

## BAB II

### TINJAUAN PUSTAKA

#### II.1. Penelitian Terdahulu

Berdasarkan penelitian yang dilakukan oleh Hartatik, dkk (2021) mengenai Sistem Rekomendasi Wisata Kuliner di Yogyakarta dengan Metode *Item-Based Collaborative Filtering*, Hartatik, dkk menyimpulkan bahwa pengujian MAE, pengujian *Confusion Matrix*, dan pengujian F1 Score. Dari hasil pengujian yang telah dilakukan diperoleh prediksi yang cukup akurat dengan akurasi 83 % (6 neighbor).

Berdasarkan penelitian yang dilakukan oleh Khairani dan Yasir (2018) mengenai penerapan metode *Item Based Collaborative Filtering* pada *e-commerce* sebagai sistem rekomendasi berdasarkan *rating*, Khairani dan Yasir menyimpulkan bahwa dari hasil pengujian, metode *item-based collaborative filtering* dapat memberikan menyusun rekomendasi dengan baik, berdasarkan *item*.

Berdasarkan penelitian yang dilakukan oleh Ramadhan (2017) mengenai pembuatan *web e-commerce* pada toko *kenime store* menggunakan sistem rekomendasi berbasis metode *collaborative filtering* dengan algoritma *adjusted cosine similirity*, Ramadhan menyimpulkan bahwa Sistem memberikan rekomendasi barang kepada pembeli (*user*) sesuai dengan kedekatannya dengan pembeli (*user*) lain yang mirip dalam pola pembelian barang.

Berdasarkan penelitian yang dilakukan oleh Prasetyo, dkk (2019) mengenai implementasi metode *item-based collaborative filtering* dalam pemberian

rekomendasi calon pembeli aksesoris *smartphone*, Prasetyo, dkk menyimpulkan bahwa dalam penelitian ini telah dihasilkan performa dari *Collaborative Filtering* dalam melakukan rekomendasi *item* dengan pertimbangan data yang bersumber pada konsumen yang memiliki kesamaan karakteristik dapat menghasilkan akurasi yang baik dalam penelitian ini telah dihasilkan performa dari *Collaborative Filtering* dalam melakukan rekomendasi *item* dengan pertimbangan data yang bersumber pada konsumen yang memiliki kesamaan karakteristik dapat menghasilkan akurasi yang baik.

Berdasarkan penelitian yang dilakukan oleh Theodorus, dkk (2021) mengenai *Machine Learning* Rekomendasi Produk dalam Penjualan Menggunakan Metode *Item-Based Collaborative Filtering*, Theodorus, dkk menyimpulkan bahwa metode *Item-based Collaborative Filtering* dapat digunakan dalam memberikan rekomendasi produk kepada pelanggan.

Berdasarkan lima penelitian terdahulu, metode *collaborative filtering* dapat digunakan dengan berbagai jenis masalah untuk diselesaikan, oleh karena itu penelitian ini menggunakan metode *collaborative filtering* untuk rekomendasi pembelian obat-obatan dan lima penelitian terdahulu dapat dijadikan referensi untuk penelitian ini.

## **II.2. Landasan Teori**

Berikut ini adalah landasan teori yang berkaitan dengan penelitian yang di ambil dari beberapa jurnal terkait penelitian:

### II.2.1. Implementasi

Implementasi merupakan proses penterjemahan dari pemodelan menjadi bentuk aplikasi sesuai dengan kebutuhan informasi pengguna. Proses implementasi adalah proses penterjemahan dari pemodelan ke dalam pengkodean atau pembentukan antarmuka. (Oktaviani dan Sauda, 2019: 179).

### II.2.2. Metode *Item Based Collaborative Filtering*

*Collaborative filtering* merupakan sebuah metode dalam membuat prediksi dengan cara menyaring informasi item dari opini orang lain. Ide utama dalam sistem rekomendasi *collaborative filtering* adalah untuk memanfaatkan riwayat opini pengguna aktif lain untuk memprediksi item yang mungkin akan disukai/diminati oleh seorang pengguna. Ide dari *item-based collaborative filtering* adalah mencari pola pemberian rating terhadap sebuah item dan kemudian mencoba memprediksi rating yang akan diberikan seorang pengguna terhadap item lain. (Hartatik, dkk, 2021: 56).

Metode rekomendasi yang bersumber pada kemiripan antara pemberi rating terhadap suatu produk dengan produk yang akan dibeli. Dilihat dari tingkat kemiripan produk yang kemudian dibagi dengan parameter kebutuhan pelanggan agar memperoleh nilai kegunaan produk. Produk yang akan dijadikan rekomendasi adalah produk yang memiliki nilai kegunaan tertinggi. Metode rekomendasi ini digunakan sebagai saran untuk beberapa permasalahan yang ada pada *user-based collaborative filtering* yakni skalabilitas dan keterbatasan serta masalah memori dan waktu. (Prasetyo, dkk, 2019: 19).

Langkah-langkah di dalam metode *Item Based Collaborative Filtering* antara lain:

- a. Memberikan penilaian *rating* kepada produk yang dibeli.
- b. Mencari nilai rata-rata *rating*, mencari jumlah nilai *rating* – (rata-rata *rating*) lalu dikuadratkan.
- c. Menghitung *similarity* antar produk.

Berikut ini adalah persamaan yang digunakan untuk menghitung nilai *similarity*:

$$\text{sim}(i, j) = \frac{N}{M} \cdot \frac{\sum_{u \in U} (R_{u,i} - R_i)(R_{u,j} - R_j)}{\sqrt{\sum_{u \in U} (R_{u,i} - R_i)^2} \sqrt{\sum_{u \in U} (R_{u,j} - R_j)^2}} \quad (1)$$

Keterangan:

$\text{sim}(i, j)$  = Nilai kemiripan antara  $i$  dan  $j$

$N$  = Jumlah pengguna yang melakukan *rating* untuk *item*  $i$  dan  $j$

$M$  = Jumlah pengguna yang melakukan *rating* *item*  $i$  atau  $j$

$u \in U$  = Himpunan user  $u$  yang merating *item* $i$  dan *item* $j$ .

$R_{u,i}$  = Rating *user*  $u$  pada *item* $i$ .

$R_i$  = Nilai rata-rata *rating* untuk *item* $i$

$R_{u,j}$  = Rating *user*  $u$  pada *item* $j$ .

$R_j$  = Nilai rata-rata *rating* untuk *item* $j$ .

- d. Menghitung nilai prediksi terhadap *user*

Berikut ini adalah persamaan yang digunakan untuk menghitung nilai prediksi terhadap *user*:

$$P(u, k) = R_k + \frac{\sum_{l=1}^n (R_{u,l} - R_l) \times \text{sim}(k, l)}{\sum_{l=1}^n |\text{sim}(k, l)|} \quad (2)$$

Keterangan:

$P(u,k)$  = Prediksi untuk *user*  $u$  pada *item*  $k$ .

$R(u,k)$  = Rating *user*  $u$  pada *item*  $k$ .

$Sim(k,l)$  = Nilai kemiripan antara *item*  $k$  dan *item* yang mirip. (Prasetyo, dkk, 2019: 19).

### II.2.3. Penjualan

Penjualan merupakan kegiatan pemasaran yang langsung berhubungan dengan konsumen pengguna atau pemakai langsung. (Mawarni, dkk, 2020: 2). “Penjualan merupakan kegiatan yang dilakukan oleh penjual dalam menjual barang atau jasa dengan harapan akan memperoleh laba dari adanya transaksi-transaksi tersebut dan penjualan dapat diartikan sebagai pengalihan atau pemindahan hak kepemilikan atas barang atau jasa dari pihak penjual ke pembeli”. (Arfianto dan Nugrahanti, 2018: 175).

### II.2.4. Semen

Semen merupakan bubuk halus yang bila dicampur dengan air akan menjadi ikatan yang akan mengeras, karena terjadi reaksi kimia sehingga membentuk suatu massa yang kuat dan keras. Reaksi-reaksi tersebut menghasilkan bermacam-macam senyawa kimia yang menyebabkan terjadinya pengerasan. Semen (*portland cement*) didefinisikan sebagai suatu hasil produk yang terdiri dari sebagian besar kalsium silikat yang didapat dari pemanasan hingga meleburnya campuran homogen, bahan yang utamanya berisikan kapur ( $CaO$ ) dan silikat ( $SiO_2$ ) dengan

jumlah kecil alumunia ( $\text{Al}_2\text{O}_3$ ) dan besi oksida ( $\text{Fe}_2\text{O}_3$ ). (Amran dan Permadi, 2021: 159).

#### II.2.5. *Web*

*Web server* merupakan inti dari suatu *website*. *Webserver* berfungsi sebagai pusat kontrol dari pengolahan data dari *website* sehingga setiap instruksi yang diberikan oleh pemakai internet akan diolah dan dikembalikan lagi kepada *user*. Ada beberapa macam *webserver* yang bisa digunakan, di antaranya milik *Windows* seperti *Apache*, *Tomcat*, *IIS*, dan lain sebagainya. Yang paling sering digunakan adalah *Apache* karena lebih banyak mendukung format *file server* tanpa perlu tambahan komponen aplikasi lagi. (Nahlah, dkk, 2017: 12).

*Web server* merupakan *software* yang memberikan layanan data yang berfungsi menerima HTTP atau HTTPS dari klain yang dikenal dengan *browser web* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman *web* yang umumnya berbentuk dokumen HTML. *Web* adalah kumpulan halaman yang digunakan untuk menampilkan informasi teks, gambar, animasi, suara, yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman yang biasa kita sebut *link*. (Mawarni, dkk, 2020: 4).

#### II.2.6. *Android*

*Android* merupakan sistem operasi berbasis *linux* yang menyediakan akses kepada pengguna untuk menggunakan suatu aplikasi. *Android* memiliki *tools* yang

dapat membantu para pengembang pada saat membangun suatu aplikasi. *Android* menyediakan *platform* terbuka bagi pengembang dalam mengembangkan aplikasi *android* mereka. Berikut ini adalah kelebihan-kelebihan dari sistem operasi *android*:

1. *Android* bersifat *open source*, dengan begitu para pengembang memiliki peluang besar untuk dapat membuat dan mengembangkan aplikasi-aplikasi.
2. Terdapat *Google Play Store* yang menyediakan berbagai macam aplikasi dengan fungsinya masing-masing, sehingga pengguna dapat memilih aplikasi-aplikasi apa saja yang ingin digunakan.
3. Setiap aplikasi berbasis *android* juga dikembangkan secara *up to date*, sehingga banyak fitur-fitur baru yang akan muncul ketika memperbaharui suatu aplikasi tersebut. (Wowiling, dkk, 2021: 508).

*Android Software Development Kit* (SDK) adalah sebuah alat untuk para pengembang aplikasi berbasis *Android*. Di dalam *android* SDK terdapat level API yang di dalamnya adalah alat perbaikan kesalahan program, pustaka-pustaka, *emulator*, dokumentasi, kode sumber serta tutorial. Level-level API tersebut adalah inisialisasi dari setiap *platform android* yang berisi paket dalam pengembangan aplikasi *Android*. Dengan adanya *android* SDK ini memudahkan para pengembang untuk menerapkan contoh kode sumber dari contoh aplikasi yang ada pada setiap API. (Maurits, 2020: 22).

### II.2.7. *Hyper Text Markup Language (HTML)*

HTML adalah singkatan dari *Hypertext Markup Language*. HTML memungkinkan seorang *user* untuk membuat dan menyusun bagian paragraf, *heading*, *link* atau tautan, dan *blockquote* untuk halaman *web* dan aplikasi. HTML bukanlah bahasa pemrograman, dan itu berarti HTML tidak punya kemampuan untuk membuat fungsionalitas yang dinamis. Sebagai gantinya, HTML memungkinkan *user* untuk mengorganisir dan memformat dokumen, sama seperti *Microsoft Word*. (Sugijanto, dkk, 2020: 2).

HTML ialah kepanjangan dari *Hypertext Markup Language*. Definisi HTML adalah bahasa yang digunakan untuk menulis halaman *web*. fungsi utama HTML ialah memberi perintah pada *browser* untuk melakukan manipulasi tampilan melalui *tag-tag* yang ditulis dalam HTML. (Rahmasari, 2019: 415).

### II.2.8. *Hypertext Preprocessor (PHP)*

PHP merupakan singkatan dari “*Hypertext Preprocessor*”. PHP adalah sebuah bahasa *scripting* yang terpasang pada HTML. Sebagian besar sintaknya mirip dengan bahasa pemrograman C, Java, ASP dan Perl ditambah beberapa fungsi PHP yang Spesifik dan mudah dimengerti. PHP digunakan untuk membuat tampilan *web* menjadi lebih dinamis, dengan PHP anda bisa menampilkan atau menjalankan beberapa *file* dalam 1 *file* dengan cara di *include* dan *require*. PHP itu sendiri sudah dapat berinteraksi dengan beberapa *database* walaupun dengan kelengkapan yang berbeda yaitu seperti DBM, MySQL, Oracle. (Rahmasari, 2019: 415).

PHP adalah bahasa pemrograman yang sering disisipkan ke dalam HTML. PHP sendiri berasal dari kata *Hypertext Preprocessor*. Sejarah PHP pada awalnya merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari *web*. Bahasa pemrograman ini menggunakan sistem *server-side*. *Server-side programming* adalah jenis bahasa pemrograman yang nantinya *script*/program tersebut akan dijalankan/diproses oleh *server*. Kelebihannya adalah mudah digunakan, sederhana, dan mudah untuk dimengerti dan dipelajari. (Sugijanto, dkk, 2020: 2).

#### II.2.9. MySQL

MySQL merupakan *software* yang tergolong sebagai DBMS (*Database Management System*) yang bersifat *Open Source*. *Open source* menyatakan bahwa *software* ini dilengkapi dengan *source code* (kode yang dipakai untuk membuat MySQL), selain tentu saja bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi, dan bisa diperoleh dengan cara *download* di internet secara gratis. (Nahlah, dkk, 2017: 12).

MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan data dengan cepat dengan menggunakan perintah-perintah SQL. MySQL adalah sebuah program *server database* yang bersifat gratis (MySQL *Community Server*). (Wijaya, 2018: 14).

### II.2.10. *Java*

*Java* merupakan bahasa pemrograman berorientasi objek yang dapat digunakan untuk membuat dan menjalankan perangkat lunak pada komputer dan berbagai *platform*. *Java* memiliki sifat *multiplatform*, yang artinya *java* dapat dijalankan diberbagai jenis *platform* asalkan *Java Virtual Machine* sudah terinstal pada *platform* tersebut. jika program yang dibuat pada *platform window*, maka program tersebut juga dapat dijalankan pada linux, unix, dan lain-lain. (Kasema, dkk, 2018: 2).

### II.2.11. *Unified Modelling Language (UML)*

UML yaitu satu kumpulan konvensi permodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak yang terkait dengan objek. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. (Andikos, 2019: 39).

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

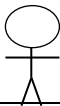
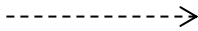
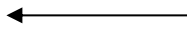
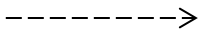
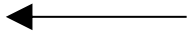
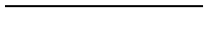
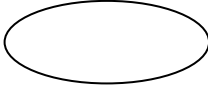
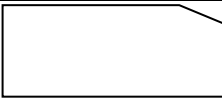
#### 1. *Use Case Diagram*

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor

dengan sistem. *Use case diagram* dapat digambarkan dengan sumber-sumber pada

Tabel II.1.

**Tabel II.1. Simbol Use Case**

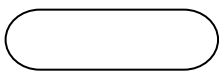
| <b>Gambar</b>                                                                       | <b>Nama</b>           | <b>Keterangan</b>                                                                                                                                                                          |
|-------------------------------------------------------------------------------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <i>Actor</i>          | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .                                                                                        |
|    | <i>Depedency</i>      | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ). |
|    | <i>Generalization</i> | Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).                                        |
|  | <i>Include</i>        | Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.                                                                                                                           |
|  | <i>Extend</i>         | Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.                                                            |
|  | <i>Association</i>    | Apa yang mnghubungkan antara objek satu dengan objek lainnya.                                                                                                                              |
|  | <i>System</i>         | Menspesifikan paket yang menampilkan sistem secara terbatas.                                                                                                                               |
|  | <i>Use Case</i>       | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .                                                               |
|  | <i>Collaboration</i>  | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya ( <i>sinergi</i> ).                                  |
|  | <i>Note</i>           | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi                                                                                              |

(Sumber: Andikos, 2019: 39)

## 2. Activity Diagram

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* dapat digambarkan dengan simbol-simbol seperti pada tabel II.2.

**Tabel II.2. Simbol Activity Diagram**

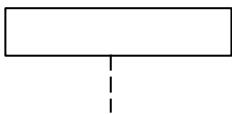
| Gambar                                                                              | Nama                  | Keterangan                                                                                 |
|-------------------------------------------------------------------------------------|-----------------------|--------------------------------------------------------------------------------------------|
|    | <i>Activity</i>       | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain. |
|   | <i>Action</i>         | State dari sistem yang mencerminkan eksekusi dari suatu aksi.                              |
|  | <i>Initial Node</i>   | Bagaimana objek dibentuk atau diawali.                                                     |
|  | <i>Activity Final</i> | Bagaimana objek dibentuk dan dihancurkan                                                   |
|  | <i>Fork Node</i>      | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran                       |

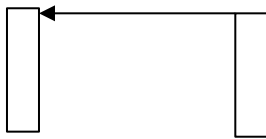
(Sumber: Andikos, 2019: 39)

## 3. Sequence Diagram

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.3.

**Tabel II.3. Simbol Sequence Diagram**

| Gambar                                                                              | Nama            | Keterangan                                        |
|-------------------------------------------------------------------------------------|-----------------|---------------------------------------------------|
|  | <i>Lifeline</i> | Objek entity, antarmuka yang saling berinteraksi. |

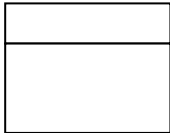
|                                                                                   |                |                                                                                                         |
|-----------------------------------------------------------------------------------|----------------|---------------------------------------------------------------------------------------------------------|
|  | <i>Message</i> | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi. |
|  | <i>Message</i> | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi. |

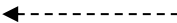
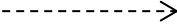
(Sumber: Andikos, 2019: 39)

#### 4. *Class Diagram* (Diagram Kelas)

*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class diagram* dapat digambarkan dengan simbol-simbol seperti pada Tabel II.4.

**Tabel II.4. *Class Diagram***

| Gambar                                                                              | Nama                    | Keterangan                                                                                                                                          |
|-------------------------------------------------------------------------------------|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <i>Generalization</i>   | Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ). |
|  | <i>Nary Association</i> | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.                                                                                         |
|  | <i>Class</i>            | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.                                                                             |
|  | <i>Collaboration</i>    | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.                                |

|                                                                                   |                    |                                                                                                                                                                    |
|-----------------------------------------------------------------------------------|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <i>Realization</i> | Operasi yang benar-benar dilakukan oleh suatu objek.                                                                                                               |
|  | <i>Depedency</i>   | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri. |
|  | <i>Assocation</i>  | Apa yang menghubungkan antara objek satu dengan objek lainnya                                                                                                      |

(Sumber: Andikos, 2019: 39)

**Tabel II.5. Multiplicity Class Diagram**

| <b>Multiplicity</b> | <b>Penjelasan</b>                 |
|---------------------|-----------------------------------|
| 1                   | Satu dan hanya satu               |
| 0..*                | Boleh tidak ada atau 1 atau lebih |
| 1..*                | 1 atau lebih                      |
| 0..1                | Boleh tidak ada, maksimal 1       |
| n..n                | Batasan antara.                   |

(Sumber: Andikos, 2019: 39)