

BAB II LANDASAN TEORI

2.1 Sistem Rekomendasi

Sistem rekomendasi membantu pengguna dalam menemukan lokasi yang memuaskan keinginan, sukacita, dan keinginan mereka. Pengguna dapat menggunakan sistem saran untuk membantu mereka menavigasi berbagai item untuk menemukan yang relevan dan bermanfaat. Penggunaan rekomendasi di dalam sistem sering menghasilkan prediksi tentang item, seperti saran film, musik, buku, berita, dan konten lain yang menarik bagi pengguna. Sistem rekomendasi dibagi menjadi dua kategori: *User-Based Collaborative Filtering* dan *Item-to-Item Collaborative Filtering*.

Tujuan dari sistem rekomendasi wisata adalah untuk memberikan rekomendasi yang sesuai untuk destinasi wisata, tempat makan, penginapan, dan kegiatan lainnya berdasarkan preferensi, minat, dan profil wisatawan.

Ada beberapa teknik yang digunakan dalam sistem rekomendasi, di antaranya:

1. *Content-Based Filtering*: Teknik ini menganalisis atribut atau karakteristik item itu sendiri dan mencocokkan *item* dengan profil preferensi pengguna. Misalnya, pada sistem rekomendasi film, konten film seperti *genre*, sutradara, pemain, atau *tagline* digunakan untuk menghasilkan rekomendasi berdasarkan kesesuaian dengan preferensi pengguna.
2. *Collaborative Filtering*: Seperti yang telah dijelaskan sebelumnya, *collaborative filtering* mencari pola atau kesamaan dalam perilaku pengguna atau preferensi mereka. Rekomendasi dihasilkan berdasarkan kerjasama atau kolaborasi antara pengguna dengan preferensi serupa. Ada dua jenis *collaborative filtering*: *user-based* dan *item-based*.
3. *Hybrid Recommendation Systems*: Pendekatan ini menggabungkan beberapa teknik, seperti *content-based filtering* dan *collaborative filtering*, untuk mengatasi kekurangan masing-masing metode dan meningkatkan kualitas rekomendasi.

2.2 Collaborative filtering

Collaborative filtering adalah data difilter tergantung pada karakteristik konsumen untuk memberikan informasi segar kepada pelanggan. Berdasarkan kelompok pola perilaku pengguna, sistem menawarkan informasi. hampir sama. tingkat minat yang berbeda untuk pengguna yang berbeda sumber informasi baru yang mungkin bermanfaat bagi konsumen. Rating tidak tersedia berarti tidak ada informasi. Menghubungkan pengguna ke objek. Ratings dapat dikumpulkan baik secara eksplisit maupun implisit, atau keduanya secara ekspres dan implisit. Penilaian yang diterima ketika klien atau pengguna dihubungi dikenal sebagai eksplisit. Bagikan pemikiran Anda tentang topik ini. Rating yang diperoleh dari aktivitas pelanggan dikenal sebagai rating implisit.

Collaborative filterin dapat dibagi kedalam dua metode yaitu:

2.2.1 Metode Collaborative Filtering User-based

Metode *Collaborative Filtering User-based* merupakan metode pada sistem rekomendasi, *Collaborative Filtering* berbasis pengguna mencari tren dalam cara masing-masing pengguna menilai item sebelum mencoba memprediksi bagaimana pengguna lain akan menilainya menggunakan skor kesamaan pengguna ke pengguna. Kemampuan untuk memantau dan membandingkan perilaku pengguna memungkinkan sistem ini untuk memprediksi tingkah laku pengguna di masa depan atau tingkah lakunya pengguna lain. Ide bahwa sebagian besar orang yang berbagi karakteristik juga menikmati hal yang sama menghasilkan *User-based Collaborative Filtering*. Misalnya, mantan teman bermain Anda mungkin akrab dengan Anda karena mereka menikmati hal yang sama seperti Anda, mungkin karena film atau lagu lama. Tidak mengecualikan potensi bahwa Anda dan teman-teman Anda masih akan menikmati film atau musik yang sama di masa depan.

2.2.2 Metode Collaborative Filtering Item-based

Metode *Collaborative Filtering Item-based* merupakan metode pada sistem rekomendasi, maka pada CF *item-based* parameter yang digunakan adalah hubungan antar item. Hubungan antar item tersebut diperoleh dari item-item

sebelumnya yang telah diberi rating oleh user aktif. Ide dari *Item-based Collaborative Filtering* adalah untuk menemukan pola peringkat untuk satu item dan kemudian mencoba memprediksi peringkat yang diberikan pengguna untuk item lain. Contohnya seperti Husain menyukai makanan “Mie Ayam” dan “Bakso”, namun dia belum mencoba makanan “Sate”. Maka Fabian melihat bahwa Danar dan Erga sudah memberikan peringkat yang sama pada makanan “Sate” dengan dua makanan sebelumnya, maka Fabian menyimpulkan bahwa dia juga akan menyukai makanan “Sate” (Mahendra, 2018).

2.2.3 Euclidean Distance

Pada Jarak peringkat yang diketahui dihitung antara pelanggan dan pelanggan yang ingin direkomendasikan dalam strategi *collaborative filtering*. *Euclidean distance* (jarak *Euclidean*) digunakan untuk mengukur kesamaan atau jarak antara dua item berdasarkan perbedaan nilai rating yang telah diketahui dari pengguna terhadap kedua item tersebut[3]. Tujuan dari penggunaan *Euclidean distance* dalam metode ini adalah untuk menemukan user lama yang mirip atau serupa sehingga dapat digunakan untuk memberikan rekomendasi yang relevan kepada pengguna baru. Berikut adalah persamaan Jarak *Euclidean*:

$$\sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Keterangan :

n adalah jumlah elemen yang kita bandingkan. Dalam konteks perhitungan kemiripan rating.

x_i adalah rating yang diberikan oleh Pengguna A untuk item ke- i .

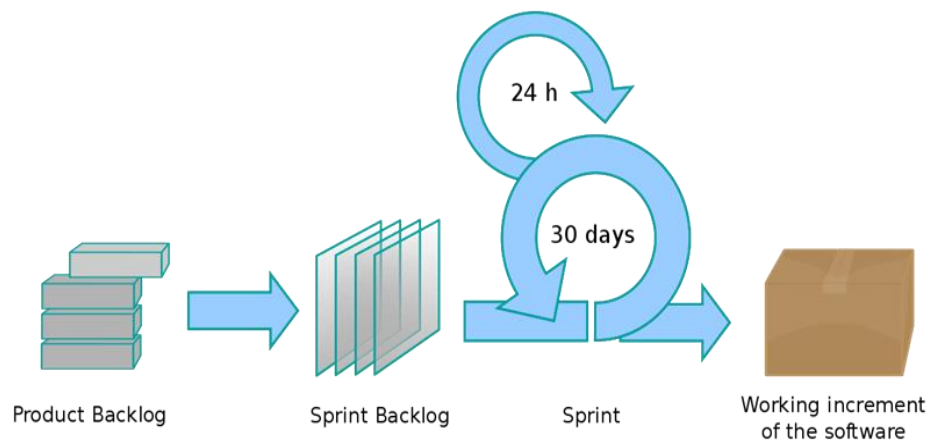
y_i adalah rating yang diberikan oleh Pengguna B untuk item ke- i .

Rumus Similarity :

$$= \frac{1}{1 + (1 + \text{Euclidean Distance})}$$

2.3 Metode Pengembangan Perangkat Lunak *Scrum*

Jeff Sutherland mendirikan *Scrum* pada tahun 1993 dengan niat untuk menciptakan pendekatan pengembangan dan manajemen yang mematuhi ideal *Agile*. *Scrum* adalah kerangka fleksibel untuk manajemen produk, pengembangan aplikasi, dan pengembangan perangkat lunak. *Scrum* memiliki prosedur yang rumit di mana banyak variabel mempengaruhi produk akhir. *Scrum* adalah sistem untuk perencanaan dan kontrol kerja. Berdasarkan serangkaian nilai, prinsip, dan praktik, kerangka kerja *Scrum* menawarkan kerangka struktur yang akan ditambahkan implementasi untuk memenuhi praktik *Scrum*. *Scrum* menggunakan metodologi periodik (*iteratif*) dan bertahap (*incremental*) untuk meningkatkan prediktabilitas dan manajemen risiko.[4]



Gambar 2.1 Metode Scrum

Metode pengembangan perangkat lunak *scrum* juga memiliki kelebihan dan kekurangan yang dapat dilihat pada table 2.1.

Tabel 2.1 Kelebihan dan Kekurangan Metode *Scrum*

Kelebihan	Kekurangan
<i>Scrum</i> dirancang untuk beradaptasi dengan perubahan kebutuhan pelanggan dan persyaratan proyek yang terus berubah.	Membutuhkan Disiplin bisa menjadi kekurangan jika tim tidak memiliki disiplin yang cukup. Jika tidak dikelola dengan baik, perubahan yang

Kelebihan	Kekurangan
	terus menerus dapat menyebabkan ketidakstabilan & ketidakpastian
<i>Scrum</i> mendorong kolaborasi yang kuat dengan pelanggan atau pemangku kepentingan.	Tidak Cocok untuk Proyek yang Sangat Terperinci: <i>Scrum</i> cenderung bekerja lebih baik untuk proyek dengan kebutuhan yang lebih luas dan umum
Pengiriman Produk Secara Berkala agar dapat menghasilkan produk dan memungkinkan tim untuk secara teratur.	Ketergantungan pada Kolaborasi yang Efektif: <i>Scrum</i> memerlukan kolaborasi yang intens antara semua anggota tim dan pemangku kepentingan

Adapun tahapan-tahapan dalam metode *scrum* adalah sebagai berikut :

1. *Product Backlog*

Peneliti akan mengumpulkan semua permintaan pengguna untuk sistem serta persyaratan sistem, seperti yaitu daftar tugas atau fitur yang harus dikerjakan untuk mengembangkan produk. *Product Backlog* dikelola oleh *Product Owner* dan diurutkan berdasarkan prioritas.

2. *Sprint Backlog*

Perencanaan *sprint* dilakukan dalam pertemuan/meeting antara pemilik produk dan tim *developer*, yang akan berkolaborasi untuk memilih product backlog untuk dimasukkan kedalam proses *sprint*. Hasil dari pertemuan tersebut adalah *sprint backlog*.

3. *Sprint*

Tim pengembangan mulai bekerja dalam *Sprint*. *Sprint* adalah periode waktu yang tetap, biasanya berlangsung selama 1bulan, di mana tim fokus untuk menyelesaikan pekerjaan yang telah ditentukan dalam *Sprint Goal*.

4. *Working Increment of The Software*

Increment merupakan hasil dari seluruh hal dalam *product backlog* yang telah selesai dikerjakan pada seluruh *sprint*. Pada akhir *sprint*, *increment* harus sudah benar-benar selesai, yang berarti harus dalam keadaan yang *useable*.

2.4 *Blackbox Testing*

Blackbox testing adalah metode pengujian yang berfokus pada spesifikasi fungsional perangkat lunak. Pengujian yang dilakukan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.





2.5 *Unified Modelling Language (UML)*

Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software* [5].

2.5.1 *Use Case Diagram*

Use case diagram merupakan pemodelan untuk melakukan (*behavior*) sistem informai yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu[5]. Komponen-komponen use case dapat dilihat pada tabel 2.2.



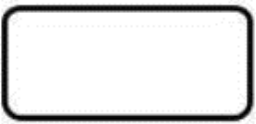
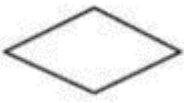

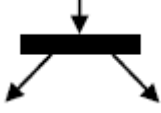

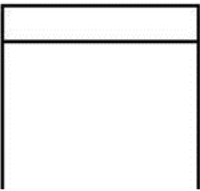
Tabel 2.2 Use Case Diagram

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Tokoh, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi itu sendiri.
2		<i>Include</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (independent).
3		<i>Association</i>	Komunikasi antar aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor
4		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.

2.5.2 Activity Diagram

Diagram ini bersifat statis yang memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Penjelasan mengenai simbol *class* diagram pada tabel. Komponen-komponen activity diagram dapat dilihat pada tabel 2.3.

Tabel 2.3 Activity Diagram






No	Simbol	Nama	Keterangan
1		Status Awal	Status awal dari diagram aktivitas untuk mengawali proses aktivitas sistem.
2		Status Akhir	Status akhir dari sistem yang merupakan akhir dari suatu sistem aktivitas.
3		Aktivitas	Aktivitas yang dilakukan pada sistem biasanya diawali dengan kata kerja.
4		Percabangan	Asosiasi percabangan ada jika pilihan aktivitas lebih dari satu.
5		Penggabungan	Asosiasi penggabungan dilakukan jika ada lebih dari satu aktivitas dan digabungkan menjadi satu.
6		<i>Fork</i>	Menunjukkan aktivitas yang dilakukan secara parallel.
7		<i>Join</i>	Menunjukkan aktivitas yang digabungkan.
8		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.5.3 Class Diagram

Class diagram merupakan gambaran struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class diagram* terdiri dari atribut

dan operasi dengan tujuan pembuat pembuat program dapat membuat hubungan antara dokumentasi perancangan dan perangkat lunak sesuai. [6]. Adapun penjelasan mengenai simbol *class diagram* dapat dilihat pada tabel 2.4.

Tabel 2.4 Activity Diagram

No	Simbol	Nama	Keterangan
1		<i>Class</i>	Kelas pada struktur sistem.
2		<i>Association</i>	Relasi antarkelas dengan makna umum.
3		<i>Generalization</i>	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
4		<i>Depedency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
5		<i>Aggregation</i>	Hubungan antar kelas dengan makna semua-bagian (<i>whole part</i>).

2.6 PHP

PHP atau kependekan dari *Hypertext Preprocessor* merupakan bahasa *script* yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk membuat program situs web dinamis. PHP sering juga digunakan untuk membangun sebuah CMS. PHP adalah bahasa pemrograman *script server-side* yang didesain untuk pengembangan web.

2.7 MySQL

MySQL (*Structured Query Language*) adalah program *database server* yang mampu menerima dan mengirimkan data dengan sangat cepat, *multi user* serta menggunakan perintah standar SQL (*Structured Query Language*) dan baik

digunakan sebagai *client* maupun *server*. *JQueryMobile* adalah proyek baru yang membahas kekurangan jQuery. Ini adalah *framework* yang dibangun di atas *jQuery* yang menyediakan berbagai elemen *user-interface* dan fitur-fitur untuk digunakan dalam aplikasi *mobile*. Mengacu latar belakang di atas dan melihat permasalahan tersebut, maka akan sangat bermanfaat jika ada aplikasi yang dapat memberikan solusi dengan menggunakan *device* yang hampir dimiliki setiap orang, antara lain: handphone, komputer dan laptop.

2.8 Penelitian Terdahulu

Dalam melakukan penelitian ini, peneliti terinspirasi dan mereferensi penelitian-penelitian terkait sesuai dengan masalah yang didapat dari latar belakang skripsi ini. Sebagai bahan pertimbangan, dalam penelitian ini akan dicantumkan beberapa hasil penelitian terdahulu yang berhubungan dengan judul pada penelitian ini yang telah dilakukan oleh beberapa orang. Sehingga penelitian ini lebih akurat, efisien dan efektif.

Tabel 2.5 Penelitian Terdahulu

No	Judul	Algoritma	Metode	Perbedaan Penelitian
1	Item Collaborative filtering untuk rekomendasi pembelian buku secara online [7]	Collaborative Filtering	Item based Collaborative filtering	Penelitian ini menggunakan User Based sebagai metode dan algoritma euclidean distance

No	Judul	Algoritma	Metode	Perbedaan Penelitian
2	Perancangan sistem rekomendasi menggunakan metode collaborative filtering dengan studi kasus perancangan website rekomendasi film[8]	Euclidean Distance	Menggunakan semua metode collaborative filtering	Penelitian ini menggunakan user based sebagai metode dan algoritma euclidean distance
3	Sistem Rekomendasi Produk Ukm Di Kota Bandung Menggunakan Algoritma Collaborative Filtering[1]	Algoritma Collaborative Filtering.	Item based collaborative filtering	Penelitian ini menggunakan user based sebagai metode dan algoritma euclidean distance
4	Pemanfaatan Metode Item Based Collaborative Filtering Untuk Rekomendasi Wisata Di		metode Item-Based Collaborative Filtering	Penelitian ini menggunakan user based sebagai metode dan algoritma euclidean distance.

No	Judul	Algoritma	Metode	Perbedaan Penelitian
	Kabupaten Malang[2]			
5	Aplikasi Rekomendasi Film menggunakan Pendekatan <i>Collaborative Filtering</i> dan <i>Euclidean Distance</i> sebagai ukuran kemiripan rating[3]	Euclidean Distance.	Collaborative Filtering.	Penelitian ini berfokus pada userbased filtering dan wisata.
6	SISTEM REKOMENDASI FILM MENGGUNAKAN METODE <i>USER BASED COLLABORATIVE FILTERING</i> [9]	<i>Cosine Simillarity</i>	User Based Filtering	Penelitian ini menggunakan user based sebagai metode dan algoritma euclidean distance.
7	Sistem Rekomendasi Mata Kuliah Pilihan Menggunakan	Adjusted Cosine Similarity	User Based Collaborative Filtering	Penelitian ini menggunakan user based sebagai metode dan

No	Judul	Algoritma	Metode	Perbedaan Penelitian
	Metode User Based Collaborative Filtering Berbasis Algoritma Adjusted Cosine Similarity[10]			algoritma euclidean distance.
8	Sistem Rekomendasi Film Menggunakan Metode Collaborative Filtering dan K-Nearest Neighbors	K- Nearest Neighbors	Collaborative Filtering	Penelitian ini menggunakan user based sebagai metode dan algoritma euclidean distance.