BAB II
TINJAUAN PUSTAKA

2.1. Penelitian Sebelumnya

Beberapa penelitian sebelumnya mengenai sistem rekomendasi telah dilakukan dan banyak diantaranya menggunakan metode yang berbeda. Cara yang biasa digunakan adalah *Content-Based Filtering*. Berikut adalah beberapa studi terkait penerapan metode *Content-Based Filtering* pada Tabel 2.1.

Berdasarkan tabel referensi penelitian menggunakan metode *content-based filtering* dibawah ini, dapat diketahui bahwa metode *content-based filtering* pada sistem rekomendasi memiliki beberapa teknik yang berbeda. Beberapa metode yang umum digunakan adalah dengan menggunakan teknik pembobotan TF-IDF dan *Cosine Similarity* untuk mengukur tingkat kemiripan antar dokumen. Hasil dari beberapa penelitian di bawah menjadi acuan peneliti untuk membuat sebuah sistem rekomendasi dengan menggunakan *content-based filtering*.

Penelitian yang dibuat memiliki perbedaan dengan penelitian terdahulu yaitu bagaimana menentukan rekomendasi pemilihan *software* yang tepat untuk perusahaan. Bagaimana mengimplementasi sistem rekomendasi pemilihan *software* yang tepat berbasis web. Penulis menerapkan Algoritma *cosine similarity* dan metode pembobotan TF-IDF keduanya digunakan dalam sistem yang dikembangkan. Hasil penelitian berupa rekomendasi modul *software* dengan menggunakan metode *content-based filtering* sesuai dengan deskripsi yang pengguna inputkan.

Tabel 2.1 Referensi dan Perbandingan Penelitian Sebelumnya

Referensi dan Perbandingan Penelitian Sebelumnya					
No	**Judul Penelitian**	**Pokok Masalah**	**Tujuan Penelitian**	**Metode**	**Hasil Penelitian**
1	**“**Aplikasi Pendukung Desain Interior dengan Sistem Rekomendasi Berdasarkan Nama Brand Perabot Menggunakan Algoritma Content-Based Filtering Berbasis Web” 2022. [12]	Aplikasi perancangan desain interior dengan 3D Model memiliki kekurangan dalam memberikan panduan ide desain kepada pengguna baru dan tidak terdapat rekomendasi perabot yang dapat digunakan dalam desain berdasarkan selera pengguna [12].	Membantu para pengguna agar menemukan ide desain ruangan yang sesuai dengan kebutuhan dan keinginan [12].	Metode *Content-Based Filtering*	Hasil pengujian dari algoritma ini adalah dengan menggunakan metode confusion matrix yang menunjukkan bahwa nilai *precision* 72%, recall 72%, accuracy 72%, dan error rate 51% [12].
2	“Sistem Rekomendasi Film Menggunakan *Content Based Filtering”* 2021. [13]	Banyaknya film yang diproduksi membuat calon penonton bingung dan kesulitan untuk mencari dan	Mendapatkan informasi yang tepat terhadap film [13].	Metode *Content-Based Filtering*	Sistem dapat memberikan hasil perhitungan mencapai 0,823254 untuk jenis kueri *single* kueri dan 0,7500556

Referensi dan Perbandingan Penelitian Sebelumnya					
No	**Judul Penelitian**	**Pokok Masalah**	**Tujuan Penelitian**	**Metode**	**Hasil Penelitian**
		menentukan film apa yang akan ditonton selanjutnya sehingga menghabiskan waktu lebih banyak dalam mencari film [13].			untuk jenis kueri *multiple seeds* kueri [13].
3	"Rancang Bangun Aplikasi Kursus *Online* Berbasis Web Dengan Sistem Rekomendasi Metode *Content-Based Filtering*" 2022. [14]	Rekomendasi kursus metode *Content- based Filtering* disesuaikan dengan minat siswa karena merekomendasikan kursus berdasarkan riwayat kursus pengguna [14].	Menciptakan aplikasi kursus daring dengan sistem rekomendasi berdasarkan metode *Content-based Filtering* [14].	Metode *Content-Based Filtering*	Aplikasi kursus online yang dapat menyarankan kursus berdasarkan perhitungan nilai kesamaan tertinggi dengan menggunakan algoritma *Cosine similarity* [14].
4	"Sistem Rekomendasi Lagu dengan Metode *Content-Based Filtering*	Keberadaan lagu yang banyak menyebabkan kesulitan bagi banyak orang untuk memilih lagu yang	Untuk mencari rekomendasi lagu [15].	Metode *Content-Based Filtering*	Sistem dapat memberikan keluaran sebagai saran untuk 10 lagu teratas dalam bahasa Indonesia [15].

Referensi dan Perbandingan Penelitian Sebelumnya					
No	**Judul Penelitian**	**Pokok Masalah**	**Tujuan Penelitian**	**Metode**	**Hasil Penelitian**
	Berbasis *Website*" 2021. [15]	ingin didengarkan. Banyak orang merasa bingung dengan jumlah lagu dari berbagai jenis genre yang tersedia [15].			
5	"Rekomendasi Sistem Dengan Metode Content-Based Filtering Untuk Pariwisata Kuliner Pada Aplikasi MANGAN" 2019. [16]	Meskipun menggunakan mesin pencari, tetapi tidak cukup untuk memenuhi kebutuhan pengguna, maka diperlukan sistem rekomendasi yang dapat menyarankan sesuai dengan kebutuhan masing-masing pengguna[16].	Informasi tentang dunia kuliner dapat diperoleh dengan mudah, terutama melalui media elektronik. Namun, banyaknya informasi yang tersedia tidak selalu membuat lebih mudah bagi wisatawan kuliner untuk memilih menu yang tepat [16].	Metode *Content-Based Filtering*	Hasil dari penelitian ini memberikan beberapa saran berupa gambar, nama, dan jarak dari restoran yang sama berdasarkan kesamaan karakteristik konten ketika pengguna memilih restoran tertentu[16].

Referensi dan Perbandingan Penelitian Sebelumnya					
No	**Judul Penelitian**	**Pokok Masalah**	**Tujuan Penelitian**	**Metode**	**Hasil Penelitian**
Penelitian Ini	"Sistem Rekomendasi pemilihan *software* Berbasis *Content-Based Filtering*"	Sangatlah sulit untuk memilih perangkat lunak yang sesuai dan memenuhi kebutuhan perusahaan. Saat ini belum ada sistem online yang dapat memberikan rekomendasi pemilihan *software* terbaik.	Mempermudah perusahaan dalam pengambilan keputusan pemilihan *software* yang tepat. Mengimplementasikan sistem pendukung keputusan metode *content-based filtering* dalam sebuah web.	Metode *Content-Based Filtering*	Hasil dari penelitian ini adalah sistem yang menerapkan metode *content-based filtering* dengan menggunakan teknik pembobotan TF-IDF dan algoritma *cosine similarity*. Sebagai acuan, penulis juga memanfaatkan data dari PT XYZ. Sistem dibuat menggunakan bahasa pemrograman Python dengan *framework* Flask sehingga sistem yang dibuat berbasis *website*.

2.2. Dasar Teori

Melalui berbagai sumber baik buku maupun jurnal, penulis merangkum materi sebagai bahan acuan diantaranya:

2.2.1. Sistem rekomendasi

Sistem rekomendasi adalah aplikasi yang berguna untuk menyarankan dan merekomendasikan item sekaligus menentukan pilihan yang diinginkan pengguna. Pengertian lain dari Sistem rekomendasi yaitu perangkat lunak dan suatu teknik yang memberikan saran sebuah item yang menarik bagi pengguna dan ditunjukan untuk mendukung penggunananya dalam berbagai proses pengambilan keputusan [17].

Sistem rekomendasi adalah suatu sistem yang berguna untuk memberikan saran kepada para pengguna dan bersifat personal, jadi sistem tersebut berbeda-beda bagi setiap masing-masing pengguna [18]. Hal ini bertujuan untuk menarik pengguna dengan memberikan informasi yang berkaitan dengan apa yang disukainya dan memberikan rekomendasi kemungkinan yang pengguna juga sukai. Rekomendasi informasi ini bersifat personal berdasarkan profil pengguna sistem. Profil pengguna biasanya didasarkan pada penilaian minat apakah pengguna telah membaca informasi tertentu atau tidak.

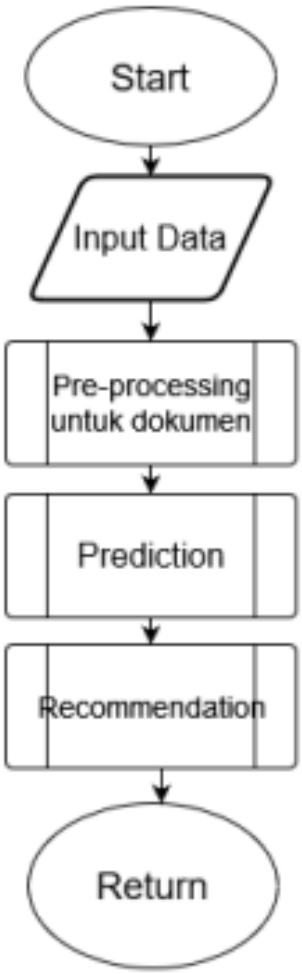
2.2.2. *Content-based filtering*

Rekomendasi produk diberikan kepada pengguna dengan menggunakan metode *Content-Based Filtering (CBF)* yang didasarkan pada deskripsi produk dan preferensi pengguna. Teknik ini menyarankan produk berdasarkan riwayat penggunaan pengguna yang sebanding dengan yang pernah pengguna gunakan sebelumnya. Kelemahan *content-based filtering* ialah batas rekomendasi berdasar kepada *item* yang memiliki kemiripan, sehingga tidak memungkinkan untuk mendapat *item* yang tidak diinginkan. Pengumpulan informasi bisa implisit atau eksplisit [19].

Content-Based Filtering digunakan pada penelitian ini karena cocok dengan dataset yang dimiliki. *Collaborative Based Filtering* kurang tepat digunakan karena membutuhkan data perilaku dari *user* lain. Sistem yang dibuat masih baru, yang

artinya belum memiliki informasi yang cukup, sehingga mengakibatkan hasil yang tidak sesuai. Masalah tersebut disebut sebagai *cold start problem,* yang dapat diminimalisir dengan penggunaan *Content-Based Filtering* [20].

Keunggulan dari *content-based filtering* adalah tidak memerlukan data dari pengguna lain, karena model dibuat sesuai masing-masing preferensi pengguna. Kelemahannya adalah model tetap memberikan rekomendasi berdasarkan kesukaan pengguna yang dapat menimbulkan potensi pembatasan pengguna untuk mengeksplor hal baru yang mungkin disukainya [21]. Dalam diagram alir pada Gambar 2.1 menjelaskan mulai dari proses awal *input*, proses, hingga *output* yang dihasilkan. Secara umum berikut diagram alir dari sistem rekomendasi yang dibuat.



Gambar 2.1 Diagram alur kerja c*ontent-based filtering* [21]

Berdasarkan alur kerja *content-based filtering* diatas, pada awalnya *user* memasukkan data berbentuk deskripsi mengenai aplikasi atau *software* yang *user* inginkan. Setelah data diupload, maka data tersebut masuk ke dalam tahap *pre-processing* data. Dimana pada proses ini, merupakan proses untuk membuat *index*. Tahap selanjutnya dilakukan proses prediction untuk melakukan perhitungan dalam hal ini menggunakan TF-IDF dan *cosine similarity* untuk pembobotan pada setiap *term* pada setiap deskripsi. Tahap terakhir adalah menghasilkan rekomendasi yang sesuai dengan kebutuhan *user*.

2.2.3. *Term frequency – inverse document frequency*

Term Frequency - Inverse Document Frequency atau TF-IDF adalah penerapan statistik numerik yang menunjukkan pentingnya kata kunci untuk laporan tertentu. dengan memberikan kata kunci, file tertentu dapat didiagnosis atau dikelompokkan sesuai dengan signifikansinya. TF-IDF adalah kumpulan dari dua frasa yang berbeda, khususnya TF (*Term Frequency*) dan IDF (*Inverse Document Frequency*). *Term frequency* digunakan untuk mengukur seberapa sering sebuah *term* muncul dalam sebuah dokumen. sebagai contoh, *file* "A" memiliki dua ribu frase dan kata "fakta" terjadi 20 kali dalam laporan. Perhatikan bahwa panjang penuh laporan dapat bervariasi dari sangat kecil hingga sangat besar, jadi istilah tertentu mungkin juga terlihat lebih teratur dalam dokumen besar daripada yang lebih kecil [22]. Untuk mengatasi masalah ini, kejadian periode waktu dalam *file* dibagi melalui berbagai frase dalam catatan untuk menentukan frekuensi kemunculan istilah tersebut. Dalam hal ini frekuensi frase “fakta” dalam *file* “A” adalah TF = 20/2000 = 0,001.

Saat menghitung frekuensi istilah dokumen, dapat dilihat bahwa algoritme memperlakukan semua kata kunci secara setara, terlepas dari apakah kata kunci tersebut mengandung kata berhenti seperti "dari", terlepas dari kenyataan bahwa *Inverse Document Frequency* (IDF) bertujuan untuk mengukur pentingnya kata dalam sebuah dokumen. Setiap kata kunci memiliki arti yang unik. Dikatakan bahwa IDF bertanggung jawab jika kata henti "dari" muncul 2000 kali dalam sebuah dokumen tetapi tidak memiliki banyak arti. Kata-kata yang sering muncul

menerima lebih sedikit bobot dari pembalikan dokumen, sedangkan kata-kata yang jarang muncul menerima bobot lebih [22]. Contoh perhitungan *Inverse Document Frequency* (IDF): Jika memiliki 10 dokumen dan kata "teknologi" muncul di lima dokumen tersebut, maka ukuran IDF = log (10/5) = 0,3010.

Dalam teknik *content-based filtering*, algoritma TF-IDF sering digunakan sebagai teknik untuk mengukur elemen deskripsi teks [23]. Algoritme TF-IDF terkenal efektif, lugas, dan menghasilkan hasil yang presisi [24]. *Term frequency* dihitung menggunakan persamaan (2.1) [25].

$$tf_{i,j} = f_{i,j} \tag{2.1}$$

TF adalah singkatan dari *term frequency*, dan $tf_{i,j}$ menyatakan berapa kali *term* t_i muncul dalam dokumen d_j. Dengan menghitung *instance* dari *term* t_i dalam dokumen d_j, *term frequency* (tf) dapat ditentukan. Selain itu, persamaan tersebut dapat digunakan untuk menentukan frekuensi dokumen terbalik (2.2).

$$idf_i = log\left(\frac{N}{df_i}\right) \tag{2.2}$$

Perhitungan *inverse document frequency* (idf_i) digunakan untuk menentukan jumlah *term* yang dicari (df_i) yang muncul pada dokumen lain, dimana idf_i adalah *inverse document frequency*, N adalah jumlah dokumen yang diambil oleh sistem, dan df_i adalah jumlah dokumen dalm koleksi di mana istilah t_i muncul di dalamnya [25].

Term yang sering muncul dalam dokumen dan jarang muncul dalam dokumen lain adalah *term* yang paling tepat menggambarkan dokumen tersebut, menurut perhitungan bobot *term* dalam dokumen dengan perkalian antara nilai *tf* dan *idf*. Perhitungan *bobot term frequency - inverse document frequency* dapat dilakukan dengan persamaan (2.3).

$$W_{i,j} = tf_{i,j} \,.\, log\left(\frac{N}{df_i}\right) \tag{2.3}$$

2.2.4. *Cosine similarity*

Cosine Similarity adalah metode yang dapat menghitung nilai kemiripan antar kalimat dengan memodelkan dokumen teks sebagai vektor kata (*terms*) [26]. Pada *cosine similarity*, algoritma ini menghitung dot objek. Dot objek sendiri merupakan perhitungan setiap komponen antara dua vektor. Vektor merupakan wujud dari dokumen-dokumen. Untuk menghitung *cosine similarity* dapat dilihat pada persamaan (2,4) dibawah ini:

$$Similarity\ (x, y)\ = \frac{\sum_{i=1}^{n} q_i d_i}{\sqrt{\sum_{i=1}^{n} q_i^2 \cdot \sum_{i=1}^{n} d_i^2}} \tag{2.4}$$

Sumber : Parwita et al., 2018 [27].

Keterangan:

q_i = objek q adalah *query* yang dibandingkan

d_i = objek d adalah dokumen yang dibandingkan

2.2.5. *Precision*

Cara konvensional untuk mengukur kualitas sebuah sistem rekomendasi dalam menanggapi permintaan adalah dengan menggunakan *precision. Precision* merupakan salah satu pengujian dasar dan paling sering digunakan dalam penentuan efektifitas *information retrival system* maupun *recommendation system*. *True positive* (tp) merupakan item relevan yang dihasilkan oleh sistem. *False positive* (fp) merupakan semua item yang dihasilkan oleh sistem. Perhitungan *precision* dapat dilakukan dengan persamaan (2.5) [28].

$$Precision\ = \frac{tp}{tp + fp}\ = \frac{relevant\ item\ retrieved}{retrieved\ item} \tag{2.5}$$

2.2.6. *Website*

Website atau Situs web adalah halaman yang dibuka oleh *browser* menggunakan nama *domain*. Situs web statis dan dinamis adalah dua jenis situs web. Situs web yang statis tidak berkomunikasi dengan sistem secara langsung.

Situs web statis tidak memakai *database* untuk memperlihatkan data. Oleh karena itu, proses update konten *website* statis harus diupdate langsung melalui dokumen *website* tersebut [29]. Sementara, situs web dinamis memerlukan *database* sebab mampu berinteraksi dengan *user*. Saat terjadinya interaksi, halaman pada web dapat terjadi perubahan secara *real time* [30].

2.2.7. UML (*Unified Modeling Language*)

Merancang, memvisualisasikan, dan mendokumentasikan sistem perangkat lunak, adalah tujuan dari UML yang mana dunia industri telah membakukannya [31]. Diagram utama dalam UML yang diterapkan kedalam penelitian ini adalah *use case diagram*, yang dapat mendeskripsikan operasi sistem, termasuk apa yang dilakukannya dan bagaimana aktor (entitas manusia dan mesin) berinteraksi untuk menjalankan tugas dengan sistem, seperti menambahkan data atau membuat laporan.

2.2.8. *Tools*

1. *Visual studio code*

   Microsoft membuat *Visual Studio Code*, juga dikenal sebagai VSCode, yang merupakan *editor* teks yang cepat dan dapat diandalkan untuk berbagai sistem operasi (*multi-platform*), termasuk Linux, Mac, dan Windows. VSCode digunakan untuk mengembangkan aplikasi *mobile, website, cloud* dan *desktop*. *JavaScript, TypeScript, Node.js*, dan bahasa pemrograman lainnya (seperti C++, C#, Python, Go, Java, dll.) didukung langsung oleh aplikasi *editor* teks ini dengan bantuan *plugin* yang dapat diunduh dari *Visual Studio Code Marketplace* [32].

2. *Command prompt*

   Command Prompt adalah aplikasi baris perintah yang tersedia untuk sistem operasi Windows dan di Linux biasanya disebut *root* atau akses administratif penuh. *Command Prompt* digunakan untuk mengeksekusi perintah atau input. Sebagian besar perintah digunakan untuk menjalankan

tugas administratif, memperbaiki masalah Windows tertentu, dan mengotomatiskan tugas menggunakan *skrip* dan *file batch*. *Command Prompt* juga disebut sebagai Windows *Command Processor*, *Command shell*, atau hanya dengan nama filenya, cmd.exe. Semua sistem operasi berbasis Windows NT, termasuk Windows 10, Windows 8, Windows 7, Windows Vista, Windows XP, Windows 2000, dan Windows Server 2012/2008/2003, menyertakan *Command Prompt* sebagai fitur standar [33].

2.2.9. Bahasa pemrograman

Bahasa pemrograman yang digunakan dalam penelitian ini adalah bahasa pemrograman tingkat tinggi, Python, beroperasi pada sistem yang ditafsirkan dan memiliki berbagai kegunaan (tujuan umum). Berlokasi di Stichting Mathematisch Centrum (CWI) di Belanda pada awal 1990-an, Guido van Rossum pertama kali mengembangkan Python. Sebuah bahasa yang memadukan kemampuan, memiliki sintaks kode yang sangat jelas, dan memiliki pustaka fitur yang sangat lengkap dan kaya [34].

2.2.10. *Framework*

1. Flask

   Python dan *Baseband* digunakan untuk membuat *framework* aplikasi web Flask yang ringan, yang didasarkan pada toolkit WSGI (*Web Server Gateway Interface*) dan mesin template Jinja2. *Framework* web berbasis Python, Flask dikategorikan sebagai *microframework*. Setelah mengimpornya ke Python, aplikasi web dapat dibuat dengan cepat menggunakan Flask. Inti tetap dapat diperluas tetapi ini membuat inti tetap sederhana [35].

2. *Bootstrap*

   Bootstrap adalah paket aplikasi siap pakai untuk membuat front-end sebuah website. Bootstrap diciptakan untuk mempermudah proses desain webbagi berbagai tingkat pengguna, mulai dari level pemula hingga yang

sudah berpengalaman [36]. Kerangka situs web *front-end*. *Bootstrap* dapat diunduh dari getBootstrap.com, situs web *Bootstrap* didokumentasikan sepenuhnya dan tersedia juga templat dasar. *Template* dapat di-*copy paste* ke *text editor*, kemudian panggil file *Bootstrap css* pada aplikasi web yang dibuat. Dalam pencarian ini, panggilan *Bootstrap css* dilakukan secara *online*. Jika dijalankan di *browser*, tampilan *website* otomatis langsung menggunakan Bootstrap tanpa mengetik sintaks CSS [37].

2.2.11. *Database*

Basis data yang digunakan dalam penelitian ini adalah MySQL. Karena MySQL adalah tipe data relasional, ia menyimpan informasi dalam bentuk tabel yang terhubung. Kemudahan menyimpan dan menampilkan data dalam bentuk tabel merupakan salah satu keuntungan menyimpan data dalam *database* [38].

2.2.12. Pengujian sistem dan pengembangan

1. Pengujian sistem *blackbox*

   Dikarenakan *blackbox* hanya memerlukan batas bawah dan batas atas untuk data yang diantisipasi, ini mudah digunakan. Aturan masukan juga harus sesuai dengan batasan bawah dan atas untuk memasukkan jumlah data uji yang dapat diestimasi dari jumlah bidang *field input* yang perlu diuji. Dengan menggunakan teknik ini, seseorang dapat menentukan apakah fitur tersebut masih dapat menerima data *input* yang tidak terduga, sehingga menurunkan validitas data yang disimpan [39].

2. Pengembangan sistem metode agile

   Agile *software development methods* atau agile methodologymerupakan sekumpulan metodologi pengembangan perangkat lunakyang berbasis pada pengembangan iteratif, di mana persyaratan dansolusi berkembang melalui kolaborasi antar tim yang terorganisir [40].