



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB II

### LANDASAN TEORI

#### 2.1 Sistem Rekomendasi

Sistem rekomendasi merupakan sebuah alat dan teknik yang menyediakan saran terkait suatu hal untuk dapat dimanfaatkan oleh *user* (Ricci dkk., 2011). Penerapan sistem rekomendasi di dalam sebuah sistem biasanya melakukan prediksi suatu item, seperti rekomendasi film, musik, buku, dan lain sebagainya yang menarik *user*. Sistem ini berjalan dengan mengumpulkan data dari *user* dan dapat dilakukan dengan dua cara (Sanjung dkk., 2011), yaitu secara eksplisit dan implisit:

1. Secara eksplisit dapat dilakukan misalnya dengan meminta *user* untuk merating sebuah item dari range tertentu, memberikan ranking berdasarkan dari item yang disukai dan yang tidak disukai oleh *user*, ataupun dengan meminta *user* untuk melakukan list terhadap item-item yang disukai
2. Implisit dilakukan dengan mengamati pola kecenderungan *user* (behavior *user*) melalui observasi social network atau dengan mengamati item-item yang sering dilihat atau didengar oleh *user*. Dengan kata lain, jika secara eksplisit *user* memberikan penilaian preferensi terhadap item secara langsung, sedangkan implisit tidak secara langsung

Dari data hasil yang dikumpulkan tersebut, kemudian diolah dengan menggunakan algoritma tertentu. Setelah itu, hasil tersebut akan dikembalikan kepada *user* sebagai sebuah rekomendasi *item* dengan parameter dari *user* tersebut.

Prediksi informasi minat pengguna dapat diperoleh berdasarkan pola aksi perilaku *user* atau sering dikatakan sebagai profil *user*. Salah satu cara yang

dilakukan untuk membentuk profil *user* adalah ketika *user* mendengarkan musik pada sebuah aplikasi maka data musik ini akan disimpan sebagai profil pengguna. Berdasarkan profil pengguna yang telah terbentuk akan diketahui minat musik yang disukai oleh *user*.

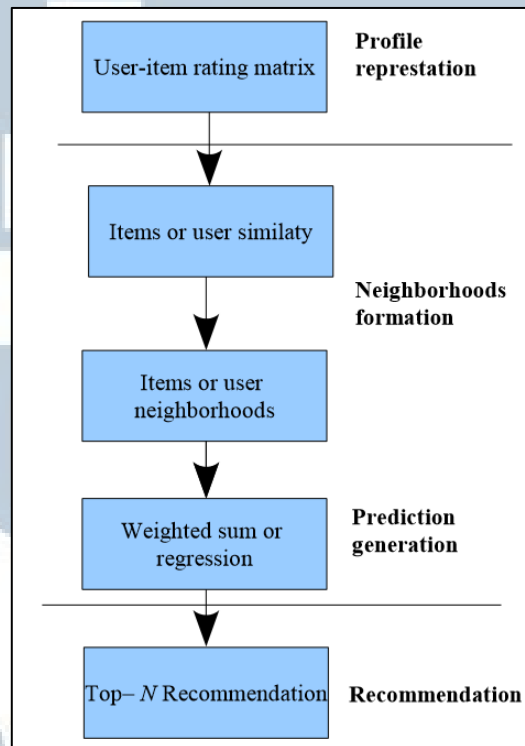
Pada sistem rekomendasi, terdapat beberapa item yang akan disaring untuk direkomendasikan kepada *user* berdasarkan profil *user*, skala *rating*, dan sebagainya sehingga menghasilkan beberapa item yang direkomendasikan kepada *user*. Dalam kategori musik sudah banyak aplikasi yang menggunakan sistem rekomendasi seperti Spotify, Joox, Apple Music, dan lain-lain.

## 2.2 Collaborative Filtering

*Collaborative filtering* merupakan salah satu pendekatan terbaik untuk membangun sistem rekomendasi (Su & Khoshgoftaar, 2009). *Collaborative filtering* bisa digunakan untuk melakukan *filtering* pada data yang akan diproses pada sistem rekomendasi. Tujuan dari metode *collaborative filtering* adalah menyarankan *item* baru untuk *user* tertentu berdasarkan preferensi pengguna sebelumnya dan pengguna lain yang berpikiran serupa. *Collaborative filtering* membutuhkan profil *user* sebagai masukan. Profil *user* diwakili oleh nilai pilihan *user* untuk beberapa *item* yang disebut *rating*.

Terdapat proses penting dalam *collaborative filtering*, yaitu *profile representation*, *neighborhood formation* dan *prediction generation*. *Profile representation* adalah pembentukan *user-item rating matrix*. *Neighborhood formation* adalah proses dari sistem rekomendasi untuk mencari kesamaan antar *item*. Setelah kesamaan antara *item* dihitung, *neighborhood* dapat ditemukan dan dicari prediksi *item* (Sarwar dkk., 2001). *Top-N recommendation* merupakan *output*

dari sistem rekomendasi, dimana  $N$  dapat berupa bilangan bulat positif yang ditentukan oleh sistem. *User* membuat *feedback* dengan mengklik nilai rasio untuk *item* rekomendasi. *Feedback* dari *user* diperlukan untuk memperbaiki sistem rekomendasi di masa depan. Empat proses utama dalam *collaborative filtering* dapat dilihat pada Gambar 2.1.

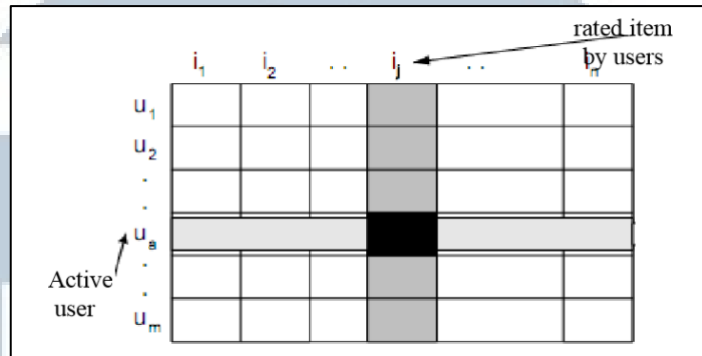


Gambar 2.1 Proses Umum Collaborative Filtering (Saptono, 2010)

### 2.3 User-Item Based Collaborative Filtering

Metode *user-item based collaborative filtering* tidak seperti metode lainnya. Metode ini melewati proses *neighborhood formation* karena sebuah *item* yang belum pernah mendapatkan *rating* sebelumnya tidak dapat dihitung *similarity* atau kesamaannya dengan *item* lainnya. Metode *user-item based collaborative filtering* dapat menghasilkan rekomendasi tanpa mencari formasi ketetanggaan karena menggunakan algoritma *missing value* untuk menghitung prediksi *rating* dan

mampu merekomendasikan *item* baru pada sistem. Pada metode ini dihasilkan sebuah matriks oleh pengguna  $u$  yang memberi *rating* pada *item*  $i$  (Saptono, 2010).



Gambar 2.2 Matriks Rating User-Item (Saptono, 2010)

Matriks pada Gambar 2.2 dianggap sebagai rancangan acak tak lengkap kemudian nilai yang hilang dapat dihitung langsung dengan menggunakan algoritma *adjusted* MVA. Pada perhitungan prediksi menggunakan *adjusted* MVA hanya memperhitungkan sel yang memiliki *rating*. Perhitungan dilakukan dengan menggunakan Rumus 2.1 sebagai berikut.

$$P(i, j) = \left[ \frac{n * R(i, *) + m * R(*, j) - \left( \frac{R(*, *)}{mn} \right) (mn - sparse)}{(m - 1)(n - 1)} \right] \dots(2.1)$$

Dimana  $n$  menyatakan jumlah *item*,  $m$  menyatakan jumlah pengguna,  $mn$  menyatakan perkalian jumlah *item* dan jumlah pengguna,  $(mn - sparse)$  menunjukkan jumlah sel *rating* yang terisi (*rating* tidak sama dengan nol),  $R(i, *)$  menyatakan jumlah *rating* yang diberikan *user*  $i$ ,  $R(*, j)$  menyatakan jumlah *rating* pada *item*  $j$  dan  $R(*, *)$  menyatakan jumlah semua *rating* pada matriks (Hadi Ati dkk., 2012).

## 2.4 Mean Absolute Error (MAE)

*Mean Absolute Error* (MAE) adalah suatu ukuran penyimpangan rekomendasi dari nilai benar *user-specified* (Sarwar dkk., 2001). Untuk mencari MAE dapat menggunakan Rumus 2.2.

$$MAE = \frac{1}{c} \sum_{i=1}^c |r_{ui} - p_{ui}| \quad \dots(2.2)$$

Dimana  $c$  merupakan jumlah *item* yang direkomendasikan dan telah di-*rating* oleh pengguna,  $p_{ui}$  merupakan nilai prediksi *rating* yang diberikan *user*  $u$  pada *item*  $i$ , dan  $r_{ui}$  adalah nilai *rating* yang sebenarnya. Semakin kecil nilai MAE, semakin akurat sistem dalam memberikan rekomendasi (Hadi Ati dkk., 2012).

## 2.5 Representational State Transfer (REST)

REST merupakan salah satu jenis *web service* yang menerapkan konsep perpindahan antar *state*. *State* dapat digambarkan seperti jika *browser* meminta suatu halaman *web*, maka *server* akan mengirimkan *state* halaman *web* yang sekarang ke *browser*. Bernavigasi melalui *link-link* yang disediakan sama halnya dengan mengganti *state* dari halaman *web*. Begitu pula REST bekerja, dengan bernavigasi melalui *link-link* HTTP untuk melaluikan aktivitas tertentu, seakan-akan terjadi perpindahan *state* satu sama lain. Perintah HTTP yang bisa digunakan adalah fungsi *GET*, *POST*, *PUT* atau *DELETE*. Balasan yang dikirimkan adalah dalam bentuk JSON sederhana tanpa ada protocol pemaketan data, sehingga informasi yang diterima lebih mudah dibaca dan di-*parsing* di sisi *client* (Sibagariang, 2016).

Dalam penerapannya, REST lebih banyak digunakan untuk *web service* yang berorientasi pada *resource* atau orientasi yang menyediakan sumber daya sebagai layanannya dan bukan kumpulan-kumpulan dari aktivitas yang mengolah sumber daya. Bentuk *web service* menggunakan REST *style* sangat cocok digunakan sebagai *backend* dari aplikasi berbasis mobile karena cara akses yang mudah dan hasil data yang dikirimkan berformat JSON sehingga ukuran *file* menjadi lebih kecil (Kurniawan, 2014).

## 2.6 API Spotify

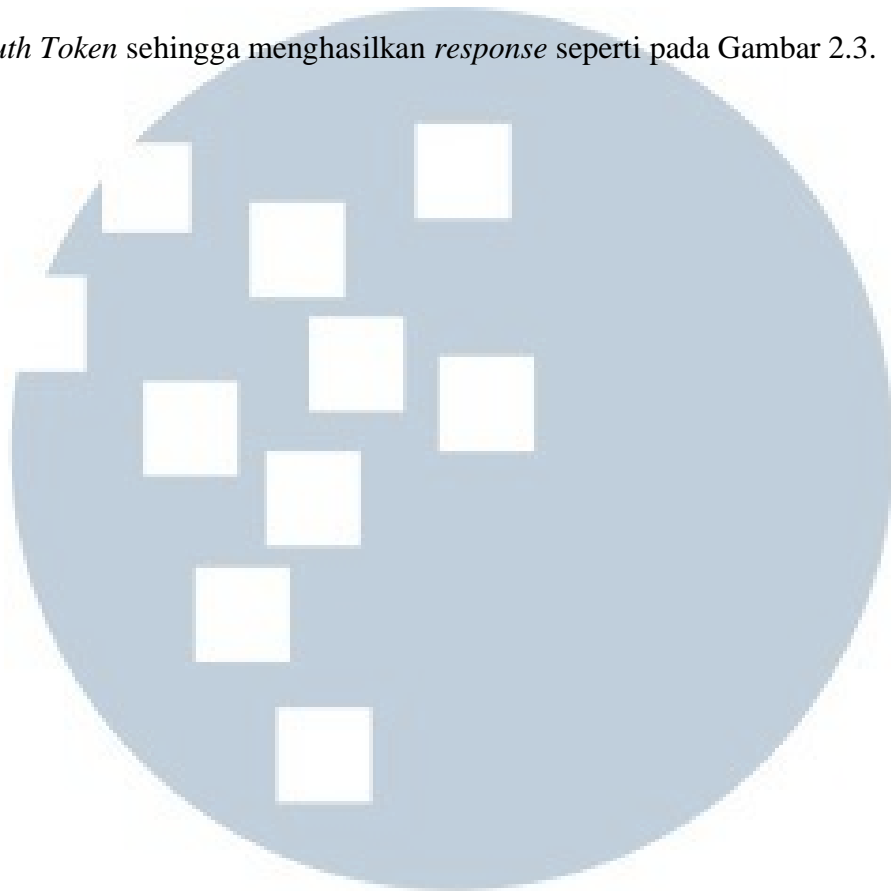
API Spotify menyediakan data lagu yang dapat diambil menggunakan format JSON dan tergolong menjadi beberapa kategori, seperti *Artist*, *Tracks*, *Albums*, dan *Playlists*.

```
"artists" : [ {
  "external_urls" : {
    "spotify" : "https://open.spotify.com/artist/6sFIWsNpZYqfjUpaCgueju"
  },
  "href" : "https://api.spotify.com/v1/artists/6sFIWsNpZYqfjUpaCgueju",
  "id" : "6sFIWsNpZYqfjUpaCgueju",
  "name" : "Carly Rae Jepsen",
  "type" : "artist",
  "uri" : "spotify:artist:6sFIWsNpZYqfjUpaCgueju"
} ],
"disc_number" : 1,
"duration_ms" : 207959,
"explicit" : false,
"external_ids" : {
  "isrc" : "USUM71703861"
},
"external_urls" : {
  "spotify" : "https://open.spotify.com/track/11dFghVXANM1KmJXsNCbN1"
},
"href" : "https://api.spotify.com/v1/tracks/11dFghVXANM1KmJXsNCbN1",
"id" : "11dFghVXANM1KmJXsNCbN1",
"is_playable" : true,
"name" : "Cut To The Feeling",
"popularity" : 66,
"preview_url" : "https://p.scdn.co/mp3-preview/3eb16018c2a700240e9dfb6817b6f2d041f15eb1?cid=8897482848704f2a6f8d7c7",
"track_number" : 1,
"type" : "track",
"uri" : "spotify:track:11dFghVXANM1KmJXsNCbN1"
}
```

Gambar 2.3 Response Get a Track (Spotify Developer, 2017)

Sebagai contoh pada bagian *Track* dapat digunakan fungsi *Get a Track* dengan *method* GET yang berfungsi untuk mengambil data lagu seperti nama artis, judul

lagu, gambar, dan *preview url* dengan menyertakan *Spotify Track ID*, *Market*, dan *Auth Token* sehingga menghasilkan *response* seperti pada Gambar 2.3.



UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA