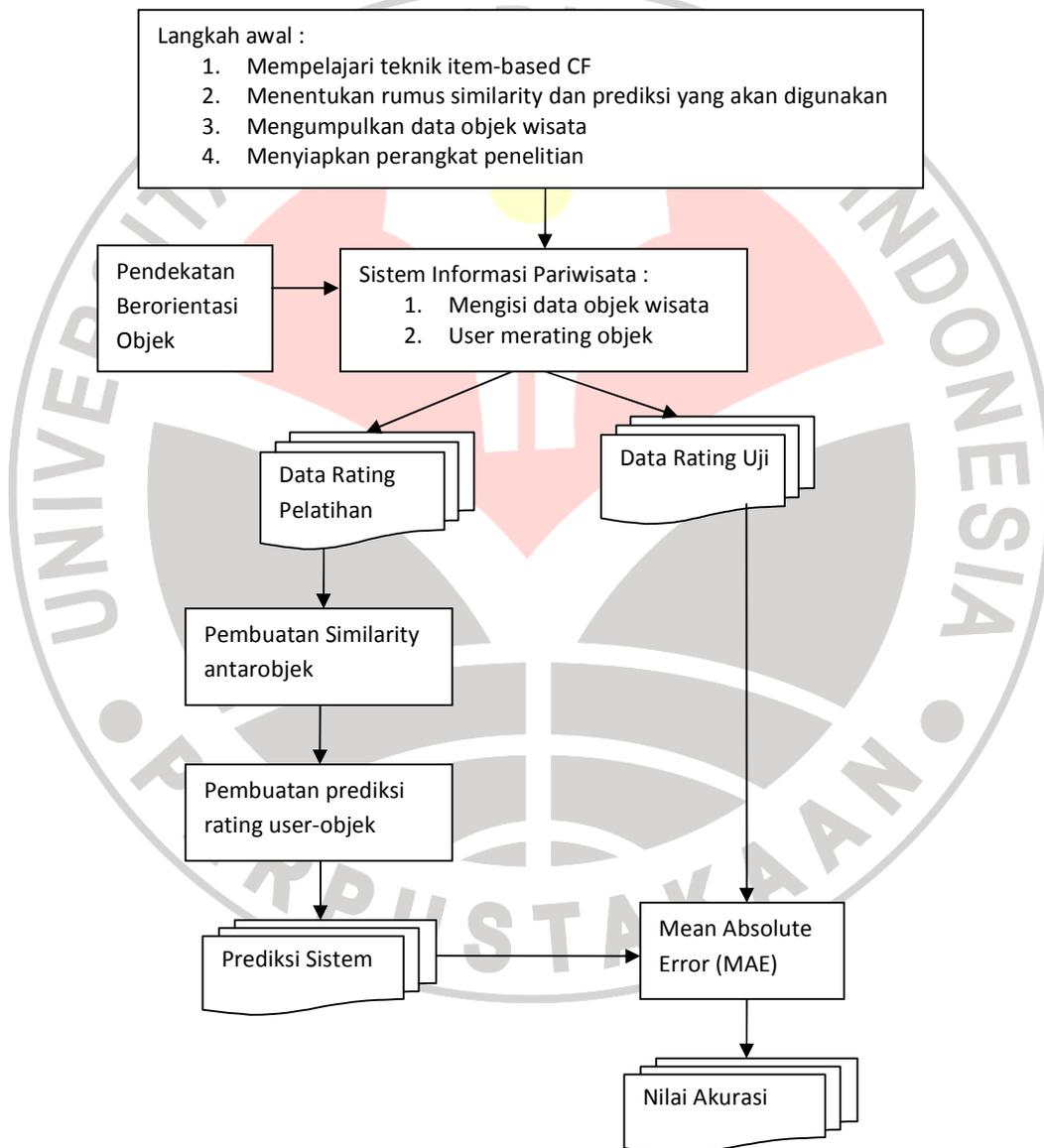


BAB III METODOLOGI PENELITIAN

1.1 Desain Penelitian

Gambar 3.1 merupakan desain penelitian yang akan digunakan pada proses pembuatan rekomendasi objek wisata pada sistem informasi pariwisata menggunakan teknik *item-based collaborative filtering*.



Gambar 3.1 Desain Penelitian

1.2 Fokus Penelitian

Fokus masalah dalam penelitian ini adalah proses pembuatan prediksi yang akan dilakukan oleh sistem serta mengukur tingkat akurasi dari prediksi yang telah dihasilkan. Proses prediksi dibuat dengan cara menghitung nilai *similarity* menggunakan dua cara, yaitu dengan menggunakan *Pearson Correlation* (persamaan 2.1) dan *adjusted-cosine* (persamaan 2.2) yang kemudian dilanjutkan dengan rumus *weighted-sum* (persamaan 2.3) untuk mendapatkan nilai prediksi. Tingkat akurasi hasil prediksi dari sistem diukur dengan menggunakan rumus *mean absolute error* (MAE, persamaan 2.4).

1.3 Alat dan Bahan yang Digunakan

Pada penelitian ini menggunakan alat penelitian berupa sistem komputer dan perangkat lunaknya serta bahan penelitian sebagai berikut, yaitu :

1. Sistem Komputer

Untuk mengembangkan sistem informasi pariwisata ini, diperlukan seperangkat komputer dengan spesifikasi sebagai berikut.

- a. Processor AMD Turion64 X2 2.0 Ghz atau yang sekelas
- b. RAM 2 GB
- c. Harddisk 40 GB dengan ruang kosong 5 GB
- d. Monitor dengan resolusi 1280 x 800, 32 bit color
- e. Mouse dan Keyboard

2. Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan sistem informasi pariwisata ini adalah sebagai berikut.

- a. Sistem Operasi Windows XP SP3

- b. Adobe Flash CS3 dan Adobe Flex Builder 3 beserta Flex SDK 3.2
- c. Paket webserver XAMPP versi 1.7.2
- d. Remoting component AMFPHP
- e. Text editor Notepad++
- f. Browser (Firefox, IE, Opera)
- g. Flash Player 10

3. Koneksi Internet

Koneksi internet digunakan untuk meng-upload sistem ke webhost, sehingga sistem bisa digunakan secara online.

1. Bahan Penelitian

- a. Data objek pariwisata Kota Bandung yang berasal dari Dinas Pariwisata Kota Bandung dan situs objek wisata lainnya.
Rating *user* terhadap suatu objek yang digunakan sebagai bahan untuk meramu rekomendasi.

1.4 Metode Penelitian

1.4.1 Proses Rekayasa Sistem

1.4.1.1 Metode Pendekatan Perangkat Lunak

Dalam mengembangkan sistem ini, penulis menggunakan pendekatan yang berorientasi objek. Hal ini dikarenakan adanya tuntutan dari tools yang digunakan oleh penulis, yakni bahasa pemrograman.

Beberapa konsep yang harus diketahui dalam pemrograman berorientasi adalah sebagai berikut :

1. *Class* dan *Object*

Class adalah suatu definisi abstrak dari suatu objek. *Class* menjelaskan karakteristik objek termasuk atribut dan perilakunya. *Class* bisa dikatakan sebagai *blueprint* yang menjelaskan suatu objek. Misalnya *class* Kucing menjelaskan karakteristik dari warna bulu dan jenisnya, serta kemampuan untuk lari, mengeong, duduk, serta mencakar.

Object adalah contoh nyata dari suatu *class*. *Class* Kucing mendefinisikan semua karakteristik dan perilaku yang mereka miliki. Misalnya, Tom adalah seekor Kucing. Tom memiliki bulu berwarna kuning dan putih serta berjenis kucing hutan. Tentu saja Tom bisa mengeong dan berlari.

2. *Property* dan *Method*

Property dan *method* adalah bagian dari definisi *class*. *Property* merupakan atribut/karakteristik dari suatu *class*. *Class* Kucing yang telah terdefinisi memiliki atribut jumlah kaki, warna bulu, jenis, panjang ekor, dan sebagainya.

Sementara *method* adalah kemampuan yang bisa dilakukan oleh *class*. Pada *class* Kucing, *method* yang umum dimiliki adalah *mengeong()*, *lari()*, *duduk()*, *menggigit()*, *makan()*, *mencakar()*, dan sebagainya.

3. *MessagePassing*

Message passing adalah proses dimana satu objek mengirim data pada objek lainnya sehingga memungkinkan objek lain tersebut melakukan sesuatu. Misalnya, seorang tuan memanggil kucing Tom, maka Tom akan merespon dengan mengerjakan *method* *mengeong()*.

4. *Encapsulation*

Encapsulation adalah menyembunyikan detail suatu class dari objek yang memanggilnya. Ketika tuan memanggil kucing Tom, maka tuan hanya ingin mendengar Tom mengeong() tanpa harus tahu bagaimana cara Tom mengeong(). Padahal yang dilakukan tom dalam method mengeong() adalah tarikNafas(), aturSuara(), keluarkanSuara(), lepaskanNapas().

5. *Inheritance*

Inheritance pada dasarnya adalah *subclass* yang memiliki *property* dan *method* yang lebih spesifik dari *parent class*-nya. Misalkan KucingRumahan adalah turunan dari *class* Kucing. Pada *class* KucingRumahan terdapat *method* malasMalasan() yang mana pada *parent class* Kucing *method* tersebut tidak ada. Tom adalah KucingRumahan. Oleh karena itu, Tom dapat melakukan *method* malasMalasan() yang tidak bisa dilakukan oleh objek lain yang berasal dari *class* Kucing.

6. *Polymorphism*

Polymorphism adalah kemampuan untuk melakukan sesuatu yang berbeda dengan memanggil *method* yang sama. Misalkan Kucing dan Anjing adalah *class* turunan dari Hewan. Pada *class* Hewan terdapat *method* bersuara(). Jika seorang tuan menyuruh Tom untuk bersuara(), maka Tom akan mengeong(), sementara jika tuan menyuruh Spike bersuara(), maka Spike akan menggonggong().

Ada standar hal yang harus diperhatikan dalam merancang model sistem berbasis objek. Pemodelan berbasis objek harus sesuai dengan standar UML. Pemilihan standar UML dikarenakan sifatnya yang umum dan bisa dimengerti oleh *developer* lain.

Menurut Wikipedia, jenis pemodelan dalam UML dikategorikan menjadi tiga, yaitu:

1. *Structure Diagram*

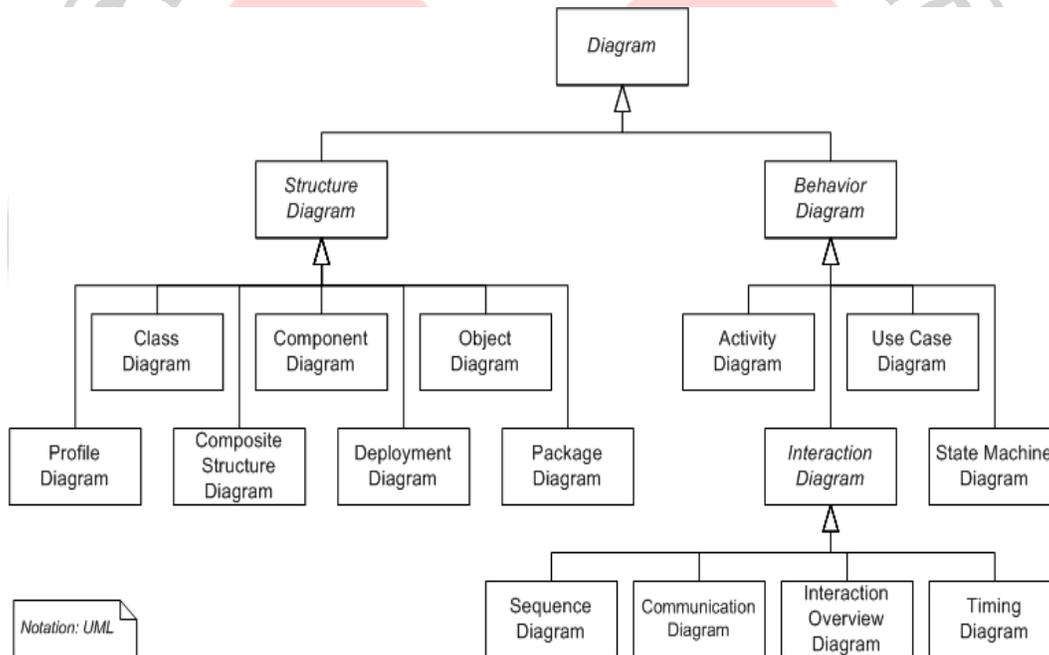
Structure Diagram menunjukkan sesuatu yang harus ada pada sistem yang dimodelkan.

2. *Behavior Diagram*

Behavior Diagram menggambarkan apa saja yang harus terjadi pada sistem yang dimodelkan.

3. *Interaction Diagram*

Pada dasarnya *interaction diagram* adalah bagian dari *behavior diagram*. Jenis ini menunjukkan alur kontrol dan data dari setiap objek pada sistem yang dimodelkan.



Gambar 3.2 Struktur Diagram UML

1.4.1.2 Model Proses

Model proses yang umum digunakan dalam pengembangan perangkat lunak yang berorientasi objek adalah *unified process* (UP). Ciri khas UP adalah proses

pengembangan perangkat lunak dikerjakan secara iteratif (berulang) dan inkremental (bertahap).

Secara umum, proses yang dilakukan dalam model proses UP ini dibagi menjadi empat fase, yaitu:

1. Tahap *Inception*

Dalam tahap ini dijelaskan kebutuhan dan spesifikasi pengembangan yang mana di dalamnya sudah termasuk identifikasi proses bisnis, asumsi, batasan dan arsitektur yang akan digunakan.

2. Tahap *Elaboration*

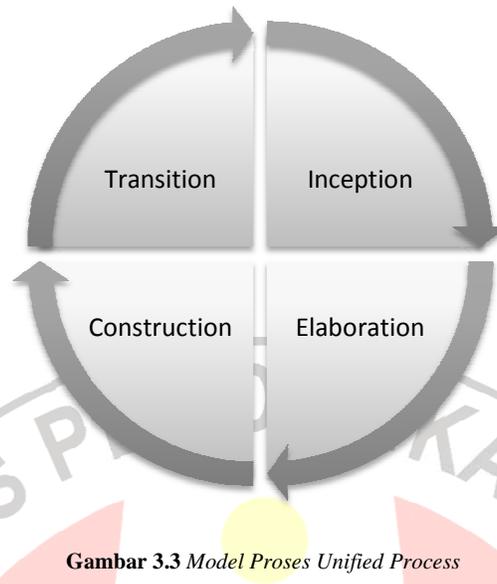
Tujuan dari tahap elaboration adalah mengubah kebutuhan dan spesifikasi yang telah dibuat pada tahap sebelumnya menjadi suatu solusi teknis yang detil dalam bentuk dokumen. Dokumen itu bisa saja berisi spesifikasi modul, fungsi, data, class, dan sebagainya.

3. Tahap *Construction*

Tahap ini adalah tahap dimana perangkat lunak dikembangkan berdasarkan dokumen spesifikasi yang telah dibuat pada tahap sebelumnya.

4. Tahap *Transition*

Dalam tahap ini, dilakukan konfigurasi pengujian di mana perangkat lunak akan digunakan. Fokus utama pada tahap ini adalah memastikan bahwa perangkat lunak dapat digunakan oleh *user*.



Gambar 3.3 Model Proses Unified Process

1.4.2 Proses Pengumpulan Data

1. Studi Pustaka

Mempelajari literatur berupa artikel, paper maupun sumber lain yang mendukung yang berkaitan dengan objek penelitian.

2. Observasi

Menghubungi Dinas Pariwisata Kota Bandung untuk memperoleh data yang dapat menunjang proses penelitian.

3. Pengumpulan Data *Rating*

Mengumpulkan data *rating* objek wisata yang diberikan oleh *user* melalui sistem informasi pariwisata yang telah dibuat.

1.5 Tahapan Penelitian

1.5.1 Langkah Awal Penelitian

Merupakan penjelasan dari setiap langkah awal penelitian, yaitu :

1. Mempelajari teknik *item-based collaborative filtering*

Teknik *item-based collaborative filtering* merupakan salah satu dari sekian banyak teknik untuk membuat suatu *recommender system*. Sumber belajar mengenai teknik ini mudah ditemukan di internet.

2. Menentukan rumus *similarity* dan prediksi

Dari sekian banyak rumus untuk menentukan *similarity* dan prediksi, penulis memilih *Pearson correlation* dan *adjusted-cosine* untuk menentukan *similarity* serta *weighted-sum* untuk membuat prediksi. Hal ini dikarenakan sudah banyak artikel, paper dan tutorial yang membahas mengenai ketiga rumus itu. Dengan demikian, penulis dapat memperoleh kemudahan dalam membuat *recommender system*.

3. Mengumpulkan data objek wisata

Penulis menghubungi Dinas Pariwisata Kota Bandung untuk memperoleh data objek wisata yang *valid*.

4. Menyiapkan perangkat penelitian

Penulis membuat suatu sistem informasi pariwisata sederhana untuk menerapkan *recommender system* yang akan dibuat. Sistem itu dibuat dengan menggunakan Flex SDK yang *output*-nya berbasis Flash. Untuk keperluan *back-end*, digunakan bahasa *scripting server-side* PHP, *remoting component* AMFPHP dan *database* MySQL.

1.5.2 Pengembangan Sistem Informasi Pariwisata

Berdasarkan model proses UP yang telah dipaparkan, tahapan kegiatan yang dilakukan dalam mengembangkan sistem informasi ini adalah sebagai berikut.

1. Tahap *Inception*

Kegiatan yang dilakukan pada tahap *inception* adalah sebagai berikut:

- a. Mengidentifikasi batasan dan kebutuhan perangkat lunak.
- b. Mengidentifikasi proses bisnis.

c. Mengidentifikasi arsitektur perangkat lunak.

2. Tahap *Elaboration*

Kegiatan pada tahap *elaboration* adalah membuat spesifikasi kebutuhan perangkat lunak yang lebih detail.

3. Tahap *Construction*

Di tahap *construction*, kegiatan yang dilakukan adalah memodelkan sistem, membangun sistem, dan menguji sistem. Di samping itu, dibuat pula dokumentasi pengembangan perangkat lunak.

4. Tahap *Transition*

Dalam tahap *transition* dilakukan pengujian terhadap sistem yang dibuat. Pengujian dilakukan baik dari sisi pengembang (internal) maupun sisi pengguna (eksternal) hingga akhirnya sistem tersebut dapat dinyatakan telah selesai.

1.5.3 Pembuatan Similarity Antarobjek

Tahapan awal untuk membuat suatu *recommender system* adalah dengan menentukan nilai *similarity* antarobjek. Nilai *similarity* ini diperoleh dari data *rating* pelatihan dengan menggunakan rumus persamaan 2.1 dan persamaan 2.2.

1.5.4 Pembuatan Prediksi Rating Objek

Setelah menghitung nilai *similarity*, tahapan berikutnya adalah membuat prediksi *ratinguser* untuk setiap objek lain selain dari data *rating* pelatihan dengan menggunakan persamaan 2.3. Nilai prediksi ini bisa diketahui dengan cara memanfaatkan nilai *similarity* dan *ratinguser* terhadap objek. Nilai prediksi inilah yang akan digunakan oleh sistem untuk membuat suatu rekomendasi.

1.5.5 Pengukuran Akurasi Hasil Prediksi

Untuk mengetahui tingkat akurasi dari prediksi yang telah dibuat, maka dibutuhkan sejumlah data uji untuk menguji hasil prediksi sistem. Persamaan 2.4 digunakan untuk mengukur akurasi hasil prediksi dengan data uji.

