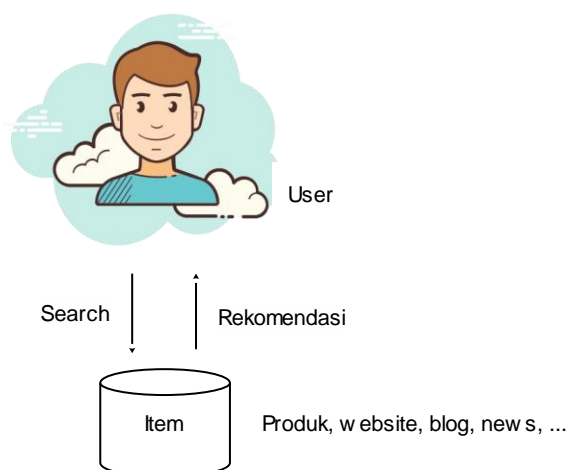


BAB II LANDASAN TEORI

2.1 Sistem Rekomendasi

Sistem rekomendasi didefinisikan sebagai sebuah strategi pengambilan keputusan bagi pengguna di bawah lingkungan informasi yang kompleks. Sistem rekomendasi berfungsi untuk menangani kerumitan penelusuran informasi/ penggalian pengetahuan yang biasanya dihadapi oleh para pengguna dengan merekomendasikan atau menyarankan konten maupun layanan eksklusif yang relevan dengan karakteristik atau kebutuhan setiap pengguna (*personalized content and service recommendations*) (Siregar dkk. 2021).



Gambar 2.1 Ilustrasi Sistem Rekomendasi

Beberapa manfaat dari sistem rekomendasi adalah sebagai berikut :

- Meningkatkan penjualan/pengunjung (*increased sales/conversion*). Sistem rekomendasi berfungsi sebagai layanan marketing otomatis dan berkesinambungan sangat efektif untuk meningkatkan penjualan atau pengunjung.
- Meningkatkan kepuasan pengguna (*increased user satisfaction*). Sistem rekomendasi memandu pengguna menemukan produk/layanan yang relevan

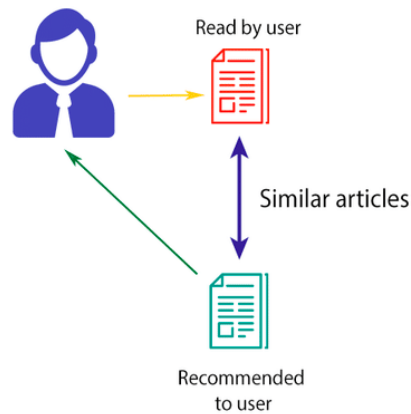
- sehingga mengurangi usaha/kerumitan yang dihadapi pengguna. Hal ini tentunya memberikan kenyamanan dan meningkatkan kepuasan pengguna.
- c. Meningkatkan loyalitas dan berbagi ide (*increased loyalty and share of mind*). Sistem rekomendasi menjadikan para pengguna lebih betah mengunjungi *website/platform* digital yang disediakan sehingga mereka semakin familiar dengan produk/merek yang ditawarkan maupun antar muka *platform*. Hal ini tentunya akan menumbuhkan loyalitas para pelanggan/pengguna.
 - d. Mengurangi risiko ditinggalkan pelanggan/pelanggan beralih (*reduced churn*). Sistem rekomendasi secara berkesinambungan memberikan notifikasi atau rekomendasi (diskon, kupon belanja, dan lainnya) melalui *e-mail*/fasilitas komunikasi lainnya, akan membangun keterikatan/hubungan emosional dengan para pelanggan. Hal tersebut akan mengurangi potensi pelanggan beralih ke perusahaan/penyedia layanan lain.

2.2 Content-Based Filtering

Kualitas rekomendasi yang dihasilkan sangat dipengaruhi oleh ketepatan teknik rekomendasi (*recommendation techniques*) yang digunakan. Pemahaman tentang karakteristik dan keunggulan yang dimiliki oleh setiap teknik rekomendasi sangat diperlukan. Salah satunya adalah dengan menggunakan teknik *filtering content-based filtering*.

Content-based filtering menggunakan informasi tambahan terkait pengguna maupun item. Misalkan pada *movie recommender system*, informasi tambahan terkait pengguna (usia, jenis kelamin, pekerjaan, dll) yang biasanya diperoleh saat pengguna melakukan pendaftaran (*subscribe*). Sedangkan informasi tambahan untuk *item* film adalah aktor/aktris utama, durasi, genre, dll. Sistem rekomendasi dengan teknik *content-based filtering* memberikan rekomendasi dengan menemukan kesamaan antar *item* berdasarkan konteks atau deskripsi dari *item-item* tersebut.

CONTENT-BASED FILTERING



Gambar 2.2 Teknik *Content-Based Filtering*

Pada intinya, *content-based filtering* memahami dan membangun pengetahuan tentang kemiripan (*similarity*) antar item. Sistem akan merekomendasikan *item* yang memiliki kesamaan dengan item yang sebelumnya pelanggan tersebut pernah tonton (film), beli (produk), atau berinteraksi (layanan). Sebagai contoh jika seorang pengguna (si A) menyukai film *Mission Impossible* maka sistem akan merekomendasikan film lain yang diperankan oleh aktor Tom Cruise atau film yang *bergenre Action* kepada si A (Siregar dkk. 2021).

2.3 Preprocessing

Preprocessing adalah proses pengubahan bentuk data yang tidak terstruktur menjadi data yang terstruktur agar sesuai kebutuhan untuk diproses selanjutnya. Data yang telah melewati *preprocessing* adalah data yang lebih terstruktur. Ada beberapa tahap *preprocessing* yang dilakukan yakni *case folding*, *tokenisasi*, *eliminasi*, *filtering* dan *stemming*. Penjelasan mengenai tahap-tahap tersebut adalah sebagai berikut :

- a. *Case Folding* adalah tahap mengubah semua huruf dalam dokumen menjadi huruf kecil.
- b. *Tokenisasi* adalah tahap pemecahan kalimat berdasarkan tiap kata yang menyusunnya.

- c. Eliminasi diterapkan untuk menghapus duplikasi kata. Kata yang duplikat diasumsikan memiliki fitur yang sama, sehingga jika dalam satu *array* terdapat beberapa data yang sama, hanya akan disimpan satu saja.
- d. *Filtering* adalah tahap mengambil kata-kata penting dari hasil tahap sebelumnya. *Filtering* dapat dilakukan dengan menghilangkan *stoplist/stopword* (kata-kata yang tidak deskriptif, seperti kata “yang” dan “dari”) atau kata yang termasuk dalam *wordlist*.
- e. *Stemming* adalah tahap transformasi suatu kata menjadi kata dasarnya (*root word*).

2.4 Cosine Similarity

Cosine similarity merupakan perhitungan yang sering digunakan untuk menghitung kemiripan diantara item-item. Secara umum, fungsi *similarity* adalah fungsi yang menerima dua buah obyek berupa bilangan riil (0 dan 1) dan mengembalikan nilai kemiripan (*similarity*) antara kedua obyek tersebut berupa bilangan riil. *Cosine similarity* merupakan salah satu metode pengukuran kemiripan yang populer. Metode ini digunakan untuk menghitung nilai kosinus sudut antara dua vektor dan biasanya digunakan untuk mengukur kemiripan antara dua dokumen. Fungsi *cosine similarity* antara item A dan item B ditunjukkan sebagai berikut (Mondi dan Wijayanto 2019) :

$$sim(A, B) = \frac{n(A \cap B)}{\sqrt{n(A) n(B)}} \dots\dots\dots(1)$$

Keterangan :

- sim(A, B)* nilai similaritas dari item A dan item B
- n(A)* banyaknya fitur konten item A
- n(B)* banyaknya fitur konten item B
- n(A ∩ B)* banyaknya fitur konten yang terdapat pada item A dan juga terdapat pada item B

Jika kedua objek memiliki nilai similaritas 1, maka kedua objek dikatakan identik dan sebaliknya. Semakin besar hasil dari fungsi similarity, maka kedua objek yang dievaluasi dianggap semakin mirip dan sebaliknya.

2.5 Top-N Recommendation

Kandidat item-item yang didapat dari perhitungan similaritas digunakan sebagai rekomendasi untuk pengguna. Item-item tersebut diseleksi menjadi hanya N item dengan nilai similaritas yang paling tinggi. Item serupa yang memiliki nilai similaritas tinggi diasumsikan cenderung lebih dipilih oleh pengguna. N item tersebut kemudian ditampilkan sebagai daftar item yang direkomendasikan untuk pengguna.

2.6 Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti *smartphone* dan komputer tablet. *Android* pada awalnya dikembangkan oleh Android, Inc., dengan dukungan keuangan dari *Google*, yang kemudian membelinya pada tahun 2005. Antarmuka pengguna *Android* umumnya merupakan manipulasi langsung, menggunakan gerakan sentuh yang mirip dengan tindakan nyata, seperti menggesek, mengetuk, dan mencubit untuk memanipulasi objek di layar, seperti serta *keyboard* virtual untuk menulis teks. *Android* adalah sistem operasi sumber terbuka, dan *Google* merilis kodenya di bawah lisensi Apache. Kode dengan *open source* dan lisensi di *Android* memungkinkan perangkat lunak untuk menjadi dimodifikasi dan didistribusikan secara bebas oleh pembuat perangkat, operator nirkabel, dan aplikasi pengembang. Selain itu, *Android* memiliki sejumlah besar Komunitas pengembang aplikasi (*apps*) yang memperluas fungsionalitas perangkat, umumnya ditulis dalam versi yang disesuaikan dari bahasa pemrograman *Java* (Karim dan Agarina 2019).

Android adalah sebuah sistem operasi perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware*, dan aplikasi. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. *Android* bisa berjalan di beberapa macam perangkat dari banyak produsen yang berbeda. *Android* menyertakan *kit development* perangkat lunak untuk penulisan

kode asli dan perakitan modul perangkat lunak untuk membuat aplikasi bagi pengguna *Android*. *Android* juga menyediakan pasar untuk mendistribusikan aplikasi. secara keseluruhan, *Android* menyatakan ekosistem untuk aplikasi seluler (Karman, Mulyono, dan Martadinata 2019).

Sistem operasi *Android* memiliki kelebihan antara lain *switching* dan *multitasking* yang lebih baik. *Android* sangat mendukung *multitasking* aplikasi sehingga pengguna dapat dengan mudah berpindah aplikasi hanya dengan menyentuh sebuah *icon* pada *system bar* (Zaini dan Rosandy 2020)

2.7 Bahasa Pemrograman dan Basis Data

2.7.1 Kotlin

Kotlin adalah bahasa pemrograman modern melalui pengetikan statis yang digunakan lebih dari 60% *developer Android* profesional untuk membantu meningkatkan produktivitas, kepuasan *developer*, dan keamanan kode. Sejak diluncurkan pada pagelaran *Google I/O* pada bulan Mei 2017, *Kotlin* menjadi bahasa kelas utama (*first class language*) dalam pembuatan aplikasi *Android*.

Kotlin adalah sebuah bahasa pemrograman dengan *Statically typed* (tipe statis) yang berjalan pada platform *Java Virtual Machine* (JVM). *Kotlin* menggunakan compiler LLVM yang artinya, dapat dikompilasi ke dalam kode *JavaScript*. Pengembang utamanya berasal dari tim programmer *JetBrains* yang bermarkas di Rusia. Bahasa pemrograman yang satu ini banyak diminati oleh para *developer*. *Kotlin* merupakan bahasa yang powerful tentu cocok bagi *developer* dalam membuat aplikasi *Android*. Karena *kotlin* adalah bahasa nomor satu untuk pengembangan aplikasi *Android*. Adapun kelebihan *Kotlin* adalah (Aljundi dan Akbar 2018) :

- a. *Concise* : *Kotlin* mampu mengurangi *boilerplate of code* atau tingkat kerumitan dari kode yang biasa kita tulis, ketika menggunakan bahasa *Java*.
- b. *Safe* : *Kotlin* mampu menjamin bahwa setiap *syntax* yang kita tulis secara proses kompilasi dapat mencegah kemungkinan terjadinya *error*, misalnya mampu mencegah terjadinya *NullPointerException* ketika kita *coding* menggunakan bahasa *Java*.

- c. *Versatile* : *Kotlin* sejatinya sama seperti *Java*, karena memang *kotlin* itu sendiri di turunkan dari bahasa induknya, yaitu *Java*. Sehingga *kotlin* juga dapat di pakai dalam pengembangan aplikasi di *Web* maupun *Mobile*.
- d. *Interoperable* : *Kotlin* tidak sama seperti bahasa *Java* turunan lainnya (misal; *Scala* ataupun *Clojure*) yang tidak dapat dijalankan bersamaan dengan kode yang kita tulis menggunakan *Java*. *Kotlin* mampu membaca kode lama atau *library* yang kita gunakan atau kita tulis dengan bahasa *Java* dan begitupun sebaliknya.

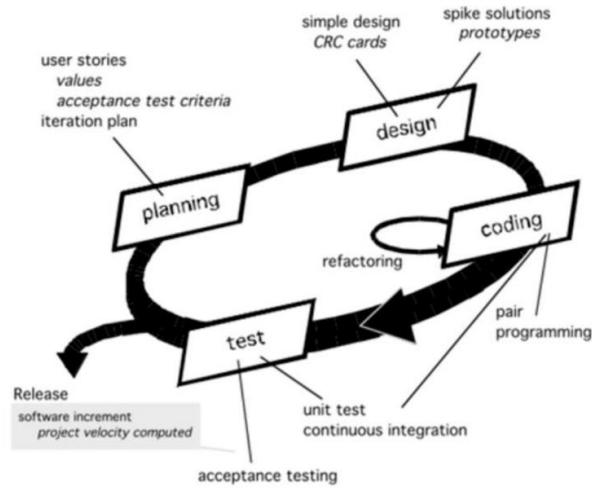
2.7.2 MySQL

Pengelolaan DBMS (*Database Management System*) yang digunakan dalam pembuatan sistem informasi administrasi desa ini adalah *MySQL*. *MySQL* adalah pengembangan lanjutan dari proyek UNIREG yang dikerjakan oleh *Michael Monty Widenius* dan *TcX* (perusahaan perangkat lunak asal Swedia). *MySQL* adalah RDBMS yang *open source* dengan dua bentuk lisensi, yaitu *free software* (perangkat lunak bebas) dan *shareware* (perangkat lunak berpemilik yang penggunaannya terbatas) Dimana setiap orang bebas menggunakan *MySQL*, namun tidak boleh dijadikan produk turunan yang dijadikan *closed source* atau komersial (Indrawan 2021). Jadi, *MySQL* adalah *database server* yang gratis dengan lisensi GNU *General Public License* (GPL) sehingga dapat dipakai untuk keperluan pribadi atau komersial tanpa harus membayar lisensi yang ada (Fitri 2020). *MySQL* merupakan *server* yang melayani *database*, kita dapat mempelajari pemrograman khusus yang disebut *query* (perintah) *SQL*. *MySQL* merupakan *database engine* atau *server database* yang mendukung bahasa *SQL* sebagai bahasa interaktif dalam mengelola data. *MySQL* adalah sebuah perangkat lunak sistem manajemen basis data *SQL* atau DBMS yang *multithread*, *multi-user* (Enterprise 2018).

2.8 Extreme Programming

Salah satu metodologi pengembangan perangkat lunak yang dapat digunakan adalah *Extreme Programming*. *Extreme Programming* mengarah ke pendekatan berorientasi objek dan sesuai digunakan ketika adanya *requirements* yang kurang dipahami maupun terjadinya perubahan *requirements* yang cepat (Sanjaya dan

Andry 2019). Terdapat empat tahapan yang harus dikerjakan pada metode *Extreme Programming* (XP) yaitu perencanaan, perancangan, *coding*, dan *testing*.




Gambar 2.3 Pemodelan *Extreme Programming*



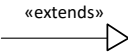

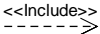
2.9 Pemodelan Sistem

2.9.1 Use Case Diagram

Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi 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. Penamaan pada *use case* simpel dan mudah dipahami (Rosa dan Shalahuddin 2018). Simbol dari *use case* diagram terlihat pada tabel 2.1.

Tabel 2.1 Simbol *Use Case Diagram*


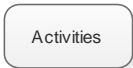
Keterangan	Simbol	Deskripsi
<i>Use Case</i>		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal-awal frase nama <i>use case</i>


Keterangan	Simbol	Deskripsi
Aktor		Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar itu sendiri.
Asosiasi		Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
Ekstensi		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> , dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan.
Generalisasi		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
<i>Include</i>		<i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan


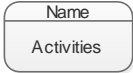

2.9.2 Activity Diagram

Activity diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis. Simbol yang terdapat pada *activity diagram* adalah seperti pada tabel 2.2 (Rosa dan Shalahuddin 2018).

Tabel 2.2 Simbol *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.

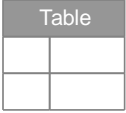
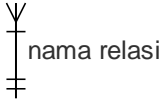
Simbol	Deskripsi
Percabangan 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.

Penggabungan 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

2.9.3 Conceptual Data Model (CDM)

CDM atau model konsep data merupakan konsep yang berkaitan dengan pandangan pemakai terhadap data yang disimpan dalam basis data. CDM dibuat dalam bentuk tabel-tabel tanpa tipe data yang menggambarkan relasi antar tabel untuk keperluan implementasi ke basis data (Rosa dan Shalahuddin 2018). Adapun simbol-simbol yang ada pada CDM adalah seperti pada tabel 2.3.

Tabel 2.3 Simbol CDM

No	Simbol	Deskripsi
1	Entitas/Tabel 	Entitas atau tabel yang menyimpan data pada basis data
2	Relasi 	Relasi antar tabel yang terdiri atas nama relasi dan <i>multiplicity</i>

2.10 *Black Box Testing*

Black box testing merupakan pengujian untuk mengetahui fungsi perangkat lunak yang telah berjalan sesuai dengan kebutuhannya (Jurnal (ABDI) dan Rahardja 2020). Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan pengeluaran perangkat lunak sesuai dengan spesifikasi yang dibutuhkan (Rosa dan Shalahuddin 2018).

Pengujian kotak hitam (*black box*) juga disebut pengujian perilaku yang berfokus pada persyaratan fungsional perangkat lunak, artinya teknik pengujian *black box* memungkinkan untuk membuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan semua kebutuhan fungsional untuk program. *Black box testing* merupakan pendekatan pelengkap yang mungkin dilakukan untuk mengungkap kelas kesalahan yang berbeda dari *white box testing*. *Black box testing* berupaya untuk menemukan kesalahan dalam kategori berikut (Pressman 2012):

- a. Fungsi yang salah atau hilang
- b. Kesalahan antarmuka
- c. Kesalahan dalam struktur data atau akses basis data eksternal
- d. Kesalahan perilaku atau kinerja
- e. Kesalahan instalasi dan penghentian

2.11 *Penelitian Terdahulu*

Penelitian terdahulu yang menunjang dibuatnya sistem pencarian *babershop* dengan penerapan metode *content-based filtering* berbasis Android pada penelitian ini adalah sebagai berikut :

- a. Pada penelitian dengan judul “Penerapan Metode Content Based Filtering Dalam Implementasi Sistem Rekomendasi Tanaman Pangan” merupakan sistem rekomendasi tanaman pangan yang menerapkan teknik *content-based filtering* untuk memberikan hasil rekomendasi tanaman pangan. Kesimpulan yang didapati dari pembuatan sistem ini adalah pada sistem rekomendasi tanaman pangan membuat pemasaran hasil pertanian khususnya kepada pedagang akan lebih efektif dan efisien, artinya sistem mampu memberikan rekomendasi lahan yang sesuai untuk pengguna (Nastiti 2019).

- b. Pada penelitian dengan judul “Penerapan Sistem Rekomendasi *Content Based Filtering* Menggunakan Algoritma Apriori Pada *Mobile E-Commerce* Berbasis Android Studi Kasus (Swaramas Music Shop)” merupakan sistem rekomendasi yang dapat membantu pelanggan untuk memilih produk yang ingin dibeli karena produk yang bervariasi dan meningkatkan penjualan dengan adanya rekomendasi produk alat musik di Swaramas Music Shop (Yusuf 2020).
- c. Pada penelitian dengan judul “Aplikasi Pencarian Barbershop Berbasis Android” merupakan sistem pencarian *babershop* yang dapat digunakan pada media Android dengan teknik *web view*. Terdapat sistem *tracking* yang digunakan sebagai penunjuk jalan menuju lokasi *babershop* (Swara dan Zirwan 2018).

