini misalnya dengan menggabungkan metode *content based* dengan *collaborative filtering* [9].

3.2. Content-based Recommendation

Sistem rekomendasi berbasis konten (*Content-based Recommendation System*) menggunakan ketersediaan konten (sering juga disebut dengan fitur, atribut atau karakteristik) sebuah item sebagai basis dalam pemberian rekomendasi [1]. Secara umum, metode *content-based filtering* mempunyai 2 teknik umum dalam membuat rekomendasi yaitu *heuristic-based* dan *model-based*. *Cosine similarity*, *Boolean query*, teknik *TF-IDF* (*term frequency-inverse document frequency*) dan *Clustering* termasuk dalam golongan *heuristic-based* sedangkan yang masuk dalam golongan *model-based* adalah teknik *Bayesian classifier & Clustering*, *Decision Tree* dan *Artificial Neural Network* [2].

Metode *content-based filtering*, item direkomendasikan berdasarkan perbandingan antara profil item dan profil pengguna. Profil pengguna adalah konten yang ditemukan relevan dengan pengguna dalam bentuk kata kunci atau fitur. Profil pengguna dapat dilihat sebagai serangkaian kata kunci yang dikumpulkan oleh algoritma dari item yang relevan atau menarik oleh pengguna. Satu set kata kunci dari suatu item adalah profil item. Keuntungan dari metode *content-based filtering* adalah dapat merekomendasikan barang yang tidak diberikan peringkat, dapat dengan mudah menjelaskan cara kerja sistem rekomendasi dengan daftar fitur konten dari suatu item [10].

3.3. Term Frequency Inverse Document Frequency

TF-IDF membantu dalam mengevaluasi pentingnya sebuah kata dalam sebuah dokumen. Dalam kasus ini dokumen yang dimaksud adalah Metode *TF-IDF* merupakan metode untuk menghitung bobot setiap kata yang paling umum digunakan pada *information retrieval*. Metode ini juga terkenal efisien, mudah dan memiliki hasil yang akurat [11].

Metode *TF-IDF* ini menggabungkan 2 konsep yaitu frekuensi kemunculan sebuah kata di dalam sebuah dokumen dan *inverse* frekuensi dokumen yang mengandung kata tersebut. Dalam menghitung bobot dengan metode ini, nilai *TF* per kata masing-masing berbobot 1. Sedangkan nilai *IDF* dihitung dengan rumus :

$$idf(w) = \log\left(\frac{N}{df_t}\right)$$

$idf(w)$ adalah nilai *IDF* dari setiap kata yang akan dicari, N adalah jumlah keseluruhan dokumen yang ada, df adalah jumlah kemunculan kata t pada semua dokumen [12]. Banyak kata yang muncul pada dokumen dalam kasus ini adalah kata yang muncul pada sinopsis cerita. Lalu akan dihitung juga bobot kata untuk aktor, sutradara, rumah produksi, dan genre yang muncul pada data film.

3.4. *Cosine Similarity*

Cosine Similarity dapat diimplementasikan untuk menghitung nilai kemiripan antar kalimat dan menjadi salah satu teknik untuk mengukur kemiripan teks yang populer. Dari hasil matriks algoritma *TF-IDF* akan dilakukan perhitungan untuk menghitung nilai kesamaan antar film. Rumus yang digunakan dalam cosine similarity adalah :

$$\cos(A,B) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{n=1}^j (nA \times nB)}{\sqrt{\sum_{n=1}^j (nA)^2} \times \sqrt{\sum_{n=1}^j (nB)^2}}$$

Keterangan :

$\cos(A,B)$ = Nilai kemiripan antara item x dan item y .

A = vektor A

B = vektor B

nA = jumlah kemunculan kata indeks ke- n dari daftar kata pada kalimat A

nB = jumlah kemunculan kata indeks ke- n dari daftar kata pada kalimat B

Disini A merupakan bobot setiap ciri pada vektor A , dan B merupakan bobot ciri pada vektor B , jika dikaitkan dengan information retrieval maka A adalah bobot istilah pada dokumen A , dan B merupakan bobot setiap istilah pada document B [13].

3.5. *Stopword Removal*

Stopword adalah proses untuk menghilangkan kata-kata yang tidak penting dalam teks seperti *a, an, etc.* Proses ini diperlukan suatu kamus yang menyimpan kata-kata yang bisa dihilangkan atau dengan kata lain adalah kata-kata yang tidak penting [12].

3.6. **Bahasa pemrograman Python**

Python adalah bahasa pemrograman tingkat tinggi yang ditafsirkan, berorientasi objek, dengan semantik dinamis. Pemrograman tingkat tinggi yang dibangun dalam struktur data, dikombinasikan dengan penyetikan dinamis dan pengikatan dinamis, membuatnya sangat menarik untuk pengembangan aplikasi secara cepat, serta digunakan sebagai bahasa scripting untuk menghubungkan komponen yang ada bersama-sama. Sintaksis *Python* yang sederhana dan mudah dipelajari menekankan keterbacaan dan karenanya mengurangi biaya pemeliharaan program.

Python mendukung modul dan paket, yang mendorong modularitas program dan penggunaan kembali kode. *Interpreter* Python dan pustaka standar yang luas tersedia dalam bentuk sumber atau biner tanpa biaya untuk semua platform utama, dan dapat didistribusikan secara bebas (<https://www.python.org/about/apps/>). Python juga dapat dikolaborasi dengan beberapa bahasa pemrograman seperti *Java, C++, Javascript.*

3.7. **Django Framework**

Django framework adalah sebuah kerangka yang didesain untuk membuat aplikasi web dinamis berbasis bahasa pemrograman *Python*. *Django* memiliki kerangka kerja dengan konsep *Model – View – Controller* (MVC). Keunggulan dari framework ini sendiri adalah memiliki panel admin atau halaman pengelolaan admin secara otomatis, memiliki library untuk melakukan otomatisasi proses pembuatan fungsi baru tanpa harus secara manual menulis kode baru, serta dibangun dengan *Object Relation Mapper* (ORM) yang membantu pengembang dalam berinteraksi dengan *database* dan dapat secara otomatis mentransfer data ke dalam *database* (<https://www.djangoproject.com/start/overview/>).