

# Aplikasi Graf dalam Pemilihan Lagu pada Playlist Spotify

Rafidika Samekto 13519207  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
13519207@std.stei.itb.ac.id

**Abstrak**—Spotify adalah salah satu platform *streaming* musik digital terbesar di dunia. Dengan begitu banyak pengguna, Spotify menggunakan "strategi" untuk memanjakan pengguna, salah satunya adalah membuat playlist berisi lagu rekomendasi untuk pengguna. Data pengguna berupa graf yang direpresentasikan menggunakan matriks ketetanggaan dapat membantu pembuatan playlist-playlist ini dengan memanfaatkan metode *collaborative filtering*



Gambar 1. Logo Spotify

**Kata Kunci**—Playlist, graf, *collaborative filtering*

## I. PENDAHULUAN

Spotify adalah salah satu penyedia jasa *streaming* lagu, video, dan podcast digital yang menyediakan konten dari artis-artis di seluruh dunia. Pada kuartar pertama 2020, Spotify memiliki 286 juta pengguna aktif yang menjadikannya sebagai penyedia jasa *streaming* lagu terpopuler dilihat dari jumlah pendengar aktifnya. Dengan jumlah pelanggan sebanyak itu, Spotify tentu memiliki strategi untuk menggaet pengguna baru dan mempertahankan pengguna lama agar tidak berpindah ke *platform streaming* lagu lain.

Seperti perusahaan teknologi pada umumnya, Spotify mengambil data pengguna seperti lagu apa yang didengarkan, berapa kali lagu itu diputar, dan termasuk genre apa lagu tersebut, untuk kemudian diolah dan dibuatkan playlist sesuai dengan data-data yang diperoleh, khusus untuk pengguna tersebut. Beberapa playlist Spotify yang bekerja dengan cara demikian adalah Release Radar, Daily Mix, dan Discover Weekly.

Release Radar adalah playlist yang berisi kumpulan lagu yang baru saja dirilis oleh artis yang diikuti oleh pengguna, didengar oleh pengguna, dan mungkin disukai oleh pengguna. Playlist yang diperbaharui setiap hari Jumat ini berisi 30 lagu dan dapat berisi lagu yang dipilih oleh artis itu sendiri atau oleh Spotify.

Discover Weekly merupakan playlist lain yang dibuat berdasarkan selera pengguna. Data selera pengguna dapat diperoleh melalui lagu apa yang disukai oleh pengguna, ditandai dengan pengguna menekan gambar hati pada lagu itu, siapa artis yang diikuti oleh pengguna, dan apakah pengguna menambahkan lagu itu ke dalam playlist buatannya.

Pada makalah ini akan dibahas aplikasi graf dalam pemilihan lagu pada dua jenis playlist yang telah disebutkan sebelumnya.

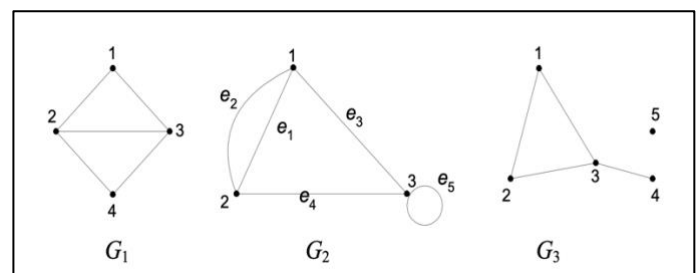
## II. LANDASAN TEORI

### A. Definisi dan Terminologi Graf

Sebuah graf  $G$  didefinisikan sebagai *tuple*  $(V, E)$  dengan  $V$  adalah himpunan seluruh simpul graf yang tidak kosong dan  $E$  adalah himpunan seluruh sisi graf (bagian yang menghubungkan simpul-simpul graf itu) yang tidak kosong. Beberapa terminologi graf adalah:

#### 1. Ketetanggaan (*Adjacent*)

Dua simpul  $V_1$  dan  $V_2$  dikatakan bertetangga jika mereka terhubung langsung oleh suatu sisi  $E$



Gambar 2. Beberapa contoh graf, diambil dari [2]

Sebagai contoh, perhatikan gambar 2. Simpul 1 dan simpul 2 pada  $G_1$  bertetangga karena dihubungkan oleh suatu sisi. Hal yang sama berlaku untuk simpul 2 dan simpul 3 pada  $G_2$ . Jika suatu simpul memiliki sisi gelang, maka simpul tersebut bertetangga dengan simpul itu sendiri, seperti yang diperlihatkan oleh simpul 3 pada graf  $G_2$ .

#### 2. Bersisian (*Incidency*)

Suatu sisi  $E$  bersisian dengan sembarang simpul  $V_1$  jika dan hanya jika sembarang simpul  $V_2$  bertetangga dengan  $V_1$  dan dihubungkan oleh sisi  $E$ .  $E$  dilambangkan dengan sebuah *tuple*

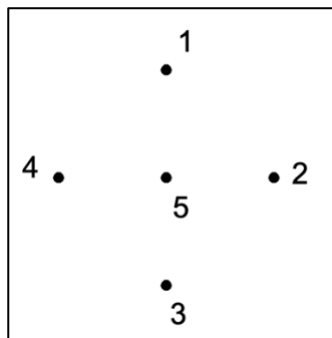
$(V_1, V_2)$ . Contohnya pada gambar 2, sisi  $(3,4)$  pada graf  $G_1$  bersisian dengan simpul 3 dan simpul dan sisi  $(1,2)$  pada graf yang sama tidak bersisian dengan simpul 3.

### 3. Simpul Terpencil (*Isolated Vertex*)

Simpul terpencil adalah simpul yang tidak bersisian dengan simpul mana pun. Dapat dikatakan juga simpul ini tidak memiliki tetangga simpul lain. Contoh simpul ini dapat dilihat pada gambar 2. Simpul 5 pada graf  $G_3$  merupakan simpul terpencil karena ia tidak bersisian dengan sisi manapun.

### 4. Graf Kosong (*Empty Graph*)

Graf kosong adalah graf yang tidak memiliki sisi sama sekali. Contoh dari graf ini dapat dilihat pada gambar 3.



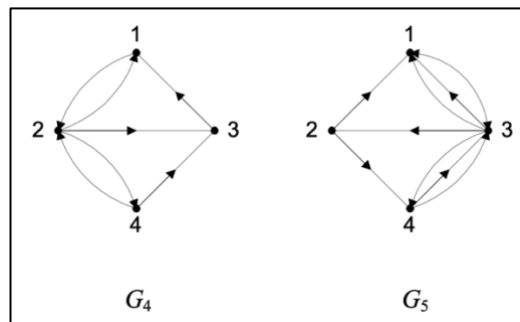
**Gambar 3.** Contoh graf kosong, diambil dari [2]

### 5. Derajat (Degree)

Derajat suatu simpul adalah jumlah sisi yang bersisian dengan simpul tersebut. Derajat simpul  $V_1$  dilambangkan dengan notasi  $d(V_1)$ . Pada gambar 2, simpul 1 dan simpul 4 pada graf  $G_1$  memiliki jumlah derajat yang sama, yaitu  $d(1) = d(4) = 2$ .

Simpul yang memiliki derajat 1, seperti simpul 4 pada graf  $G_3$ , disebut sebagai simpul anting-anting (*pendant vertex*). Simpul yang memiliki derajat 0 merupakan simpul terpencil. Jika suatu simpul bersisian dengan sisi yang menghubungkan simpul itu dengan dirinya sendiri, derajat simpul tersebut bertambah lagi sebanyak 1. Dengan kata lain, sisi tersebut berkontribusi sebanyak 2 derajat pada simpul tersebut. Jika kita melihat gambar 2 kembali, simpul 3 pada graf  $G_2$  memiliki derajat 2 jika gelang (sisi yang mengarah ke simpul itu sendiri) tidak dihitung. Namun, jika gelang itu diperhitungkan, akan didapat derajat 4 pada simpul 3 karena sebuah gelang berkontribusi sebanyak 2 derajat, bukan 1.

Lain halnya dengan graf berarah, derajat graf ini dibedakan menjadi dua, yaitu derajat masuk (dilambangkan dengan  $d_{in}(V_1)$ ) dan derajat keluar (dilambangkan dengan  $d_{out}(V_1)$ ). Derajat masuk adalah jumlah sisi yang "masuk" ke simpul  $V_1$  (ditandai dengan sisi berpanah mengarah masuk ke  $V_1$ ) sementara derajat keluar adalah jumlah sisi yang "keluar" dari simpul  $V_1$  (ditandai dengan sisi berpanah mengarah keluar dari  $V_1$ ).



**Gambar 4.** Contoh graf berarah, diambil dari [2]

Sebagai contoh simpul 4 pada graf  $G_4$  pada gambar 4 memiliki derajat masuk  $d_{in}(4) = 1$  dan memiliki derajat keluar  $d_{out}(4) = 2$ . Sementara itu, simpul 3 pada graf  $G_5$  memiliki derajat masuk  $d_{in}(3) = 2$  dan derajat keluar  $d_{out}(3) = 4$ .

### 6. Lintasan (*Path*)

Lintasan adalah "jalan" yang menghubungkan sembarang simpul  $V_1$  pada suatu graf  $G$  ke suatu simpul sembarang  $V_2$ . "Jalan" ini dapat berupa sebuah sisi yang langsung menghubungkan  $V_1$  dan  $V_2$  atau barisan sisi yang berawal dari  $V_1$  dan berakhir di  $V_2$ . Pada gambar 2, terdapat lintasan yang menghubungkan simpul 1 dan simpul 4 pada graf  $G_3$ . Kedua lintasan itu adalah,  $1,2,3,4$  dan  $1,3,4$ .

Panjang suatu lintasan adalah jumlah sisi yang menghubungkan lintasan itu. Pada contoh sebelumnya, lintasan  $1,2,3,4$  memiliki panjang 3 karena sisi yang menghubungkan lintasan adalah  $(1,2)$ ,  $(2,3)$ ,  $(3,4)$ , sedangkan lintasan  $1,3,4$  memiliki panjang 2 karena sisi yang menghubungkan lintasan adalah  $(1,3)$  dan  $(3,4)$ .

### 7. Siklus (*Cycle*) atau Sirkuit (*Circuit*)

Sirkuit atau siklus adalah lintasan yang berawal dan berakhir di simpul yang sama. Contoh sirkuit adalah lintasan  $1,2,4,3,1$  pada graf  $G_1$  pada gambar 2. Panjang sirkuit memiliki definisi yang sama dengan panjang lintasan, banyaknya sisi yang menghubungkan lintasan (sirkuit) itu.

### 8. Keterhubungan (*Connectivity*)

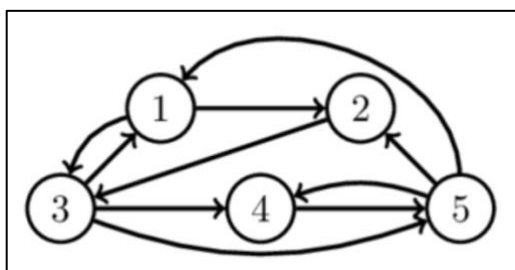
Simpul sembarang  $V_1$  dikatakan terhubung dengan simpul sembarang  $V_2$  jika dan hanya jika terdapat lintasan dari  $V_1$  ke  $V_2$  atau sebaliknya. Contoh yang dapat diambil adalah simpul 1 dan simpul 4 pada graf  $G_1$  pada gambar 2. Karena ada lintasan yang menghubungkan kedua simpul, maka simpul 1 dan simpul 4 dikatakan terhubung.

Sebuah graf dikatakan terhubung jika setiap simpul pada graf terhubung dengan setiap simpul lain pada graf. Graf  $G_1$  pada gambar 2 merupakan contoh graf terhubung karena setiap simpul pada graf terhubung dengan setiap simpul lain di graf. Kebalikan dari graf terhubung adalah graf tak-terhubung, yang dari namanya, merupakan sebutan untuk graf yang tidak terhubung (ada simpul yang tidak terhubung dengan setidaknya satu simpul dalam graf). Contoh ini adalah graf  $G_3$  pada gambar 2, di mana tidak ada simpul lain pada graf ini yang bisa terhubung dengan simpul 5.

Ada sedikit perbedaan masalah keterhubungan untuk graf berarah. Suatu graf berarah  $G$  dikatakan terhubung jika graf tak

berarahnya terhubung. Dengan kata lain, jika kita menghilangkan panah dari graf berarah, kita mendapatkan versi graf tak-berarah dari graf tersebut. Jika versi ini merupakan graf terhubung, maka versi asli dari graf juga merupakan graf terhubung.

Pada graf berarah, jika suatu simpul sembarang  $V_1$  memiliki lintasan ke suatu simpul sembarang  $V_2$  dan sebaliknya, kedua simpul tersebut disebut terhubung kuat. Jika keduanya tidak terhubung kuat, namun versi graf tak berarahnya menunjukkan bahwa kedua simpul terhubung, maka pada versi aslinya, simpul  $V_1$  dan  $V_2$  disebut terhubung lemah.



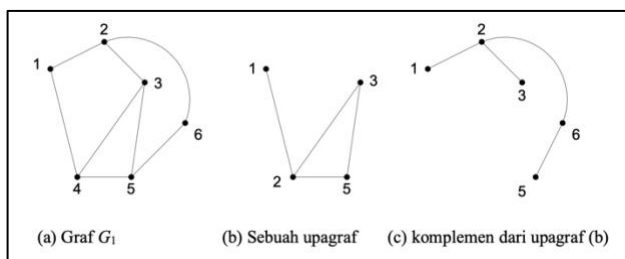
Gambar 5. Contoh graf berarah 2, diambil dari [2]

Pada gambar 5, simpul 1 dan simpul 4 terhubung kuat karena terdapat lintasan dari simpul 1 ke simpul 4 dan lintasan dari simpul 4 ke simpul 1.

Graf berarah  $G$  dikatakan terhubung kuat jika seluruh simpul graf terhubung kuat ke setiap simpul lain. Jika tidak, graf  $G$  disebut terhubung lemah. Graf terhubung pada gambar 5 merupakan graf terhubung kuat karena setiap simpul terhubung kuat dengan simpul lain pada graf.

### 9. Upagraf (Subgraph) dan Komplemen Upagraf

Misalkan sebuah graf  $G$  didefinisikan sebagai  $G = (V, E)$  dan  $G_1$  sebagai  $G_1 = (V_1, E_1)$ .  $G_1$  dikatakan sebagai upagraf dari  $G$  jika  $V_1$  merupakan subset dari  $V$  dan  $E_1$  merupakan subset dari  $E$ . Sementara komplemen dari upagraf  $G_1$  adalah graf  $G_2$  sedemikian sehingga  $E_2 = E - E_1$  dan  $V_2$  berisi simpul-simpul yang bersisian dengan  $E_2$ .



Gambar 6. Contoh graf dengan upagraf beserta komplemennya, diambil dari [2]

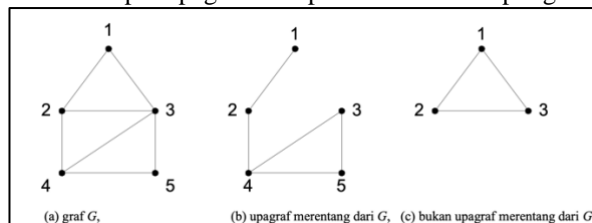
Contoh upagraf dapat dilihat dalam gambar 6. Upagraf dari  $G_1$  ditunjukkan oleh gambar 6(b). Terlihat bahwa seluruh simpul dalam upagraf merupakan bagian dari simpul pada graf  $G_1$ , begitu pula sisi upagraf yang seluruhnya merupakan bagian dari  $G_1$ . Komplemen dari upagraf ditunjukkan oleh gambar 6(c). Di sini, seluruh sisinya merupakan sisi-sisi yang berada di  $G_1$  namun tidak berada pada upagraf.

Upagraf suatu graf dapat merupakan sebuah graf terhubung.

Upagraf seperti ini dinamakan komponen (*connected component*) dari graf tersebut. Versi graf berarah dari komponen mensyaratkan upagraf dari graf tersebut terhubung kuat.

### 10. Upagraf Merentang (Spanning Subgraph)

Upagraf merentang adalah upagraf yang himpunan simpulnya merupakan himpunan graf aslinya juga. Dengan kata lain, seluruh simpul upagraf merupakan seluruh simpul graf.

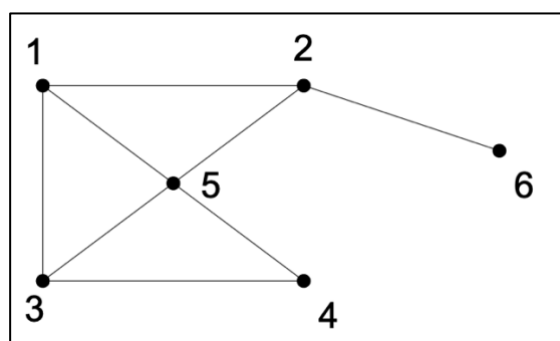


Gambar 7. Contoh upagraf merentang, diambil dari [2]

Contoh upagraf merentang dapat dilihat di gambar 7. Gambar 7(b) menunjukkan upagraf merentang dari  $G_1$  dengan ciri semua simpul upagraf merupakan simpul dari  $G_1$ . Gambar 7(c) bukan merupakan upagraf merentang karena simpul-simpulnya tidak sama dengan  $G_1$  (dengan kata lain, ada simpul yang tidak ada dalam gambar 7(c)).

### 11. Cut-Set

Ini adalah himpunan sisi dari  $G$  yang jika dibuang seluruhnya, menyebabkan  $G$  menjadi graf tak terhubung. Dengan kata lain, *cut-set* memecah  $G$  menjadi dua komponen.



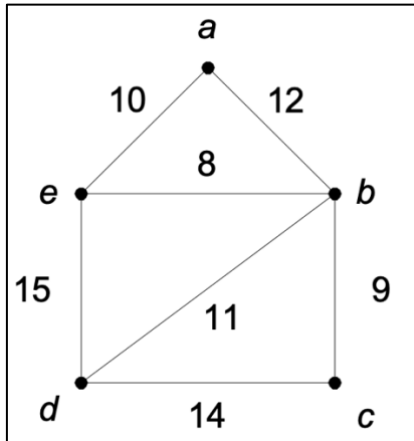
Gambar 8. Contoh graf terhubung  $G_6$ , diambil dari [2]

Misalkan ada graf  $G_6$  seperti yang terlihat pada gambar 8. Himpunan sisi yang menyebabkan  $G_6$  menjadi tak terhubung salah satunya adalah  $\{(1,2), (2,5)\}$ . Perhatikan bahwa terdapat himpunan lain yang dapat memisahkan  $G_6$  menjadi dua komponen. Salah satunya adalah himpunan  $\{(2,6)\}$ .

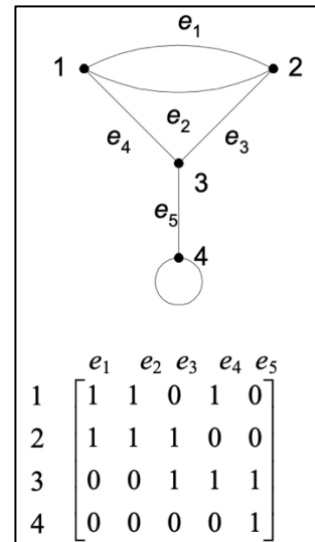
*Cut-set* haruslah berisi sisi-sisi yang perlu dibuang agar graf menjadi tak terhubung. Himpunan ini tidak boleh berisi sisi yang jika sisi itu tidak ada, graf tetap akan terpecah menjadi dua komponen. Dalam graf  $G_6$  pada gambar 8 di atas, salah satu *cut-set*-nya adalah  $\{(1,2), (2,5)\}$ . Himpunan  $\{(1,2), (2,5), (4,5)\}$ , walaupun merupakan subset dari himpunan sebelumnya, bukan merupakan *cut-set* karena sisi  $(4,5)$  tidak diperlukan jika sisi dalam himpunan tersebut akan dihapus dari  $G_6$ .

### 12. Graf Berbobot (Weighted Graph)

Graf berbobot adalah graf yang tiap sisinya diberi bobot. Gambar 9 merupakan contoh graf berbobot.



**Gambar 9.** Contoh graf berbobot, diambil dari [2]



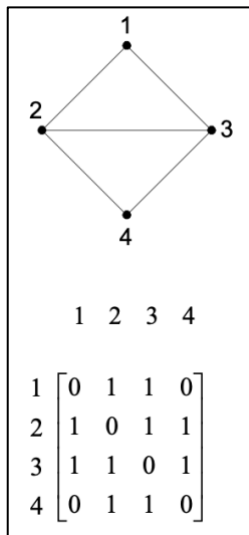
**Gambar 11.** Contoh representasi graf menggunakan matriks bersisian, diambil dari [2]

### B. Representasi Graf

Sebuah graf dapat direpresentasikan menggunakan beberapa cara berikut.

#### 1. Matriks Ketetanggaan

Pembentukan matriks ketetanggaan didasari oleh apakah suatu simpul merupakan tetangga dari simpul lain. Matriks ini berukuran  $n \times n$  dengan  $n$  menyatakan jumlah simpul dalam graf. Suatu elemen pada matriks bernilai 1 jika baris elemen tersebut merupakan tetangga dari kolom elemen tersebut. Selain itu, elemen bernilai 0.



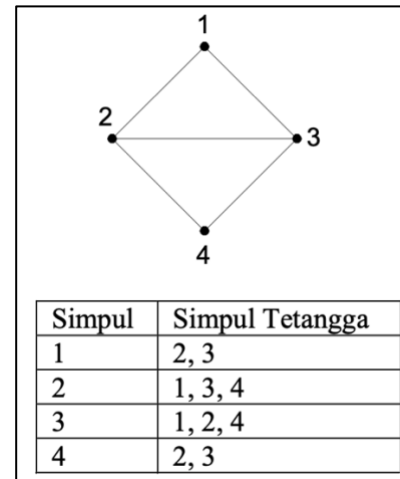
**Gambar 10.** Contoh representasi graf menggunakan matriks ketetanggaan, diambil dari [2]

#### 2. Matriks Bersisian

Mirip dengan matriks ketetanggaan, matriks bersisian menyatakan hubungan suatu simpul pada graf dan sisi yang bersisian dengannya. Elemen matriks bernilai 1 jika baris elemen, yang menyatakan simpul, bersisian dengan kolom elemen, yang menyatakan sisi. Jika tidak, elemen akan bernilai 0.

#### 3. Senarain Ketetanggaan

Cara ini mirip dengan matriks ketetanggaan, hanya saja representasinya menggunakan tabel berkolom 2. Kolom pertama berisi seluruh simpul yang ada pada graf dan kolom kedua berisi simpul-simpul yang bertetangga dengan simpul pada kolom pertama dan sebaris dengannya.



**Gambar 12.** Contoh representasi graf menggunakan senarain ketetanggaan, diambil dari [2]

## III. PENGGUNAAN GRAF DALAM PEMILIHAN LAGU

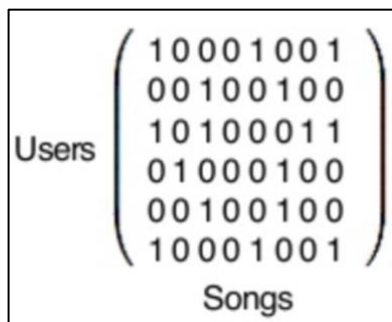
### A. Discover Weekly

Discover Weekly merupakan salah satu playlist yang dibuat Spotify berdasarkan selera penggunanya. Salah satu metode rekomendasi yang digunakan Spotify untuk membuat playlist ini adalah *collaborative filtering*.

*Collaborative filtering*, singkatnya, adalah suatu metode penyocokan aktivitas mendengarkan suatu pengguna dengan pengguna lain, dalam hal layanan *streaming* musik seperti Spotify. Metode ini dapat diilustrasikan sebagai berikut. Misalkan terdapat dua orang A dan B. A menyukai musik berjudul P, Q, R, S, dan T. Sementara B menyukai musik

berjudul Q, R, S, T, dan U. *Collaborative filtering* membandingkan selera kedua orang ini dengan melihat musik apa yang mereka sukai. Dalam contoh tadi, A dan B memiliki kesamaan, yaitu menyukai musik berjudul Q, R, S, dan T. Kemudian, metode ini menganggap bahwa kedua orang tersebut memiliki selera musik yang sangat mirip dan akan merekomendasikan lagu apa yang didengarkan oleh orang lain yang selernya mirip dengan orang itu. Kembali ke contoh tadi, A akan direkomendasikan untuk mendengar lagu berjudul U dan B akan diminta untuk mendengar lagu berjudul P.

Dalam praktiknya, Spotify menerapkan metode *Collaborative filtering* dengan memanfaatkan sebuah matriks yang dapat dianggap juga sebagai matriks ketetanggaan. Representasi graf dengan menggunakan matriks ini dapat dilihat dalam gambar berikut.



**Gambar 13.** Representasi graf dalam matriks ketetanggaan metode *Collaborative filtering*, diambil dari [4]

Gambar 13 merupakan contoh kecil matriks ketetanggaan yang digunakan. Baris matriks yang sebenarnya berjumlah lebih dari 280 juta menyatakan seluruh pengguna Spotify dan kolom matriks yang sebenarnya berjumlah lebih dari 50 juta merepresentasikan lagu yang ada di *server* Spotify. Untuk memudahkan analisis, penulis menganggap elemen 1 dan 0 pada matriks menyatakan apakah pengguna X (yang berada di baris X) menyukai lagu Y (yang berada di kolom Y).

Seperti yang sudah dijelaskan sebelumnya, *collaborative filtering* membandingkan selera pengguna, dalam hal ini lagu apa yang pengguna sukai, dengan orang lain untuk mendapatkan rekomendasi yang sekira sesuai untuk orang itu. Untuk mendapat hasil yang lebih akurat, metode ini tentu tidak membandingkan selera seorang pengguna dengan pengguna lain hanya sekali saja. Karena playlist Discover Weekly terdiri dari 30 lagu pilihan, setidaknya ada 30 pengguna lain yang dibandingkan selera musiknya dengan pengguna itu. Semakin banyak perbandingan yang dilakukan, semakin akurat juga rekomendasi lagu untuk pengguna. Jika dilihat dari matriks, cara membandingkan selera pengguna adalah cukup melihat kolom lagu pengguna A dengan pengguna B. Jika keduanya bernilai 1, maka pengguna A dan B sama-sama menyukai lagu yang sama. Kita dapat menelusuri lebih jauh seberapa banyak lagu yang disukai oleh pengguna A juga disukai oleh pengguna B. Jika cukup banyak, maka dengan suatu kriteria tertentu, pengguna A dan B dapat dikatakan memiliki selera yang mirip sehingga lagu yang direkomendasikan untuk pengguna A kemungkinan besar adalah lagu yang belum disukai olehnya tapi sudah disukai oleh pengguna B. Metode ini berlaku sebaliknya untuk pengguna B.

Ini dilakukan berulang kali hingga setidaknya sudah terkumpul 30 lagu yang cocok dimasukkan ke playlist Discover Weekly khusus untuk pengguna.

### B. Release Radar

Playlist ini berisi lagu-lagu yang baru saja dirilis oleh artis yang diikuti oleh pengguna, lagu artis yang tidak diikuti tapi disukai oleh pengguna, dan lagu yang mungkin disukai oleh pengguna. Dengan alasan kepraktisan, penulis menganggap bahwa lagu yang ada pada playlist terdiri dari 3 lagu oleh artis yang mungkin disukai pengguna, maksimum 15 lagu dari artis yang diikuti, dan 12-27 lagu dari artis yang didengar lagunya, namun tidak diikuti. Jumlah tersebut bergantung pada jumlah artis yang diikuti oleh pengguna.

Langkah pertama untuk menentukan lagu-lagu tersebut adalah melihat kembali matriks ketetanggaan yang digunakan dalam *collaborative filtering* seperti (contoh kecil) yang ditunjukkan oleh gambar 13. Dari matriks tersebut, dapat diperoleh seluruh lagu yang disukai oleh pengguna dan dari lagu-lagu itu, diperoleh artis-artis yang menyanyikannya sekaligus diikuti oleh pengguna. Data nama-nama artis tersebut ditampung untuk kemudian dilihat apakah ada dari artis-artis tersebut yang mengeluarkan lagu baru (bisa dalam bentuk *single* atau album) dalam tujuh hari terakhir terhitung dari hari Jumat, tepat sebelum playlist dibuat. Terakhir, program akan memasukkan lagu baru tersebut ke dalam playlist pengguna.

Sebagai contoh sederhana, ambil data pada baris terakhir matriks pada gambar 13. Elemen yang bernilai 1 pada baris tersebut terdapat pada kolom pertama, kelima, dan kedelapan. Asumsikan artis yang menyanyikan lagu di tiga kolom ini adalah artis yang berbeda. Lalu, akan diperiksa apakah ada setidaknya satu dari ketiga artis tadi yang mengeluarkan lagu baru dalam kurun waktu tujuh hari terakhir. Jika iya, lagu terbaru dari artis tersebut akan dimuat dalam playlist Release Radar pengguna. Jumlah lagu yang dimasukkan berdasarkan artis yang diikuti pengguna ini akan disimpan sebagai acuan untuk menentukan lagu lain dalam playlist.

Penulis sebelumnya menyatakan asumsi bahwa bagian playlist yang berisi lagu dari artis yang diikuti adalah sebesar maksimum 50%. Maksimum memiliki arti bahwa bisa saja artis yang diikuti oleh pengguna berjumlah kurang dari 50% alokasi lagu untuk playlist atau sebanyak 15 lagu (artis). Gambaran mengenai ini dijelaskan dalam contoh sebelumnya. Namun, terdapat pula kondisi saat artis yang diikuti berjumlah lebih dari 15. Jika hal ini terjadi, program akan tetap mengumpulkan seluruh data artis yang diikuti, lalu mereka akan diurutkan berdasarkan jumlah lagu yang disukai oleh pengguna hingga terdapat 15 artis dengan lagu mereka yang paling banyak disukai pengguna.

Misalkan jumlah lagu yang dipilih berdasarkan metode sebelumnya adalah  $n$ , maka jumlah persis lagu  $x$  yang berasal dari artis yang tidak diikuti namun disukai oleh pengguna adalah sebanyak  $30 - n - 3$  lagu. Cara pemilihan lagu berikutnya adalah dengan kembali melihat matriks ketetanggaan dan pada baris yang digunakan sebelumnya. Langkah berikutnya adalah mengeliminasi elemen-elemen bernilai 1 yang artis lagu tersebut sudah ada di playlist Release Radar pengguna. Dengan kata lain, seluruh lagu yang disukai pengguna yang artisnya sudah berada

di playlist tidak akan menjadi perhatian kita. Langkah kedua adalah memeriksa masing-masing lagu yang disukai pengguna dan jika artis dari lagu tersebut belum ada dalam playlist Release Radar, lagu terbaru dari artis tersebut akan dimasukkan ke dalam playlist jika lagu itu baru dirilis dalam waktu yang ditentukan sebelumnya. Langkah ini diulangi sebanyak  $x$  kali. Jika pada suatu kesempatan ternyata  $x$  tidak mungkin dipenuhi, artinya tidak terdapat  $x$  artis berbeda yang lagunya disukai oleh pengguna, maka "jatah" lagu yang seharusnya diisi oleh lagu dari artis yang tidak diikuti pengguna dipindahkan ke artis yang lagunya mungkin disukai oleh pengguna. Seperti contoh, misalkan ada  $r$  artis berbeda yang lagu terbarunya dimasukkan karena artis tersebut tidak diikuti namun disukai lagunya oleh pengguna. Maka, sebanyak  $x - r$  buah lagu akan menjadi jatah untuk lagu baru yang karya artisnya mungkin disukai oleh pengguna. Oleh sebab itu, untuk langkah terakhir dalam pemilihan lagu, pemilihan dilakukan sebanyak  $3 + (x - r)$  kali.

Tahap terakhir dalam pemilihan lagu ini sebenarnya cukup sederhana. Hal pertama yang dilakukan adalah mencari lagu yang mungkin disukai oleh pengguna menggunakan metode *collaborative filtering* seperti yang dilakukan saat membuat playlist Discover Weekly. Setelah lagu tersebut didapat, periksa apakah artis dari lagu tersebut sudah ada pada playlist (kondisi 1). Jika sudah, maka akan dicari kembali lagu lain menggunakan metode yang sama hingga didapat lagu yang artisnya belum ada di playlist Release Radar. Setelah lagu yang cocok didapat, periksa kembali apakah artis tersebut merilis lagu terbarunya dalam jangka waktu yang ditentukan (kondisi 2). Jika tidak, pencarian ini harus diulang dari kondisi 1. Setelah kedua kondisi terpenuhi, lagu terbaru dari artis yang dipilih dimasukkan ke dalam playlist pengguna. Langkah ini diulangi sebanyak  $3 + (x - r)$  kali.

#### IV. KESIMPULAN

Spotify merupakan salah satu platform *streaming* musik digital terbesar di dunia. Penggunaan graf, dalam hal ini direpresentasikan dengan matriks ketetanggaan, dapat membantu Spotify mencatat berbagai data atau aksi yang dilakukan pengguna. Dari data ini, Spotify dapat membuat playlist berisi rekomendasi lagu untuk seorang penggunanya yang salah satu metodenya adalah menggunakan *collaborative filtering*, seperti yang digunakan untuk playlist Discover Weekly.

#### V. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih sebesar-besarnya kepada Tuhan Yang Maha Esa. Penulis juga berterima kasih kepada orang tua penulis, para dosen pengampu mata kuliah IF2120 Matematika Diskrit, dan teman serta kerabat penulis yang telah mendukung penyelesaian makalah ini.

#### REFERENSI

- [1] K. Rosen, *Discrete Mathematics and Its Applications*. New York: McGraw-Hill, 2012.
- [2] R. Munir, *Matematika Diskrit*. Bandung: Departemen Teknik Informatika Institut Teknologi Bandung, 2003.

- [3] <https://medium.com/systems-ai/spotify-s-machine-learning-algorithms-and-your-daily-mix-f49d97db4b16>, diakses pada 8 Desember 2020.
- [4] <https://medium.com/s/story/spotify-s-discover-weekly-how-machine-learning-finds-your-new-music-19a41ab76efe>, diakses pada 8 Desember 2020.
- [5] <https://www.businessofapps.com/data/spotify-statistics/>, diakses pada 8 Desember 2020.
- [6] <https://newsroom.spotify.com/2019-05-02/five-ways-to-make-your-discover-weekly-playlists-even-more-personalized/>, diakses pada 8 Desember 2020.

#### PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 11 Desember 2020



Rafidika Samekto 13519207