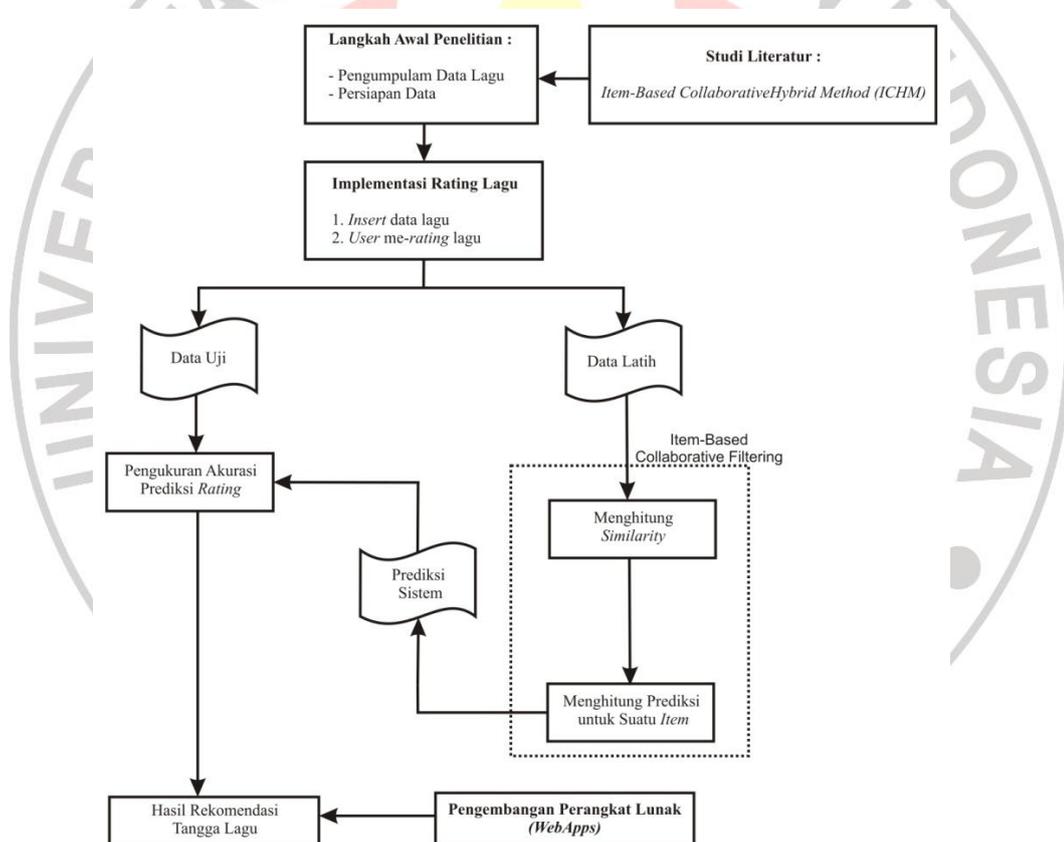


BAB III

METODOLOGI PENELITIAN

3.1 Desain Penelitian

Gambar 3.1 merupakan desain penelitian sistem yang akan digunakan pada sistem rekomendasi lagu dengan menggunakan *Item-Based Collaborative Filtering*.



Gambar 3.1 Desain Penelitian

Berikut adalah penjelasan dari bagan desain penelitian pada gambar 3.1:

Dalam proses pengumpulan data dilakukan dalam berbagai cara, diantaranya :

a. Studi Literatur

Mengumpulkan dan mempelajari literatur atau kepustakaan yang berhubungan dengan penelitian ini, berupa jurnal ilmiah, artikel maupun sumber-sumber bacaan lainnya.

b. Langkah Awal Penelitian

1. Pengumpulan Data Lagu

Mengumpulkan data lagu, dimana data lagu yang akan digunakan adalah sebagian data lagu yang tergolong *top chart (top 500)* Indonesia (Februari 2013).

Dalam penelitian ini, pengumpulan data lagu menggunakan *Application Programming Interface (API)* dari <http://www.last.fm> dimana peneliti membuat aplikasi khusus untuk mendapatkan data lagu dari *database* mereka.

2. Persiapan Data

Data yang sudah terkumpul (data lagu) dipersiapkan untuk dapat diproses dan di-*rating* oleh *user*. Prosesnya yaitu *user* secara sukarela mendaftar pada aplikasi dan merating untuk setiap lagu yang telah didengarnya dengan skala *rating* 1-5.

c. Implementasi Rating Lagu

Langkah selanjutnya yaitu tahap implementasi *rating* lagu, tahapan yang dilaksanakan dalam implementasi ini adalah sebagai berikut :

1. *Insert* data lagu

Memasukan data lagu yang akan diproses kedalam sistem. Data yang dimasuk meliputi artis, judul lagu dan album yang tergabung dalam satu paket data lagu. Data lagu ini diambil dari <http://www.last.fm> dengan menggunakan API yang disediakan oleh <http://www.last.fm>.

2. *User me-rating* lagu

Data lagu yang telah dimasukan lalu di *rating* oleh *user* yang sudah mendaftar sebelumnya. Di tahapan ini *user me-rating* sesuai dengan penilaian sendiri dan bersifat subjektif. Data lagu yang di-*rating* sudah tersedia di sistem dan tidak bisa ditambahkan oleh *user*. Penilaian yang diberikan *user* berdasarkan karakteristik berikut ini (**Lampiran A**):

- a. artis
- b. genre
- c. lirik
- d. mood
- e. musikalitas
- f. sedang hits

Setelah tahapan implementasi data lagu selesai, lalu dilanjutkan dengan penerapan metode yang digunakan yaitu *Item-Based Collaborative Filtering*. Pada tahapan ini, seluruh data yang telah di-*input*-kan pada tahapan implementasi dibagi menjadi dua, yaitu :

1. Data Latih

Data latih yaitu data yang belum diproses dan akan melalui proses dari *Item-Based Collaborative Filtering*. Data yang diambil untuk diproses yakni 90% dari keseluruhan.

2. Data Uji

Data uji yaitu data yang akan dibandingkan dengan data latih.

Data yang akan diproses yaitu sisa dari data latih yakni 10%. Setelah data dibagi menjadi dua yakni data latih dan data uji proses yang akan dilakukan yaitu penerapan dari *Item-Based Collaborative Filtering*.

d. Menghitung *Similarity*

Tahap ini adalah tahap untuk menghitung *similarity* dari matriks *group-rating* dan matriks *item-rating* dan menggabungkan hasil *similarity* untuk perhitungan prediksi *rating*. Dasar perhitungan untuk kedua *similarity* tersebut dengan menggunakan metode *adjusted-cosine similarity* dan dilakukan untuk masing-masing matriks.

e. Menghitung Prediksi untuk Suatu *Item*

Setelah menetapkan jumlah *similarity* lalu dilanjutkan dengan menghitung prediksi. *Weighted sum* digunakan untuk menghitung prediksi *rating* pada *item* yang belum dirating.

f. Prediksi Sistem

Setelah data latih melewati serangkaian proses maka akan menghasilkan prediksi sistem. Dimana prediksi sistem ini nilai akurasi belum mutlak didapat karena belum melalui perbandingan dengan data uji. Untuk

mengetahui akurasi *rating* maka harus melewati tahapan berikutnya yakni pengukuran akurasi prediksi *rating*.

g. Pengukuran Akurasi Prediksi *Rating*

Untuk mengetahui akurasi prediksi *rating* maka dilakukan pengukuran data antara data uji dan data latih yang telah melalui serangkaian perhiungan dan menghasilkan prediksi sistem. Pengukuran akurasi prediksi system ini menggunakan perhitungan *MAE (Mean Absolute Error)*

h. Hasil Rekomendasi Lagu (Sistem)

Setelah proses perhitungan dengan menggunakan *Item-Based Collaborative Filtering* di dapat dan diterapkan pada perangkat lunak, maka sistem rekomendasi lagu bisa ditampilkan kepada *user*. Data yang ditampilkan berupa lagu dan artis yang direkomendasikan oleh *user* lain setelah melalui proses *rating*.

i. Pengembangan Perangkat Lunak

Untuk mendukung metode dari *Item-Based Collaborative Filtering* dalam sistem rekomendasi tanggal lagu digunakan pendekatan perangkat lunak berbasis objek. Secara mendetail akan dibahas pada metode penelitian (3.3).

3.2 Fokus Penelitian

Penelitian ini memfokuskan pada tingkat akurasi prediksi yang dihasilkan oleh sistem, dimana proses pembuatan prediksi dilakukan dengan menghitung nilai kemiripan antara *item* dengan menggunakan persamaan *adjusted cosine*

(persamaan 2.1). Setelah nilai kemiripan antara *item* didapatkan, maka selanjutnya digunakan persamaan *weighed sum* (persamaan 2.2) untuk menghasilkan nilai prediksi untuk tiap *item*.

Untuk mengukur tingkat akurasi prediksi yang dihasilkan oleh sistem, digunakan persamaan *mean absolute error* (persamaan 2.3).

3.3 Metode Penelitian

3.3.2 Metode Pengembangan Perangkat Lunak

Dalam proses pengembangan perangkat lunak ini, pendekatan yang digunakan adalah pendekatan berorientasi objek, dimana dalam paradigma ini domain permasalahan diabstraksikan sebagai suatu set objek yang mempunyai atribut dan perilaku tertentu.

Pada paradigma berorientasi objek ini, ada beberapa konsep yang harus diketahui, yaitu :

1. *Class* dan *Object*

Class merupakan model yang berisi kumpulan *attribute* dan *method* dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh *class* manusia memiliki *attribute* berat, tinggi, usia kemudian memiliki *method* makan, minum, tidur. *Method* dalam sebuah *class* dapat merubah *attribute* yang dimiliki oleh *class* tersebut. Sebuah *class* merupakan dasar dari modularitas dan struktur dalam pemrograman berorientasi object.

Sedangkan *Object* merupakan perwujudan dari *class*, setiap *object* akan mempunyai *attribute* dan *method* yang dimiliki oleh *class*-nya, contohnya:

amir, ahmad, yani merupakan *object* dari *class* manusia. Setiap *object* dapat berinteraksi dengan *object* lainnya meskipun berasal dari *class* yang berbeda.

2. *Attribute*

Adalah berbagai variabel yang mengitari *class*, dengan nilai datanya bisa ditentukan di *object*.

3. *Operations, Method, dan Services*

Setiap *object* membungkus data (yang direpresentasikan dalam suatu koleksi *attribute*) dan algoritma yang akan mengolah data tersebut.

Algoritma-algoritma tersebutlah yang dimaksud dengan *operations, method* atau *services*.

4. *Messages*

Suatu *class* harus berinteraksi dengan *class* lainnya untuk mencapai suatu tujuan tertentu. *Messages* ini memungkinkan *object* untuk menstimulasi *object* lainnya untuk melakukan suatu *behavior* tertentu.

5. *Encapsulation, Inheritance, dan Polymorphism*

Ketiga hal ini merupakan karakteristik dari paradigma berorientasi objek, *Encapsulation* yaitu merupakan suatu mekanisme untuk menyembunyikan atau memproteksi suatu proses dari kemungkinan interferensi atau penyalahgunaan dari luar sistem dan sekaligus menyederhanakan penggunaan sistem tersebut.

Inheritance merupakan konsep mewariskan *attribute* dan *method* yang dimiliki oleh sebuah *class* kepada *class* turunannya (*subclass*). Dengan

konsep ini *class* yang dibuat cukup mendefinisikan *attribute* dan *method* yang spesifik didalamnya, sedangkan *attribute* dan *method* yang lebih umum akan didapatkan dari *class* yang menjadi induknya.

Polymorphism merupakan konsep yang memungkinkan digunakannya suatu *interface* yang sama untuk memerintah suatu *object* agar melakukan suatu tindakan yang mungkin secara prinsip sama tetapi secara proses berbeda.

Untuk pemodelan perangkat lunak berorientasi objek, digunakan UML (*Unified Modeling Language*) yang merupakan bahasa standar yang digunakan untuk memvisualisasikan dan menjelaskan artefak dari proses analisis dan desain berorientasi objek. UML menyediakan standar notasi dan diagram-diagram yang bisa digunakan untuk memodelkan sistem.

Diagram-diagram pada UML terbagi kedalam 3 klasifikasi, yaitu :

1. *Behavior Diagrams*

Jenis diagram yang menggambarkan perilaku fitur dari sistem atau proses bisnis. Diagram-diagram yang termasuk dalam klasifikasi ini adalah *activity diagram*, *state machine diagram*, *use case diagram*, dan ke 4 subset dari *interaction diagrams*.

2. *Interaction Diagrams*

Sebuah subset dari diagram perilaku yang menekankan pada interaksi antar objek. Diagram-diagram yang termasuk dalam klasifikasi ini adalah *communication diagram*, *interaction overview diagram*, *sequence diagram*, dan *timing diagrams*.

3. *Structure Diagrams*

Jenis diagram yang menggambarkan unsur-unsur yang harus ada pada sistem. Diagram-diagram yang termasuk dalam klasifikasi ini adalah *composite structure diagram*, *component diagram*, *deployment diagram*, *object diagram*, dan *package diagrams*.

3.3.2 Model Proses

Perangkat lunak yang akan dikembangkan dalam penelitian ini merupakan perangkat lunak berbasis *web*, sehingga untuk model proses yang akan digunakan adalah *Web Engineering Process* (Roger S. Pressman, 2004).

Web Engineering (WebE) merupakan suatu proses yang digunakan untuk menciptakan aplikasi berbasis *web* yang berkualitas tinggi. Konsep dasar dan prinsip dari WebE secara umum tidak jauh berbeda dengan *Software Engineering*.

Menurut Roger S. Pressman (2010), ada beberapa karakteristik yang membedakan perangkat lunak berbasis *web* (*WebApp*) dengan perangkat lunak lainnya, yaitu :

1. *Network Intensiveness*

Sebuah *WebApp* berada pada suatu jaringan dan harus melayani kebutuhan dari klien yang berbeda-beda. *WebApp* ini bisa berada di internet sehingga setiap orang bisa mengaksesnya, atau intranet (hanya satu atau beberapa organisasi yang bisa mengaksesnya).

2. *Concurrency*

Sejumlah besar pengguna dapat mengakses dan menggunakan *WebApp* pada satu waktu secara bersamaan.

3. *Unpredictable load*

Jumlah pengguna suatu *WebApp* bisa berbeda-beda setiap harinya, dan itu tidak bisa diprediksi. 100 pengguna datang pada hari senin, tapi mungkin saja waktu hari selasa bisa 100.000 pengguna yang datang.

4. *Performance*

Jika pengguna *WebApp* harus menunggu terlalu lama (untuk mengakses, pemrosesan di sisi *server* atau sisi klien) maka pengguna mungkin akan memutuskan untuk meninggalkan *WebApp* tersebut.

Maka *WebApp* dituntut harus mempunyai performa yang baik.

5. *Availability*

Walaupun harapan untuk 100 persen *availability* (ketersediaan) tidak masuk akal, tetapi pengguna dari *WebApp* populer sering menuntut ketersediaan selama 24/7/365.

6. *Data driven*

Fungsi utama dari sebagian besar *WebApp* adalah dengan menggunakan *hypermedia* untuk menyajikan teks, grafis, audio, dan video ke pengguna. Selain itu, *WebApp* juga digunakan untuk mengakses informasi yang berada pada *database*.

7. *Content sensitive*

Kualitas dan estetika dari konten tetap menjadi penentu penting dari kualitas sebuah *WebApp*.

8. *Continous Evolution*

Tidak seperti aplikasi perangkat lunak konvensional yang berkembang melalui serangkaian perencanaan, *WebApp* berkembang secara terus menerus.

9. *Immediacy*

Meskipun *immediacy* (kebutuhan mendesak untuk perangkat lunak segera dirilis dan disebar ke publik) adalah karakteristik dari banyak domain perangkat lunak, tetapi *WebApp* sering menunjukkan bahwa waktu untuk segera rilis bisa dalam hitungan hari atau minggu.

10. *Security*

Karena *WebApp* tersedia dalam suatu jaringan, diperlukan penerapan keamanan yang lebih demi berjalan baiknya suatu *WebApp*.

11. *Aesthetics*

Salah satu komponen penting dan menjadi daya tarik suatu *WebApp* yaitu desain tampilan dan tata letak konten yang estetik.

Berdasarkan karakteristik khas yang dimiliki oleh suatu perangkat lunak berbasis *web*, maka diperlukan juga teknik dan metode yang tepat dalam proses pengembangannya.

Menurut Roger S. Pressman (2004), model proses *Agile* (seperti *Extreme Programming*, *SCRUM*, *Adaptive Software Development*) bisa berhasil diaplikasikan sebagai *WebE* dengan beberapa adaptasi, karena pada intinya *Agile* merupakan metodologi pengembangan perangkat lunak yang berdasar pada pengembangan secara iteratif dan inkremental dengan tempo yang pendek. Hal

tersebut juga merupakan hal yang akan ditemui dalam pengembangan suatu *WebApp*, yaitu :

- Perangkat lunak akan sering dirilis secara bertahap.
- Perubahan akan sering terjadi secara berkala.
- Mempunyai *timeline* yang pendek untuk setiap tahap dalam proses.

Secara umum, tahap-tahap proses pengembangan perangkat lunak berbasis *web* dengan model proses WebE terdiri dari 5 tahap (Roger S. Pressman, 2004), yaitu :

1. *Formulation/Business Analysis*

Formulation merupakan tahap mencari seluruh kebutuhan dari *WebApp* dan dengan melibatkan seluruh *stakeholders*, bertujuan untuk menjelaskan permasalahan yang harus diselesaikan oleh *WebApp* dengan informasi kebutuhan yang tersedia.

Business Analysis mendefinisikan konteks dari *business* untuk *WebApp*, yaitu pengidentifikasian *stakeholder*, memprediksi kebutuhan *WebApp*, *database*, pendefinisian fungsi.

2. *Planning*

Tahap perencanaan untuk *WebApp* dimana perencanaan berisi definisi dari tiap pekerjaan, lalu jadwal dari jangka waktu yang telah diproyeksikan untuk pengembangan *WebApp* secara inkremental.

3. *Modeling*

Desain dan analisis rekayasa perangkat lunak secara konvensional diadaptasi pada proses pengembangan *WebApp*, digabungkan dan lalu

menyatu pada tahap *modeling* ini. Tahap ini bermaksud untuk menghasilkan suatu analisis dan desain model yang telah didefinisikan pada *requirement*.

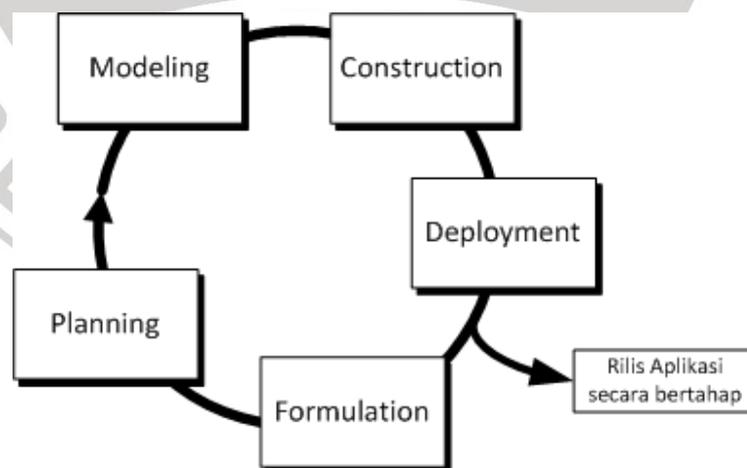
4. *Construction*

Tahap implementasi, dimana *tools* dan teknologi diterapkan untuk membangun *WebApp* yang telah sebelumnya dimodelkan. Setelah itu *WebApp* yang telah dibangun secara bertahap ditest untuk menemukan kesalahan yang terjadi pada tahap desain (isi, arsitektur, antarmuka).

5. *Deployment*

WebApp dikonfigurasi, dirilis terhadap pengguna, dan tahap evaluasi dilakukan. Pada tahap ini, *feedback* dari pengguna dijadikan acuan untuk pengembangan tahap berikutnya.

Kelima alur proses WebE tadi diaplikasikan secara berulang dan bertahap seperti pada ilustrasi dibawah ini.



Gambar 3.2 Model Proses WebE (Roger S. Pressman, 2004)

3.4 Alat dan Bahan Penelitian

3.1.1 Alat Penelitian

Penelitian ini menggunakan seperangkat sistem komputer beserta perangkat lunak pendukungnya, yaitu :

1. Satu unit komputer dengan spesifikasi :
 - a. *Processor* Intel(R) Atom(TM) CPU N455 @ 1.66 GHz (2 CPUs)
 - b. RAM 1 GB
 - c. *Harddisk* 320 GB dgn *free space* 51GB.
 - d. Layar monitor dengan resolusi 1024 x 600 pixel, 32 Bit Color
 - e. *Mouse* dan *Keyboard*
2. Perangkat lunak :
 - a. Sistem Operasi Windows XP SP 2
 - b. XAMPP 1.7.2
 - c. Sublime Text 3 dan Notepad++
 - d. Web Browser (Firefox, Google Chrome, Opera, Safari)
 - e. *Framework* Codeigniter

3.4.2 Bahan Penelitian

Data lagu berasal dari situs <http://www.last.fm> dengan memanfaatkan API-nya yang bisa diakses di <http://www.last.fm/api>. Lalu, data *rating* berasal dari *user* yang telah merating pada sistem.