

1. BAB I

PENDAHULUAN

1.1 Latar Belakang

Sistem rekomendasi memiliki peranan penting dalam bidang *e-commerce* dalam membantu *user* memilih produk dari sejumlah produk yang banyak (Sarwar et al, 2000), contoh sistem rekomendasi yaitu sistem rekomendasi musik yang memberikan rekomendasi ke *user* berdasarkan preferensi musik dan *rating* (Su et al, 2013). Seiring pertumbuhan jumlah musik, sistem rekomendasi pada umumnya hanya menggunakan *rating* dengan 1 kriteria akan mendapatkan masalah berupa *rating diversity* yang menyebabkan hasil rekomendasi tidak begitu dapat diandalkan (Su et al, 2013). Hasil penelitian mengatakan bahwa menerapkan *rating* pada lebih dari satu kriteria (misalnya *like* makanan, *like* dekorasi, *like* pelayanan) kurang disukai oleh *user*, meskipun bisa memberikan hasil rekomendasi yang lebih baik (Konstan, J.A. & Riedl, J., 2012). Penggunaan sistem *tagging* yang semakin umum ternyata mampu memberikan hasil yang mirip dengan sistem *rating* multi kriteria (Konstan, J.A. & Riedl, J., 2012). Permasalahan yang dihadapi *user* adalah *rating* dengan multi kriteria tidak dapat diandalkan untuk mendapatkan jenis musik yang disukai *user*, oleh karena itu dibutuhkan suatu sistem rekomendasi yang mampu merekomendasikan musik yang mirip dengan musik kesukaan *user* secara spesifik bukan hanya melalui sistem *rating* multi kriteria tetapi memaksimalkan fungsi *tag* yang ada pada musik.

Sistem rekomendasi musik yang sudah ada saat ini umumnya menerapkan *collaborative filtering (CF)* (Su, X. & Khoshgoftaar, T.M., 2009). Produk baru yang dimasukkan ke dalam sistem akan segera diberikan *rating* sesuai dengan *rating* yang diberikan oleh sekelompok orang pada produk yang sejenis (Lekakos, G. & Caravelas, P., 2008). Berbeda dengan *CF*, *Recommender by Tag-driven Item Similarity (RTIS)* adalah metode rekomendasi pada musik yang menggunakan *Playcount* (seberapa seringnya sebuah musik dimainkan) sebagai *Implicit Rating* dan *Item Tags* sebagai parameter rekomendasi, dibandingkan dengan *CF* pada umumnya yang menggunakan *rating* secara eksplisit (Su et al, 2013). Metode RTIS

menghitung similaritas antar musik berdasarkan jumlah varian dan frekuensi kemunculan *Social Tag* (Su et al, 2013). Berdasarkan perbandingan nilai *Root-Mean-Square-Error* (RMSE), metode RTIS memberikan hasil yang lebih baik dari *CF* (Su et al, 2013). Hal ini juga disebabkan oleh karena *CF* memberikan hasil yang kurang baik ketika *rating* yang diberikan tidak konsisten antara user dan item (Su et al, 2013).

Berdasarkan uraian tersebut, maka dilakukan penelitian untuk mengimplementasi metode RTIS dalam memberikan rekomendasi, dalam implementasinya akan ditanamkan fitur *Push Notification*, penelitian ini akan dituangkan ke dalam Tugas Akhir dengan judul “**Sistem Rekomendasi Musik Dengan Metode Tag-driven Item Similarity Berbasis Mobile dan Web**”.

1.2. Rumusan Masalah

Berdasarkan penjelasan pada latar belakang, masalah dalam penulisan tugas akhir ini adalah kesulitan *user* untuk mendapatkan musik yang disukai.

1.3. Tujuan

Tujuan dari penulisan tugas akhir ini adalah mengembangkan aplikasi yang akan menghasilkan rekomendasi berbasis *mobile* dan *web* dengan mengimplementasi metode RTIS.

1.4. Manfaat

Manfaat tugas akhir ini adalah sebagai berikut:

1. Memberikan pilihan sistem rekomendasi dengan metode RTIS kepada platform streaming musik untuk dapat menerapkan metode ini ke dalam sistem pihak yang bersangkutan.
2. Mempermudah pengguna untuk dapat mendengarkan musik yang disukai sesuai dengan *tag* dan *playcount* milik pengguna tersebut.
3. Dapat dijadikan bahan referensi untuk diteliti di sistem rekomendasi bidang lainnya.

1.5. Batasan Masalah

Untuk mencegah pembahasan keluar dari topik dan mencegah adanya perluasan masalah, maka dilakukan pembatasan masalah sebagai berikut:

1. *Dataset* yang digunakan berasal dari *offline dataset* milik Last.fm (<https://labrosa.ee.columbia.edu/millionsong/lastfm>, 2011). Berikut rincian *dataset*:

- a. 943.347 *tracks*
- b. 522.366 *unique tags*

Kemudian dilakukan *sampling* dengan teknik *Quota Sampling*. Karena *dataset* belum memiliki data *user*, maka *dataset* akan digabungkan dengan *offline dataset* lainnya (<http://last.fm>, 2009). Berikut rincian isi *dataset* setelah proses *sampling* dan penggabungan:

- a. 947 *tracks*
- b. 807 *users*
- c. 904 *artists*
- d. 7962 *unique tags*

2. Sistem operasi yang digunakan pada *platform mobile* adalah Android dengan Minimal *SDK* yang digunakan adalah *SDK API Level 21* yaitu Lollipop atau Android 5.0 ke atas.

3. Platform website menggunakan *browser* Chrome dengan minimal version 66.0.3359.181 (64-bit).

1.6. Metodologi Pengembangan Sistem

Metodologi yang digunakan dalam pengembangan sistem ini adalah metodologi *Waterfall* versi Sommerville (Sommerville, 2011), yaitu:

1. *Requirement analysis and definition*

- a. Menggambar aliran proses pada sistem dengan *Flowchart* dengan *web editor* Draw.io. *Flowchart* adalah sebuah diagram alur untuk merepresentasikan workflow dari suatu proses atau aktivitas dari suatu sistem.

- b. Menggambar interaksi antara pengguna dan sistem dengan *Use Case Diagram* menggunakan *web editor* Draw.io. *Use Case Diagram* adalah sebuah diagram yang digunakan untuk menggambarkan secara ringkas siapa yang menggunakan sistem dan apa saja yang dapat dilakukannya.
 - c. Menganalisis kebutuhan non fungsional dengan menggunakan Diagram PIECES (*Performance, Information, Economic, Control, Efficiency, Service*).
 - d. Memberlakukan teknik *quota sampling* pada dataset untuk keperluan pengujian.
2. *System and software design*
- Pada tahapan perancangan sistem, dilakukan perancangan tampilan untuk melakukan input dan output dalam merekomendasi item dan juga perancangan *User Interface (UI)* dalam bentuk *Mock-up* dengan aplikasi Balsamiq Mockup 3 yang selanjutnya akan digunakan pada sistem. Selain melakukan perancangan tampilan, dilakukan perancangan model basis data (*database*) dalam bentuk *Entity Relationship Diagram* dengan web editor Draw.io.
3. *Implementation*
- Pengembangan aplikasi ini akan menggunakan beberapa *tools* seperti:
- a. Pengembangan aplikasi dalam *platform website* akan digunakan bahasa pemrograman Python dengan *framework* Django Versi 2.0 dan NodeJS versi 8 untuk *backend*-nya dan ReactJS untuk *frontend*-nya.
 - b. Pengembangan aplikasi dalam *platform mobile* akan digunakan bahasa pemrograman Kotlin pada *Android Studio* 3.1.1
 - c. *Database Management System* yang digunakan pada aplikasi ini adalah MySQL dengan Django sebagai ORM (*Object Relational Mapping*)
4. *Integration and system testing*:
- Pengujian yang dilakukan meliputi:
- a. Pengujian kepuasan penggunaan system pada aplikasi rekomendasi musik menggunakan kuesioner dengan *Technology Acceptance Model (TAM)* yang akan disebarakan melalui *Google Forms*

- b. Pengujian sistem rekomendasi RTIS dengan perhitungan RMSE (Root Mean Square Error)



UNIVERSITAS
MIKROSKIL