

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terkait**

Pada penelitian ini menggali informasi dari penelitian-penelitian sebelumnya sebagai bahan perbandingan, baik mengenai metode yang digunakan dan kesimpulan yang ada. Selain itu, juga menggali dari buku, skripsi dan jurnal dalam rangka mendapatkan suatu informasi yang ada sebelumnya tentang teori tentang judul yang digunakan untuk memperoleh landasan teori ilmiah.

(Tirtana et al., 2020) melakukan penelitian mengenai Rancang Bangun Aplikasi *E-Commerce* Guna Meningkatkan Pendapatan UMKM melakukan perancangan sistem perdagangan elektronik (e-commerce) diberi nama Swadagang untuk mendukung efisiensi dan efektivitas pemasaran dari UMKM yang berada di wilayah Kabupaten Malang. Efisiensi didukung dari sisi optimalisasi biaya pemasaran, biaya sumber daya manusia, maupun biaya overhead. Dan efektivitas ditunjang dengan dukungan teknologi internet yang memungkinkan untuk memasarkan produk secara domestik maupun internasional. Dengan harapan dapat meningkatkan tingkat perekonomian pelaku UMKM di wilayah Kabupaten Malang pada nantinya.

(Badriyah et al., 2018) melakukan penelitian mengenai rekomendasi produk berdasarkan transaksi belanja yang pernah dilakukan oleh pembeli di e-commerce. Sistem dapat menampilkan produk rekomendasi untuk setiap user member yang aktif. Dan pada setiap detail produk yang ditampilkan, dapat direkomendasikan produk lain berdasarkan persentase kemunculan dari produk lain dari user yang sedang menampilkan detail produk pada saat itu. Semakin besar nilai support dan confidence yang diberikan maka akan semakin sedikit rekomendasi yang ditampilkan dan semakin kecil nilai *support* dan *confidence* yang diberikan maka akan semakin banyak rekomendasi yang ditampilkan.

(Larasati & Februariyanti, 2021) melakukan penelitian mengenai sistem rekomendasi produk emina cosmetics dengan menggunakan metode content based filtering menggunakan metode content-based filtering dimana metode tersebut dapat digunakan untuk merekomendasikan produk berdasarkan ketersediaan

konten/deskripsi produk. Untuk menghitung kesamaan antar kalimat menggunakan algoritma cosine similarity. Mulanya deskripsi produk akan dilakukan pembobotan dengan tf-idf, lalu akan dihitung nilai similaritasnya dengan algoritma cosine similarity. Produk yang mengandung kata kunci akan dihitung nilai kemiripannya dan dilakukan perankingan berdasarkan nilai similaritas tertinggi hingga terendah. Dalam penelitian ini, produk dengan similaritas tertinggi didapat dengan nilai sebesar 0,7195.

## **2.2 Sistem**

Menurut (A.O'Brien, 2008), sistem adalah kumpulan elemen yang saling terhubung atau berinteraksi membentuk suatu kesatuan atau sekumpulan komponen yang saling terhubung dan bekerja sama untuk mencapai sasaran dengan menerima *input* dan menghasilkan *output* dalam sebuah proses *transformasi* yang terorganisir.

Sedangkan (Romney & Marshall B, 2015) Sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan. Sebagian besar sistem terdiri dari subsistem yang lebih kecil yang mendukung sistem yang lebih besar.

Berdasarkan pengertian diatas dapat disimpulkan bahwa sistem adalah kumpulan dari lebih dari satu komponen yang saling berkaitan satu dengan yang lain nya untuk mencapai suatu tujuan dalam melaksanakan suatu kegiatan pokok perusahaan.

## **2.3 Sistem Rekomendasi**

Sistem rekomendasi adalah merupakan sistem atau aplikasi yang dibuat untuk dapat menyediakan dan memberikan rekomendasi dari suatu item untuk membuat suatu keputusan yang diinginkan oleh pengguna sistem (Ungkawa et al., n.d.). Selanjutnya sistem rekomendasi merupakan suatu alat dan teknik dalam software (perangkat lunak) yang dapat memberikan saran-saran kepada pengguna untuk item yang sekiranya dapat bermanfaat bagi pengguna dalam menentukan pilihan. (Ricci et al., 2011) Saran-saran yang diberikan berhubungan dengan proses dalam pengambilan suatu keputusan, misalnya dalam menentukan item mana yang akan dibeli atau lagu mana yang ingin didengarkan. Ada 2 (dua) tipe

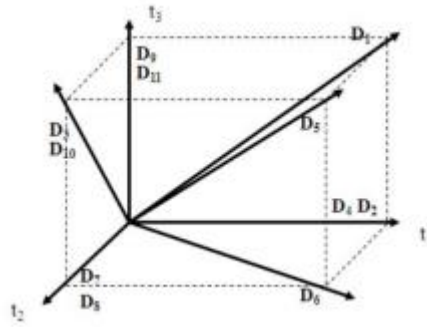
inputan (masukkan) berbeda yang sangat diperlukan dalam sistem rekomendasi, yaitu implicit input dan explicit input. Implicit input dapat dihasilkan melalui pengamatan secara langsung dari kebiasaan pengguna dalam menggunakan sistem misalnya dari catatan penelusuran, catatan pembelian, pola pencarian. Sedangkan Explicit input didapatkan dari hasil penilaian yang diberikan oleh pengguna sistem secara langsung, misalnya adalah pemberian rating atau thumbs-up/down untuk suatu item (Ricci et al., 2011)

## **2.4 Content-Based Filtering**

Metode Content-Based Filtering merupakan metode dalam sistem rekomendasi yang mampu menghasilkan sebuah rekomendasi berdasarkan content/atribut yang terkait dengan item lain yang dibandingkan. Rekomendasi produk menurut hasil uraian kemiripan produk yang sudah ditafsir pemakainya adalah konsep dari Content-Based Filtering (Tommy et al., 2019). Metode ini biasanya banyak diterapkan pada sistem penjualan *online*. Teknik – teknik yang biasa digunakan dalam content-based filtering seperti TF-IDF, Bayesian Classifiers, Cluster analysis, decision trees dan *vector space model* dalam penelitian ini saya menggunakan *vector space model*.

### **2.4.1 Vector Space Model**

Vector space model adalah suatu model yang digunakan untuk mengukur kemiripan antara suatu dokumen dengan suatu *query*. Pada model ini, query dan dokumen dianggap sebagai vektor-vektor pada ruang n-dimensi, dimana n adalah jumlah dari seluruh term yang ada dalam leksikon (Gadge et al., 2015). Leksikon adalah daftar semua term yang ada dalam indeks. Vector space model digunakan karena pada metode ini memungkinkan proses pemeringkatan dokumen. Metode ini menghitung nilai cosinus dari dua vektor. Dua vektor tersebut adalah bobot dari tiap dokumen dan bobot dari query. Bobot dokumen dan query digunakan untuk proses pemeringkatan dokumen, seperti yang terlihat pada Gambar 2.1



**Gambar 2.1** *Vector space model*

Pembobotan term dalam vektor dokumen dapat ditentukan dalam banyak cara. Pendekatan yang umum, dan digunakan dalam tugas akhir ini adalah metode TF-IDF (Nastiti, 2019). Pada metode ini, bobot term ditentukan oleh dua faktor: seberapa sering term  $j$  terdapat dalam dokumen (term frequency  $tf_{i,j}$ ) dan seberapa sering muncul dalam seluruh dokumen koleksi (document frequency  $df_j$ ). Tepatnya bobot term  $j$  pada dokumen  $i$  dirumuskan seperti berikut:

$$w_{i,j} = tf_{i,j} \times idf_j = tf_{i,j} \times \log N / df_j \quad (2.1)$$

$N$  merupakan jumlah dokumen yang terdapat dalam koleksi dokumen.  $idf$  merupakan singkatan dari inverse document frequency. Metode ini memberikan bobot tinggi pada term yang sering muncul pada sedikit dokumen pada dokumen set. Setelah bobot term dihitung, diperlukan suatu fungsi pemeringkatan untuk mengukur kemiripan antara query dan dokumen vektor. Pengukuran kemiripan yang umum dikenal sebagai pengukuran kosinus. Pengukuran ini menentukan sudut antara vektor dokumen dan query ketika direpresentasikan dalam  $V$ -dimensional Euclidean, dimana  $v$  adalah ukuran vocabulary. Tepatnya kemiripan antara dokumen  $D_i$  dan query  $Q$  didefinisikan sebagai berikut:

$$sim(q, d_i) = \frac{d_{i,q}}{|d_i| \cdot |q|} = \frac{\sum_{i=1}^t w_{iq} \cdot w_{ij}}{\sqrt{\sum_{i=1}^t (w_{ij})^2} \cdot \sqrt{\sum_{j=1}^t (w_{i,q})^2}} \quad (2.2)$$

Metode ini memiliki kelebihan yaitu dapat memberikan rekomendasi tanpa diperlukan adanya rating oleh *customer*, melainkan memberikan rekomendasi berdasarkan informasi item *query* di dalam keranjang *customer*, sedangkan

kekurangan dari metode yaitu sistem tidak dapat memberikan rekomendasi pada user yang belum pernah melakukan aktivitas apapun.

## **2.5 E-commerce**

Menurut (Sarwono, 2012) *E-Commerce* adalah proses membeli dan menjual atau tukar menukar produk, jasa dan informasi melalui komputer. Dalam E-Commerce terjadi proses pembelian dan penjualan jasa atau produk antara dua belah pihak melalui Internet atau pertukaran dan distribusi informasi antara dua pihak di dalam satu perusahaan dengan menggunakan Internet.

Menurut (Turban, 2010) *E-commerce* adalah penggunaan internet dan web untuk transaksi bisnis atau transaksi perdagangan yang dimungkinkan secara digital antar organisasi dengan organisasi atau antar individu. Dari pengertian para ahli diatas, maka dapat disimpulkan bahwa *E-commerce* adalah suatu kegiatan membeli atau menjual secara elektronik yang dilakukan pada jaringan internet. E-commerce terlihat lebih nyata, dengan adanya kebutuhan penjual dan pembeli untuk melakukan transaksi.

## **2.7 Software Development Life Cycle (SDLC)**

Menurut (Sukamto & Shalahuddin, 2019) *Software Development Life Cycle* adalah proses pengembangan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan best practice atau cara-cara yang sudah teruji baik), SDLC memiliki beberapa model dalam penerapan tahapan pemrosesannya antara lain waterfall, prototype, rapid application development (RAD), iteratif, dan spiral.

Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linear (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau berurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support*) (Sukamto & Shalahuddin, 2019).

Adapun tahapan dalam metode *waterfall* yaitu:

#### 1. Analisis

Analisis merupakan suatu kegiatan yang dimulai dari proses awal didalam mempelajari sesuatu serta mengevaluasi suatu bentuk permasalahan yang ada.

#### 2. Desain

Desain merupakan kegiatan dalam penggambaran, perencanaan, dan perancangan atau pengaturan dari beberapa elemen yang terpisah di dalam sistem menjadi kesatuan dan berfungsi dengan baik.

#### 3. Pengkodean

Pengodean merupakan upaya dalam mengimplementasikan desain menjadi perangkat lunak.

#### 4. Pengujian

Pengujian merupakan upaya dalam menelusuri lebih lanjut terhadap perangkat lunak yang telah dibuat untuk mendapatkan informasi mengenai kualitas perangkat lunak yang sedang diuji.

#### 5. Pemeliharaan

Pemeliharaan merupakan kegiatan yang dilakukan dalam perawatan dan perubahan atau pengembangan dari perangkat lunak yang telah dibuat dan tidak terdeteksi saat pengujian.

### **2.8 Data Flow Diagram (DFD)**

*Data flow diagram* (DFD) merupakan diagram yang menggunakan notasi-notasi atau simbol-simbol untuk menggambarkan sistem jaringan kerja antara fungsi-fungsi yang berhubungan satu sama lainnya dengan aliran data dan penyimpanan data (Nugroho, 2011).

*Data flow diagram* (DFD) sering digunakan untuk menggambarkan suatu sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan fisik dimana data tersebut mengalir atau dimana data tersebut akan disimpan. Salah satu keuntungan menggunakan diagram aliran data adalah memudahkan pemakai (*user*) yang kurang menguasai bidang komputer untuk mengerti sistem yang akan dikerjakan. DFD terdiri dari diagram konteks (*context diagram*) dan diagram rinci (*level diagram*).

## 2.9 Entity Relationship Diagram (ERD)

*Entity Relationship Diagram* (ERD) adalah gambaran pada sistem yang di dalamnya terdapat hubungan antara *entity* beserta relasinya. Untuk setiap *entity* biasanya mempunyai atribut yang merupakan ciri *entity* tersebut. Relasi adalah hubungan antar *entity* yang berfungsi sebagai hubungan yang mewujudkan pemetaan antar *entity* (Marlinda, 2004).

ERD memiliki 3 komponen, yaitu menurut (Jogiyanto H.M, 2005) adalah sebagai berikut :

### 1. Atribut

Sifat atau karakteristik dari suatu entitas.

### 2. Cardinality

Di Dalam ERD terdapat tingkat hubungan antara entitas yang satu dengan entitas lainnya atau disebut juga relasi. Meliputi :

#### a. *One to One* (1:1)

Setiap anggota entitas A hanya boleh berhubungan dengan satu anggota entitas B, begitu pula sebaliknya.

#### b. *One to many* (1:M / Many)

Setiap anggota entitas A dapat berhubungan dengan lebih dari satu anggota entitas B tetapi tidak sebaliknya.

#### c. *Many to Many* (M:M)

Setiap entitas A dapat berhubungan dengan banyak entitas himpunan entitas B dan demikian pula sebaliknya.

### 3. Entitas


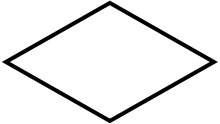
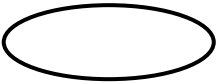


Memberikan identitas yang mewakili suatu objek dan dapat dibedakan dengan objek lain.

Pemodelan relasi basis data memiliki varian model/aliran, menurut (Sukanto & Shalahuddin, 2019), ERD memiliki beberapa aliran notasi seperti notasi Chen yang dikembangkan oleh Peter Chen; notasi Barker yang dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis; notasi *Crow's Foot*, dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi dari Chen.

Berdasarkan (Connolly & Begg, 2010), ERD merupakan representasi grafis dari logika *database* dengan menyertakan deskripsi detail mengenai seluruh *entity*, *relationship*, dan *constraint*.

Berikut ini adalah simbol-simbol pada ERD dengan notasi Chen, dapat dilihat pada Tabel 2.1 :

**Tabel 2. 1** Simbol-Simbol ERD

Simbol ERD	Keterangan
	Entitas : Memberikan identitas yang mewakili suatu objek
	Relasi : Menunjukkan hubungan diantara sejumlah entitas yang berbeda.
	Atribut : Properti yang dimiliki oleh suatu entitas, yang dapat mendeskripsikan karakteristik dari entitas tersebut.
	Atribut Kunci Primer : <i>Field</i> atau kolom data yang disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan, bersifat unik.
	Atribut <i>Multivalued</i> : <i>Field</i> atau kolom data yang disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.



## 2.10 Pengujian

Menurut (Sukamto & Shalahuddin, 2019) Pengujian perangkat lunak adalah sebuah elemen sebuah topik yang memiliki cakupan luas dan sering dikaitkan dengan verifikasi (*verification*) dan validasi (*validation*) (V&V). Verifikasi mengacu pada sekumpulan aktivitas yang menjamin bahwa perangkat lunak mengimplementasi fungsi yang benar dan spesifik. Validasi mengacu pada sekumpulan aktivitas yang berbeda yang menjamin bahwa perangkat lunak yang dibangun dapat ditelusuri sesuai dengan kebutuhan pelanggan (*customer*).

### 2.10.1 *Blackbox*

*Blackbox testing* atau pengujian *blackbox* yaitu metode uji coba yang menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan (Sukamto & Shalahuddin, 2019).

Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses *login* maka kasus uji yang dibuat adalah (Sukamto & Shalahuddin, 2019):

- Jika user memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
- Jika user memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.

### 2.10.2 *User Acceptance Testing (UAT)*

Menurut (Perry, 2006), *User Acceptance Testing* merupakan pengujian yang dilakukan oleh *end-user* dimana *user* tersebut adalah staf/karyawan perusahaan yang langsung berinteraksi dengan sistem dan dilakukan verifikasi apakah fungsi yang ada telah berjalan sesuai dengan kebutuhan/fungsinya.

Menurut (Lewis, 2009), setelah dilakukan *system testing*, *acceptance testing* menyatakan bahwa sistem software memenuhi persyaratan. *Acceptance testing* merupakan pengujian yang dilakukan oleh pengguna yang menggunakan teknik pengujian *black box* untuk menguji sistem terhadap spesifikasinya. Pengguna akhir bertanggung jawab untuk memastikan semua fungsionalitas yang relevan telah diuji.

Menurut (Kartikawati & Wibawa, 2020) menjelaskan bahwa *User Acceptance Test* (UAT) atau biasa disebut dengan uji penerimaan pengguna ialah suatu proses pengujian oleh pengguna dengan hasil output berupa dokumen hasil uji yang dapat dijadikan bukti bahwa *software* sudah diterima dan memenuhi kebutuhan yang diminta.

Berikut adalah rumus menghitung skor pengujian *user acceptance testing* (UAT), yaitu persamaan 2.1 (Muqsith, 2016):

$$Skor = \frac{\text{Jumlah Jawaban}}{(\text{Jumlah Pertanyaan} \times \text{Jumlah Responden})} \times 100\% \quad (2.3)$$

Keterangan:

Jumlah jawaban = Jumlah jawaban responden ya/ tidak

Jumlah pertanyaan = Jumlah pertanyaan yang diajukan kepada responden

Jumlah responden = Jumlah responden yang mengisi kuisisioner

## 2.11 HTML

Menurut (Hidayatullah & Khawistara, 2015) "*Hypertext Markup Language* (HTML) adalah bahasa standar yang digunakan untuk menampilkan halaman *web*". Yang bisa dilakukan dengan HTML yaitu:

1. Mengatur tampilan dari halaman *web* dan isinya.
2. Membuat tabel dalam halaman *web*.
3. Mempublikasikan halaman *web* secara *online*.

Membuat form yang bisa digunakan untuk menangani *registrasi* dan *transaksi* via *web*.

## 2.12 Codeigniter

Menurut (Raharjo Budi, 2017) *Codeigniter* memiliki banyak fitur (fasilitas) yang membantu para pengembang (developer) *PHP* untuk dapat membuat aplikasi website secara mudah dan cepat. Dibandingkan dengan framework web *PHP* lainnya. Harus diakui bahwa *Codeigniter* memiliki desain yang lebih sederhana dan bersifat fleksibel (tidak kaku). *Codeigniter* mengizinkan para pengembang untuk menggunakan framework secara parsial maupun secara keseluruhan. Ini berarti bahwa *Codeigniter* masih memberi kebebasan kepada para pengembang untuk menulis bagian-bagian kode tertentu di dalam aplikasi menggunakan cara konvensional (tanpa *framework*).

## 2.13 PHP

Menurut (Madcoms, 2016) “*PHP (Hypertext Preprocessor)* adalah bahasa script yang dapat ditanamkan atau disisipkan ke dalam HTML. *PHP* banyak dipakai untuk membuat program situs web dinamis”. *PHP* dapat digunakan dengan gratis (*free*) dan bersifat *Open Source*. *PHP* dirilis dalam lisensi *PHP license*. Untuk membuat program *PHP* kita diharuskan untuk menginstal *web server* terlebih dahulu. *PHP* mendukung komentar seperti pada bahasa ‘C’, ‘C++’, dan Unix shell-style. (*Perl style*)

## 2.14 CSS

*CSS* kepanjangan dari *Cascading Style Sheet* adalah bahasa-bahasa yang merepresentasikan halaman *web*. Seperti warna, *layout*, dan font. Dengan menggunakan *CSS*, seorang *web developer* dapat membuat halaman *web* yang dapat beradaptasi dengan berbagai macam ukuran layar. Pembuatan *CSS* biasanya terpisah dengan halaman HTML. Meskipun *CSS* dapat disisipkan di dalam halaman HTML. Hal ini ditujukan untuk memudahkan pengaturan halaman HTML yang memiliki rancangan yang sama (Henderson, 2009).

## 2.15 JavaScript

Menurut (Kustiyahningsih & Yeni, 2011), *JavaScript* adalah bahasa yang berbentuk kumpulan skrip berjalan pada suatu dokumen HTML. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap HTML dengan mengizinkan pengekseskuan perintah-perintah di sisi user

variabel atau fungsi dengan nama TEST berbeda dengan variabel dengan nama test dan setiap instruksi diakhiri dengan artinya disisi browser bukan di sisi server web. JavaScript adalah bahasa yang “case sensitive” artinya membedakan penamaan variabel dan fungsi yang menggunakan huruf besar dan huruf kecil, contoh karakter titik koma.

## **2.16 XAMPP**

Menurut (Madcoms, 2016) “Xampp adalah sebuah paket kumpulan *software* yang terdiri dari Apache, MySQL, PhpMyAdmin, PHP, Perl, Filezilla, dan lain.” Xampp berfungsi untuk memudahkan instalasi lingkungan PHP, di mana biasanya lingkungan pengembangan web memerlukan PHP, Apache, MySQL dan PhpMyAdmin.