

BAB II

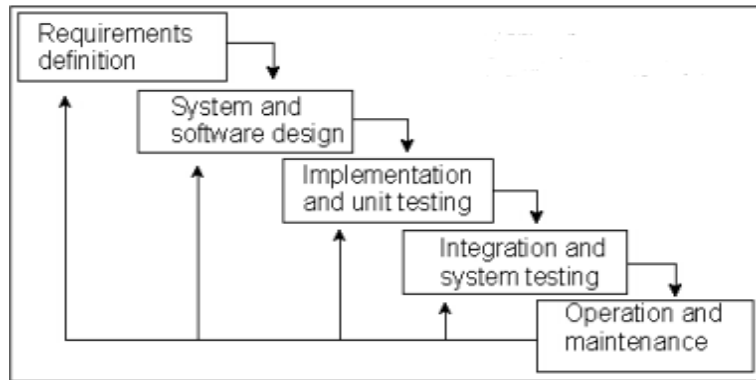
LANDASAN TEORI

2.1. Penerapan

Menurut Kamus Besar Bahasa Indonesia (KBBI), pengertian penerapan adalah perbuatan menerapkan. Penerapan atau implementasi adalah realisasi dari aplikasi, atau eksekusi dari rencana, ide, model, desain, spesifikasi, standar, algoritma, atau kebijakan.

2.2. Metode *Waterfall*

Model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Nama model ini sebenarnya adalah “Linear Sequential Model”. Model ini sering disebut juga dengan “*classic life cycle*” atau metode *waterfall*. Model ini termasuk ke dalam model *generic* pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam *Software Engineering (SE)*. Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Di bawah ini merupakan gambar dari tahapan-tahapan metode *waterfall*.



Gambar 2.1 Metode *Waterfall* (Pressman, Roger S.)

Berikut ini merupakan rincian penjelasan dari tahapan-tahapan metode *waterfall*:

2.2.1. Requirement Definition (Definisi Kebutuhan)

Merupakan tahapan penetapan fitur, kendala dan tujuan sistem melalui konsultasi dengan pengguna sistem. Semua hal tersebut akan ditetapkan secara rinci dan berfungsi sebagai spesifikasi sistem.

2.2.2. System and Software Design (Desain Sistem dan Perangkat Lunak)

Dalam tahapan ini akan dibentuk suatu arsitektur sistem berdasarkan persyaratan yang telah ditetapkan. Dan juga mengidentifikasi dan menggambarkan abstraksi dasar sistem perangkat lunak dan hubungan-hubungannya.

2.2.3. Implementation and Unit Testing (Implementasi dan Testing Unit)

Dalam tahapan ini, hasil dari desain perangkat lunak akan direalisasikan sebagai satu set program atau unit program. Setiap unit akan diuji apakah sudah memenuhi spesifikasinya.

2.2.4. *Integration and System Testing (Integrasi dan Testing Sistem)*

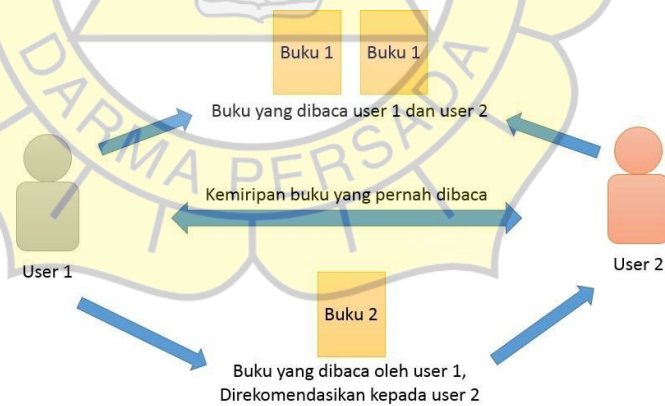
Dalam tahapan ini, setiap unit program akan diintegrasikan satu sama lain dan diuji sebagai satu sistem yang utuh untuk memastikan sistem sudah memenuhi persyaratan yang ada. Setelah itu sistem akan dikirim ke pengguna sistem.

2.2.5. *Operation and Maintenance (Operasional dan Pemeliharaan)*

Dalam tahapan ini, sistem diinstall dan mulai digunakan. Selain itu juga memperbaiki *error* yang tidak ditemukan pada tahap pembuatan. Dalam tahap ini juga dilakukan pengembangan sistem seperti penambahan fitur dan fungsi baru.

2.3. *Collaborative Filtering*

Collaborative filtering adalah sistem yang memberikan rekomendasi berdasarkan pada ketertarikan sebelumnya dari pengguna dan opini dari pengguna-pengguna lain yang mempunyai ketertarikan yang mirip.



Gambar 2.2 *Collaborative Filtering*

Pada pendekatan *collaborative filtering*, dilakukan perhitungan jarak rating yang telah diketahui, antara pengguna dengan pengguna yang ingin diberikan rekomendasi. Salah satu ukuran jarak yang dapat dipakai adalah *euclidean distance*.

Berikut langkah-langkah menghitungnya:

1. Menghitung *euclidean distance* antara *rating* yang diberikan oleh *user*

$$d(x,y) = \sqrt{\sum_{i=1}^3(x_i - y_i)^2}$$

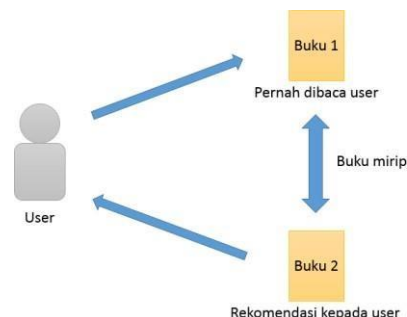
2. Menghitung nilai kemiripan *rating* antar *user*

$$\text{Sim}(x,y) = \frac{1}{1+d(x,y)}$$

3. Membuat urutan rekomendasi dengan cara mengalikan nilai kemiripan setiap *user* dengan *rating* yang diberikan untuk setiap buku

2.4. **Content Based Filtering**

Content Based Recommendation memanfaatkan informasi beberapa *item* atau data untuk direkomendasikan kepada pengguna sebagai referensi yang terkait dengan informasi yang digunakan sebelumnya. Tujuan dari *content based recommendation* agar dapat memprediksi persamaan dari sejumlah informasi yang didapat dari pengguna. Misalnya, jika murid hanya meminjam dan membaca buku tentang web *programming*, maka sistem akan memprediksinya hanya buku yang mirip dengan web *programming*.



Gambar 2.3 *Content Based Filtering*

Berikut langkah-langkah menghitungnya:

1. Melakukan tokenisasi

Tokenisasi adalah proses untuk membagi teks yang dapat berupa kalimat, paragraf atau dokumen, menjadi token-token atau bagian-bagian tertentu.

2. Melakukan *stemming*

Stemming adalah proses untuk menemukan kata dasar dari sebuah kata.

3. Melakukan *indexing*

Indexing adalah pengumpulan, penguraian, dan penyimpanan data untuk memfasilitasi pengambilan informasi yang cepat dan akurat.

4. Melakukan pembobotan dan menghitung panjang vektor

$$\text{Idf (kata)} = \log(\text{Jumlah dokumen/df kata})$$

5. Menghitung kemiripan kata kunci dengan setiap dokumen menggunakan *cosine similarity*

$$\text{similarity}(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}}$$

2.5. Sistem

Sekelompok *item* yang berinteraksi secara teratur atau saling bergantung membentuk satu kesatuan yang utuh. Kata "sistem" banyak sekali digunakan dalam percakapan sehari-hari, dalam forum diskusi maupun dokumen ilmiah. Kata ini digunakan untuk banyak hal, dan pada banyak bidang pula, sehingga

maknanya menjadi beragam. Dalam pengertian yang paling umum, sebuah sistem adalah sekumpulan benda yang memiliki hubungan di antara mereka.

2.6. Pengelolaan

Pengertian pengelolaan adalah pemanfaatan sumber daya manusia ataupun sumber daya lainnya yang dapat diwujudkan dalam kegiatan perencanaan, pengorganisasian, pengarahan dan pengawasan untuk mencapai suatu tujuan tertentu.

2.7. Perpustakaan Berbasis Website

Istilah ‘perpustakaan digital’ atau digital library pertama sekali muncul pada bulan Juli tahun 1945 ketika Vannevar Bush merasa akses informasi yang terpublikasi terhambat dengan metode model cetak manual. Idennya ini dituangkan dalam karya tulisnya yang berjudul “*As We May Think*”.

Menurut pandangan para pakar yang berorientasi pada konten atau koleksi, maka perpustakaan digital adalah perpustakaan yang memuat informasi atau koleksi yang diatur dalam format digital. Pelayanan koleksi digital ini juga dilayanan secara digital dengan menggunakan bantuan perangkat elektronik dan teknologi komunikasi dan informasi.

2.8. Bahasa Pemrograman

2.8.1. PHP

Menurut Supono & Putratama (2018:1) mengemukakan bahwa “PHP (PHP: hypertext preprocessor) adalah suatu bahasa pemrograman yang digunakan

untuk menterjemahkan basis kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat *server-side* yang ditambahkan ke HTML”.

Sebagai sebuah *scripting language*, PHP menjalankan instruksi pemrograman saat proses *runtime*. Hasil dari instruksi tentu akan berbeda tergantung data yang diproses. PHP merupakan bahasa pemrograman *server-side*, maka *script* dari PHP nantinya akan diproses di server. Jenis server yang sering digunakan bersama dengan PHP antara lain Apache, Nginx, dan LiteSpeed.

2.8.2. JavaScript

JavaScript adalah bahasa pemrograman yang digunakan dalam pengembangan *website* agar lebih dinamis dan interaktif. JavaScript dapat meningkatkan fungsionalitas pada halaman web. Bahkan dengan JavaScript ini kamu bisa membuat aplikasi, *tools*, atau bahkan *game* pada web.

Bicara teknis, JavaScript atau kita singkat menjadi JS merupakan bahasa pemrograman jenis interpreter, sehingga kamu tidak memerlukan *compiler* untuk menjalankannya. JavaScript memiliki fitur-fitur seperti berorientasi objek, *client-side*, *high-level programming*, dan *loosely typed*.

2.9. Database MySQL

Website tentu saja membutuhkan *database* server untuk menampung berbagai macam informasi. Berbagai macam data dibutuhkan oleh *website* seperti *username*, *password*, *font*, *URL*, dan sejenisnya. MySQL adalah salah satu sistem manajemen *database* yang biasa digunakan untuk mengelola data tersebut. MySQL merupakan *software database open source* yang sering digunakan untuk mengolah basis data yang menggunakan bahasa SQL (Subagia, 2018:67).

MySQL adalah *DBMS* yang *open source* dengan dua bentuk lisensi, yaitu *Free Software* (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik yang penggunaannya terbatas). Jadi MySQL adalah database server yang gratis dengan lisensi *GNU General Public License (GPL)* sehingga dapat Anda pakai untuk keperluan pribadi atau komersil tanpa harus membayar lisensi yang ada. Alasan Penulis memilih menggunakan MySQL sebagai *database management system* karena mudah digunakan.

2.10. UML

Kependekan dari *Unified Modeling Language*, adalah bahasa pemodelan standar yang terdiri dari seperangkat diagram terintegrasi, yang dikembangkan untuk membantu pengembang sistem dan perangkat lunak untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan artefak sistem perangkat lunak, serta untuk pemodelan bisnis dan non-sistem perangkat lunak.

2.10.1. Use Case Diagram

Diagram yang menunjukkan peran *user* dan bagaimana peran tersebut ketika menggunakan sistem. *Use case* diagram juga dapat digunakan untuk mempresentasikan interaksi *user* dengan sistem dan menggambarkan spesifikasi kasus penggunaan.

2.10.2. Activity Diagram

Diagram yang menggambarkan alur kerja dari berbagai aktivitas *user* atau sistem, orang yang melakukan aktivitas, dan aliran berurutan dari aktivitas ini.

2.10.3. *Sequence Diagram*

Diagram yang menggambarkan alur kerja dari berbagai aktivitas *user* atau sistem, orang yang melakukan aktivitas, dan aliran berurutan dari aktivitas ini.

2.10.4. *Class Diagram*

Sebuah diagram yang menjelaskan hubungan antar *class* dalam sebuah sistem yang sedang dibuat dan menjelaskan bagaimana caranya agar mereka saling berkolaborasi.

2.11. *Entity Relationship Diagram (ERD)*

Banyak implementasi *database* dikembangkan, namun model relasional adalah yang paling umum, dan pilihan utama untuk sebagian besar sistem perangkat lunak. Model relasional membutuhkan definisi tabel yang masing-masing terdiri dari catatan yang berisi berbagai bidang data untuk menggambarkan entitas bisnis seperti produk, pelanggan, dan sejenisnya. Rekaman berhubungan satu sama lain menggunakan bidang kunci yang merupakan subkumpulan bidang tabel. Untuk merancang skema relasional yang baik, (yaitu, tabel, *fields*, dan *keys*), seseorang perlu memahami persyaratan sistem seperti yang dinyatakan oleh pelanggan. Spesifikasi persyaratan ini biasanya mengacu pada kemungkinan skenario interaksi sistem pengguna yang menentukan input dan output data. Model data kemudian diekstraksi dari persyaratan ini, untuk mendukung fungsionalitas sistem yang ditentukan. *Entity-Relationships-Diagram (ERD)* adalah model visual yang banyak digunakan untuk menggambarkan entitas bisnis, atributnya, dan hubungannya satu sama lain, yang diperkenalkan oleh Chen.