

Pemanfaatan Graf Berbobot dalam Pembuatan Playlist Musik

Aditya Bimawan 13519064¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

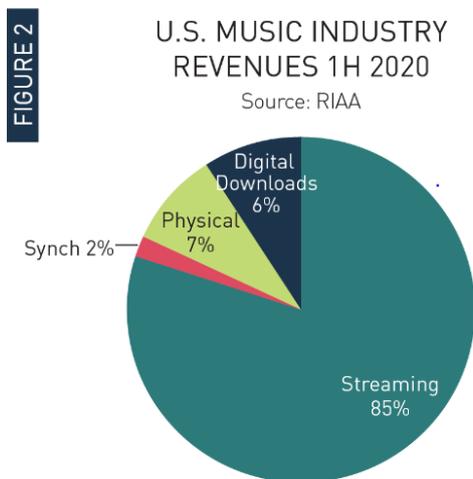
¹13519064@std.stei.itb.ac.id

Abstract—Dalam dunia musik, playlist sangatlah penting. Berbeda dengan album yaitu susunan lagu yang berisi lagu—lagu buatan satu penyanyi, *playlist* dapat berisi lagu dari berbagai penyanyi berbeda. Playlist semakin penting pada zaman digital ini, karena sebagian besar pendengar musik sudah tidak pergi ke toko musik untuk membeli kaset atau CD, sehingga tanpa playlist sangatlah sulit bagi para pendengar untuk menemukan musik baru, ataupun untuk para musisi yang berkembang untuk dapat mengembangkan kepopuleran mereka. Pada makalah ini penulis akan mencoba untuk memberikan suatu metode pembuatan playlist dari sekumpulan data lagu dengan memanfaatkan graf berbobot.

Keywords—Playlist, graf berbobot, pembuatan playlist

I. PENDAHULUAN

Pada zaman sekarang, industri musik tidak seperti beberapa tahun yang lalu. Sebagian besar dari konsumsi musik sekarang dilakukan melalui media elektronik, yaitu *streaming*. Berdasarkan laporan dari RIAA, 85% penghasilan pada industri musik di Amerika Serikat bersumber dari *streaming*, dan hanya 7% dari seluruh penghasilan bersumber dari penjualan media fisik.



Gambar 1. Diagram persentase sumber penghasilan industri musik Amerika Serikat.

Dari data tersebut dapat disimpulkan bahwa media elektronik menjadi sangat penting bagi para pendengar maupun para

musisi. Apabila seorang musisi ingin meningkatkan tingkat kepopuleran mereka, maka mereka harus dapat menjangkau para pendengar melalui media elektronik.

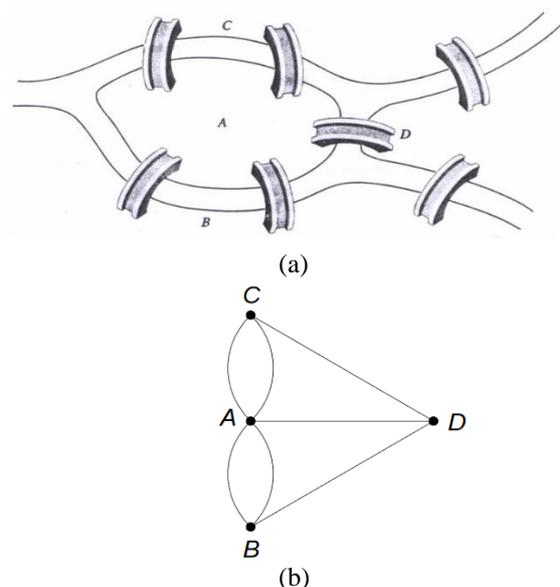
Salah satu cara para pendengar dapat menemukan musik baru adalah melalui *playlist*. *Playlist* adalah daftar atau susunan beberapa lagu, seringkali dari berbagai musisi dan dibuat secara pribadi oleh pendengar. Melalui *playlist*, para pendengar bisa membagikan dan memperkenalkan musik yang sesuai dengan selera mereka kepada orang-orang yang mereka kenal, sehingga meningkatkan kepopuleran musisi-musisi yang ada pada *playlist* tersebut.

Sebuah *playlist* seringkali dibuat secara perlahan-lahan oleh para pendengar, agar lagu-lagu di dalam *playlist* tersebut dapat sesuai dan memiliki alur yang bagus. Tetapi, tidak semua orang memiliki waktu untuk membuat *playlist* mereka sendiri. Maka dari itu, penulis akan mencoba untuk memanfaatkan teori graf untuk membuat metode pembuatan playlist dari sekumpulan data lagu.

II. LANDASAN TEORI

A. Graf

Untuk merepresentasikan hubungan antara beberapa objek berbeda, dapat digunakan graf. Hubungan antar objek digambarkan oleh sisi antar simpul pada graf.



Gambar 2.a. Permasalahan jembatan Königsberg
Gambar 2.b. Representasinya dalam bentuk graf

1. Definisi graf

Graf $G = (V, E)$, dimana

$V =$ Himpunan tidak kosong dari simpul-simpul pada graf

$$= \{ v_1, v_2, \dots, v_n \}$$

$E =$ himpunan sisi (edges) yang menghubungkan sepasang simpul

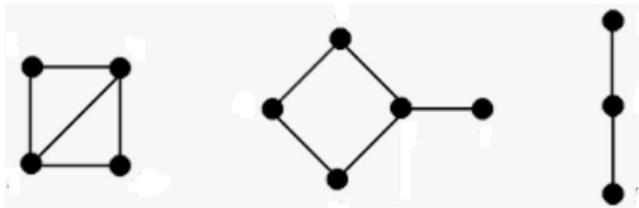
$$= \{ e_1, e_2, \dots, e_n \}$$

Tiap-tiap simpul pada graf dapat diberi nama, misalnya dengan huruf (a, b, c, ...), dengan angka (1, 2, 3, ...), atau dengan nama lainnya.

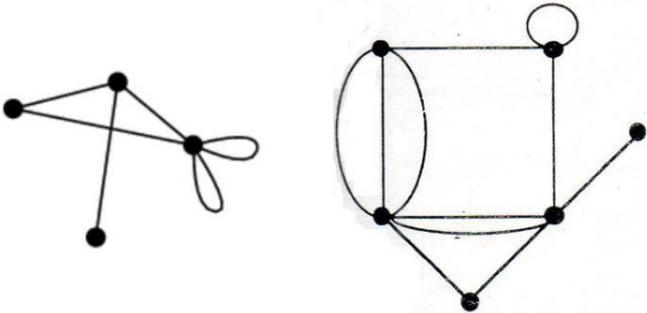
2. Jenis-jenis graf

Berdasarkan ada atau tidaknya gelang atau sisi ganda pada suatu graf, graf digolongkan menjadi dua jenis:

- Graf sederhana (*simple graph*)
Graf yang tidak mengandung gelang maupun sisi ganda.
- Graf tak sederhana (*unsimple graph*)
Graf yang mengandung sisi ganda atau gelang.



(a)



(b)

Gambar 3.a. Contoh graf sederhana

Gambar 3.b. Contoh graf tak sederhana

3. Terminologi graf

- Ketetanggaan
Dua buah simpul dikatakan bertetangga apabila keduanya terhubung langsung oleh sebuah sisi.
- Bersisian
Untuk sembarang sisi $e = (v_j, v_k)$, dikatakan e bersisian dengan v_j , atau e bersisian dengan v_k .
- Simpul terpencil
Simpul yang tidak mempunyai sisi yang bersisian dengannya,
- Graf Kosong
Graf dengan himpunan sisi yang kosong.

- Derajat (*degree*)

Jumlah sisi yang bersisian dengan suatu simpul.

- Lintasan

Lintasan yang panjangnya n dari simpul awal v_0 ke simpul tujuan v_n ialah barisan berselang-seling simpul-simpul dan sisi-sisi yang terbentuk $v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n$ sedemikian sehingga $e_1 = (v_0, v_1), e_2 = (v_1, v_2), \dots, e_n = (v_{n-1}, v_n)$ adalah sisi-sisi dari graf G. Panjang lintasan adalah jumlah sisi dalam lintasan tersebut.

- Siklus

Siklus atau sirkuit adalah lintasan yang berawal dan berakhir pada simpul yang sama. Panjang sirkuit adalah jumlah sisi pada sirkuit tersebut.

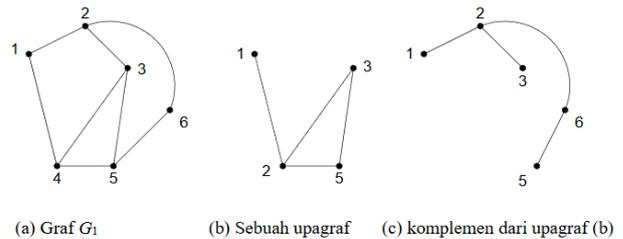
- Keterhubungan

Dua buah simpul v_1 dan v_2 dikatakan terhubung apabila ada suatu lintasan yang menghubungkan dari v_1 ke v_2 .

Suatu graf G dikatakan terhubung (*connected graph*) apabila setiap pasang simpul v_i dan v_j pada G memiliki lintasan yang menghubungkan keduanya.

- Upagraf

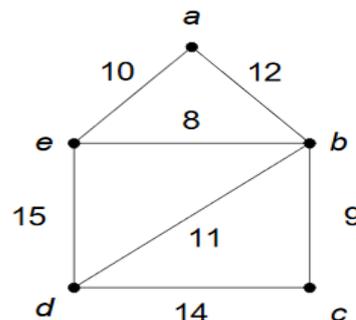
Misalkan $G = (V, E)$ adalah sebuah graf. $G_1 = (V_1, E_1)$ adalah sebuah upagraf dari G jika $V_1 \subseteq V$ dan $E_1 \subseteq E$.



Gambar 4. Contoh upagraf

- Graf Berbobot

Graf yang tiap sisinya diberi harga (bobot).



Gambar 5. Contoh graf berbobot.

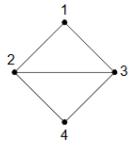
4. Representasi graf

- Matriks Ketetanggaan (*adjacency matrix*)

$$A = [a_{ij}]$$

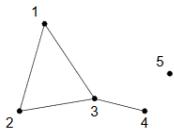
a_{ij} bernilai 1 jika simpul i dan j bertetangga.

Bernilai 0 jika tidak bertetangga.



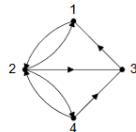
	1	2	3	4
1	0	1	1	0
2	1	0	1	1
3	1	1	0	1
4	0	1	1	0

(a)



	1	2	3	4	5
1	0	1	1	0	0
2	1	0	1	0	0
3	1	1	0	1	0
4	0	0	1	0	0
5	0	0	0	0	0

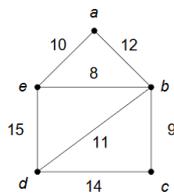
(b)



	1	2	3	4
1	0	1	0	0
2	1	0	1	1
3	1	0	0	0
4	0	1	1	0

(c)

(a)



	a	b	c	d	e
a	∞	12	∞	∞	10
b	12	∞	9	11	8
c	∞	9	∞	14	∞
d	∞	11	14	∞	15
e	10	8	∞	15	∞

(b)

Gambar 6.a. Contoh representasi graf dalam matriks ketetanggaan

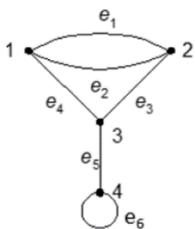
Gambar 6.b. Contoh representasi graf berbobot dalam matriks ketetanggaan

- Matriks bersisian (*incidency matrix*)

$$A = [a_{ij}]$$

a_{ij} bernilai 1 jika simpul i bersisian dengan sisi j .

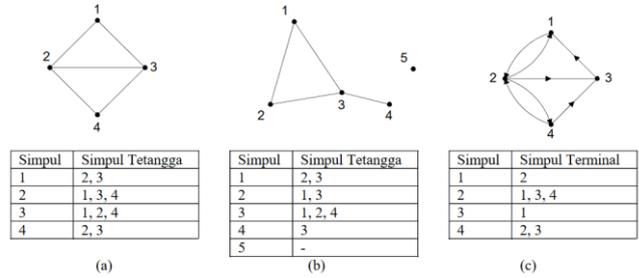
bernilai 0 jika tidak.



	e1	e2	e3	e4	e5	e6
1	1	1	0	1	0	0
2	1	1	1	0	0	0
3	0	0	1	1	1	0
4	0	0	0	0	1	1

Gambar 7. Contoh representasi graf dalam matriks bersisian

- Senarai ketetanggaan (*adjacency list*)



Gambar 8. Contoh representasi matriks dengan senarai ketetanggaan.

B. Kemiripan musik

Musik adalah suatu wujud seni, dimana medium utamanya adalah suara. Pada umumnya, musik didefinisikan dengan ritme, *pitch*, tempo, *timbre*, dan banyak cara-cara lainnya.

Pada graf yang akan dibuat, diperlukan metode pembobotan antar tiap simpul. Metode yang akan dipakai adalah pembobotan menggunakan *metadata* dari tiap lagu, yaitu data yang ada pada *file* music berisi judul, musisi, genre, dan lain-lain. Secara detail, pembobotan yang akan digunakan adalah sebagai berikut:

- Untuk tiap genre yang sama, bobot ditambah 1. Bobot ini untuk mengukur kemiripan lagu secara umum.
- Apabila selisih BPM (*beats per minute*) pada lagu lebih kecil dari atau sama dengan 10, bobot ditambah 1. Bobot ini untuk mengukur kemiripan ritme lagu.

Dengan pembobotan seperti diatas, dapat dibentuk suatu graf berbobot dari sekumpulan lagu.

C. Algoritma Dijkstra

Setelah didapatkan graf, diperlukan suatu algoritma untuk menentukan lintasan yang melalui bobot terbesar, yaitu kemiripan paling besar. Untuk itu, digunakan algoritma Dijkstra, yaitu algoritma untuk traversal graf dengan lintasan terkecil. Agar didapatkan lintasan dengan bobot terbesar, sebelum melakukan traversal dengan algoritma Dijkstra tiap bobot pada sisi akan diubah nilainya menjadi negatif. Berikut pseudocode untuk algoritma Dijkstra.

```
function dijkstra(G, S)
  for each vertex V in G
    distance[V] <- infinite
    previous[V] <- NULL
  If V != S, add V to Priority Queue Q
  distance[S] <- 0

  while Q IS NOT EMPTY
    U <- Extract MIN from Q
    for each unvisited neighbour V of U
      tempDistance <- distance[U] + edge_weight(U, V)
      if tempDistance < distance[V]
        distance[V] <- tempDistance
        previous[V] <- U
  return distance[], previous[]
```

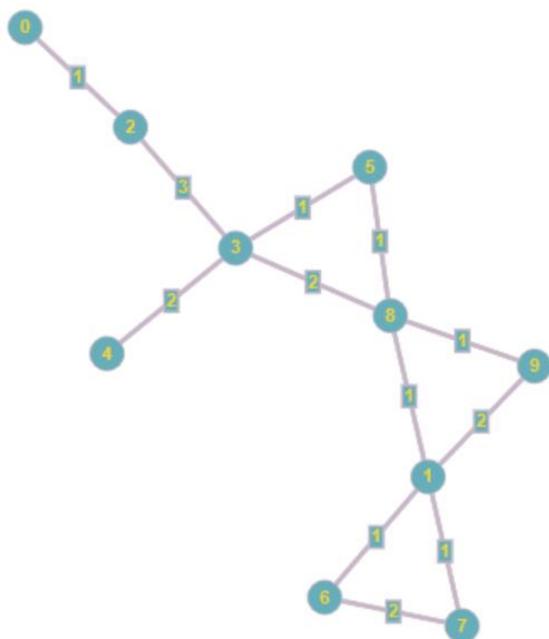
III. PERCOBAAN

Untuk membuat graf percobaan akan digunakan data sekumpulan lagu sebagai berikut:

No	Judul lagu - Musisi	Genre	BPM
0	Oddloop – Frederic	Rock, dance rock, experimental rock	172
1	Dream Fighter – Perfume	Electronic, Electropop	135
2	Looks – Steve Lacy	Rock, indie rock, indie, funk	94
3	Playground – Steve Lacy	Pop, funk, indie rock, R&B	96
4	Waves – Kanye West	Hip hop, pop	96
5	Best Part – Daniel Caesar	R&B	75
6	One More Time – Daft Punk	House, Electronic, Dance	123
7	Get Lucky – Daft Punk	Disco, Electronic	116
8	Starboy – The Weeknd	R&B, Electropop	93
9	Nee - Perfume	Electropop, Dance, J-Pop	128

Tabel 1. Data Contoh Lagu

Dari data diatas, dapat dibuat sebuah grafik berbobot sebagai berikut:



Gambar 9. Graf berbobot dari lagu-lagu pada Tabel 1, nomor simpul berkorespondensi dengan nomor pada tabel

Dari graf yang sudah dibuat, dapat dilihat hubungan antar lagu sesuai bobotnya. Tetapi, untuk membuat playlist dari data

tersebut diperlukan “titik awal” untuk playlist. Titik awal ini akan didapat dari input pengguna.

Dengan mendapat lagu dari input pengguna, lagu tersebut akan menjadi titik awal, kemudian untuk menemukan lagu selanjutnya akan digunakan algoritma Dijkstra, dengan bobot tiap sisi dinegasikan sehingga yang ditemukan adalah lintasan dengan bobot terbesar.

IV. KESIMPULAN

Teori graf yang merupakan bagian dari Matematika Diskrit memiliki banyak aplikasi. Salah satu aplikasi yang penulis coba lakukan adalah dalam pembuatan *playlist* musik, dengan tiap lagu direpresentasikan dalam bentuk simpul pada graf, dan dihubungkan oleh sisi berbobot yang menggambarkan kemiripan tiap lagu.

Setelah dilakukan percobaan, dapat dihasilkan sebuah metode pembuatan *playlist* musik menggunakan graf berbobot dan algoritma Dijkstra. Tentunya hasil yang didapat tidaklah sempurna, masih banyak yang bisa dikembangkan dari metode pembuatan *playlist* ini.

Salah satu saran dari penulis untuk pihak yang ingin mengembangkan metode pembuatan *playlist* ini adalah mengembangkan sistem pembobotan. Dengan sistem pembobotan yang lebih baik, misalnya menggunakan data genre yang disukai pendengar, tentu *playlist* yang dihasilkan akan lebih sesuai dengan selera pendengar.

V. UCAPAN TERIMA KASIH

Dengan selesainya makalah ini, penulis ingin memanjatkan puji syukur kepada Allah SWT, karena tanpa rahmat-Nya penulis tidak akan bisa menyelesaikan makalah ini.

Penulis juga ingin berterima kasih kepada pihak-pihak yang telah membantu penulis dalam rangka menyusun makalah ini, terutama Ibu Nur Ulfa Maulidevi selaku dosen pengampu dalam mata kuliah IF2120 Matematika Diskrit kelas K4, karena sudah memberi materi dan bimbingan selama satu semester kepada mahasiswa-mahasiswa di K4 Teknik Informatika ITB. Tanpa ilmu yang diajarkan Ibu Ulfa, tentu penulis tidak dapat menyusun makalah ini.

REFERENSI

- [1] <https://www.riaa.com/reports/2020-mid-year-music-industry-revenue-report-riaa/> diakses pada 10 Desember 2020.
- [2] <https://graphonline.ru/en/> diakses pada 10 Desember 2020.
- [3] <https://support.spotify.com/us/article/create-a-playlist/> diakses pada 10 Desember 2020.
- [4] R. Munir, “Diktat Kuliah Matematika Diskrit ITB (IF2120)”, 2020, diakses dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/matdis20-21.htm> pada 10 Desember 2020.
- [5] <https://www.programiz.com/dsa/dijkstra-algorithm> diakses 10 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

A handwritten signature in black ink, consisting of a large, stylized initial 'A' followed by a series of loops and a long horizontal stroke extending to the right.

Aditya Bimawan 13519064