

# Penerapan Graf Bipartite dalam *Music Recognition* Pada Aplikasi Shazam

Muhammad Syauqi Jannatan - 13521014<sup>1</sup>

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

<sup>1</sup>13521014@std.stei.itb.ac.id

**Abstrak**—Akhir-akhir ini, industri musik berkembang dengan pesat. Setidaknya satu lagu baru dirilis setiap harinya. Hal ini kadang membuat kita bingung untuk mengidentifikasi lagu yang terdengar asing bagi kita. Namun, seiring dengan perkembangan teknologi, masalah tersebut sudah dapat dipecahkan. Banyak dari kita cenderung menggunakan layanan aplikasi dengan fitur *music recognition* setiap kali kita menemukan lagu asing. Lagi pula, sangat mudah untuk mengeluarkan ponsel kita, membuka aplikasi, dan mengetahui segalanya tentang lagu tersebut dalam hitungan detik. *Shazam* merupakan salah satu aplikasi paling terkenal yang menggunakan sistem tersebut. Namun, bagaimana *Shazam* memberi kita semua informasi ini dengan begitu cepat?

Dalam makalah ini akan dibahas mengenai cara kerja *Shazam* dalam mengidentifikasi musik yang menggunakan salah satu prinsip dari graf, yaitu graf bipartite.

**Kata Kunci**—*Shazam*, sidik jari, spektogram, *hash*, graf bipartite, *anchor point*.



Logo Aplikasi Shazam

## I. PENDAHULUAN

Tak bisa dipungkiri, musik telah menjadi bagian dari kehidupan manusia. Sebagai produk kebudayaan, musik telah memainkan peran yang sangat penting dan tidak dapat dipisahkan dari masyarakat karena musik adalah representasi gagasan manusia sebagai individu maupun masyarakat. Ia adalah ungkapan rasa, ekspresi dan indikator eksistensi manusia.

Seringkali, ketika kita sedang bersantai atau mengerjakan sesuatu, tanpa sengaja kita mendengar sebuah lagu yang menurut kita bagus dan nyaman didengar, tetapi tidak dapat mengingat atau bahkan tidak mengetahui judul lagu tersebut. Mungkin, jika kita bisa mendengar lirik lagu tersebut, kita bisa mencarinya di internet. Namun, bagaimana jika lagu yang kita dengar tersebut tidak ada liriknya? Maka muncul lah salah satu aplikasi yang dapat membantu kita mengetahui informasi dari lagu yang kita dengar menggunakan *music recognition* bernama *Shazam*.

*Shazam* merupakan sebuah perusahaan yang didirikan pada tahun 2000 dan menyediakan sebuah jasa untuk menghubungkan orang dengan musik melalui *music recognition* lingkungan dengan bermodalkan *gadget* yang dimilikinya mulai tahun 2002.

*Shazam* adalah aplikasi yang dapat digunakan untuk mengenali lagu yang sedang diputar berdasarkan grafik frekuensi waktu atau yang biasa disebut spektogram. Kita dapat menjalankan aplikasi *Shazam* dan mengarahkan ponsel ke sumber suara lagu. *Shazam* menyimpan katalog sidik jari audio dalam database. *Shazam* bekerja dengan menganalisis suara yang ditangkap dan mencari rekaman yang relevan berdasarkan sidik jari akustik di *database* yang mereka miliki. Jika menemukan kecocokan, informasi seperti artis, judul lagu, dan album akan dikirim kembali ke pengguna. *Shazam* juga dapat digunakan untuk menemukan lirik lagu dan mengakses preview lagu di layanan streaming musik seperti Spotify. Salah satu alasan popularitas *Shazam* adalah kemampuannya untuk memberikan hasil identifikasi musik yang singkat. Waktu respons yang singkat ini dimungkinkan karena algoritma yang memanfaatkan sistem kerja yang disebut spektogram.

Waktu komputasi dan kebisingan eksternal adalah tantangan yang dihadapi oleh *Shazam* yang telah ditangani dengan sangat baik dan membuat aplikasi ini sukses besar. Dulu, *Shazam* hanya bisa menyamai lagu-lagu yang diputar di radio atau pemutar musik seperti stereo dan lain-lain. Dewasa ini, ada banyak versi untuk lagu yang sama disebut *remix* dan *coversongs*, yang juga dapat diidentifikasi melalui algoritma ini. *Shazam* dapat mengetahui versi mana yang telah diputar di telepon. Aplikasi *Shazam* juga memberi kita peta yang menunjukkan pencarian terbaru, deskripsi lagu terperinci, referensi ke layanan pihak ketiga. Setiap hari, sebuah lagu diperbarui di basis data *Shazam* dan dapat mengenali lagu dari hampir setiap bahasa dan sekarang pengembang telah berhasil meningkatkan aplikasi di setiap kriteria seperti antarmuka pengguna, implementasi pembelajaran mesin, dan lain-lain.

Disini, penulis tidak akan membahas lebih lanjut tentang algoritma yang dipakai perusahaan *Shazam* secara mendalam.

Jadi, penulis akan membahas tentang sistem kerja unik yang digunakan aplikasi *Shazam* pada pembacaan identitas musik yang memanfaatkan prinsip kerja dari graf.

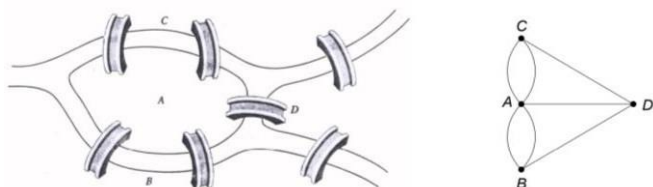
Spektrogram adalah dasar dari algoritme sidik jari audio *Shazam*. Kita dapat menganggapnya sebagai ringkasan digital dari sebuah lagu. Sama seperti sidik jari manusia, sidik jari akustik setiap lagu itu unik dan dapat dengan mudah diidentifikasi meskipun ada sedikit variasi data. Ini memungkinkan algoritma *Shazam* menghapus semua informasi yang tidak perlu tentang lagu tertentu. Hal tersebut dapat dicapai dengan beberapa langkah yang akan dibahas lebih lanjut dalam makalah ini.

## II. LANDASAN TEORI

### A. Graf dan Jenis-jenisnya

Graf adalah sebuah struktur yang digunakan untuk menggambarkan objek-objek diskrit dan hubungannya di antara objek-objek tersebut.

Secara detail, graf didefinisikan sebagai  $G = (V, E)$  dengan  $V$ , yaitu himpunan tak kosong dari simpul-simpul yang menggambarkan objek-objek diskrit, misal  $\{v_1, v_2, v_3, \dots, v_n\}$ , dan  $E$  yaitu himpunan dari sisi yang menghubungkan objek atau simpul, misal  $\{e_1, e_2, e_3, \dots, e_n\}$ . Sisi  $e$  yang menghubungkan simpul  $u$  dengan simpul  $v$  dinotasikan dengan:  $e = (u, v)$ .



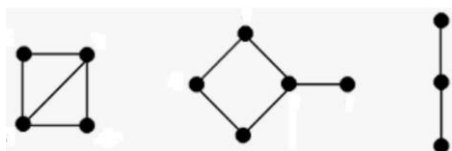
Jembatan Königsberg dan contoh graf, sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf>

Graf digolongkan dalam dua jenis berdasarkan ada atau tidaknya sisi ganda dan gelang, yaitu graf sederhana dan graf tak-sederhana.

#### 1. Graf sederhana

Graf yang tidak mengandung gelang maupun sisi ganda dinamakan graf sederhana.

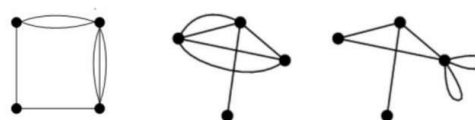


Graf Sederhana, sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf>

#### 2. Graf tak sederhana

Graf yang mengandung sisi ganda atau gelang dinamakan graf tak-sederhana (unsimple graph).



Graf Tak-Sederhana, sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf>

Graf juga dapat digolongkan berdasarkan orientasi arah pada sisi, yaitu graf tidak berarah dan graf berarah.

#### 1. Graf tak-berarah

Graf yang sisinya tidak mempunyai orientasi arah disebut graf tak-berarah.

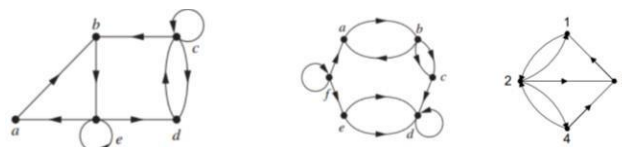


Graf Tak-Berarah, sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf>

#### 2. Graf berarah

Graf yang setiap sisinya diberikan orientasi arah disebut sebagai graf berarah.

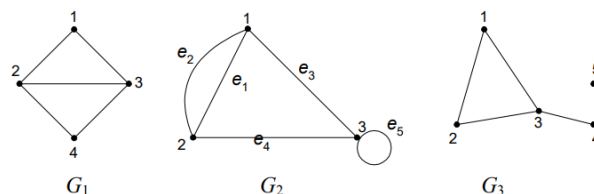


Graf Berarah, sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf>

### B. Terminologi Graf

Berikut merupakan beberapa terminologi penting yang berhubungan dengan graf.



Gambar 1 sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf>

#### 1. Ketetanggaan

Dua buah simpul dikatakan bertetangga jika terdapat suatu garis yang menghubungkan kedua simpul tersebut. Tinjau graf

G1 pada gambar 1, Simpul 1 dan simpul 2 dikatakan bertetangga karena keduanya terhubung langsung.

## 2. Bersisian

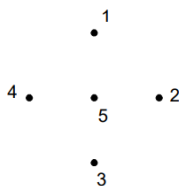
Tinjau graf G2 pada Gambar 1, untuk sembarang sisi  $e = (e1, e2)$ , dikatakan  $e$  bersisian dengan simpul 1 atau 2.

## 3. Simpul Terpencil

Simpul terpencil adalah suatu simpul yang tidak memiliki sisi yang bersisian dengannya. Tinjau graf G3 pada Gambar 1. Node 5 merupakan salah satu contoh simpul terpencil.

## 4. Graf Kosong

Graf Kosong adalah graf yang himpunan sisinya merupakan himpunan kosong. Semua simpul pada graf kosong adalah simpul terpencil. Tinjau Graf pada gambar dibawah, Graf tersebut dikatakan graf kosong karena tidak ada garis yang menghubungkan simpul-simpul pada graf.

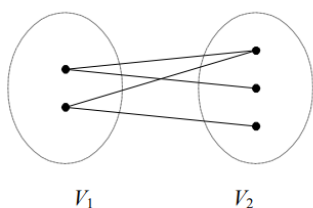


Graf Kosong, sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf>

## 5. Graf Bipartite

Graf Bipartite adalah suatu graf yang himpunan simpulnya dapat dipisah menjadi dua himpunan bagian  $V_1$  dan  $V_2$ , sedemikian sehingga setiap sisi pada graf menghubungkan sebuah simpul di  $V_1$  ke sebuah simpul di  $V_2$ .

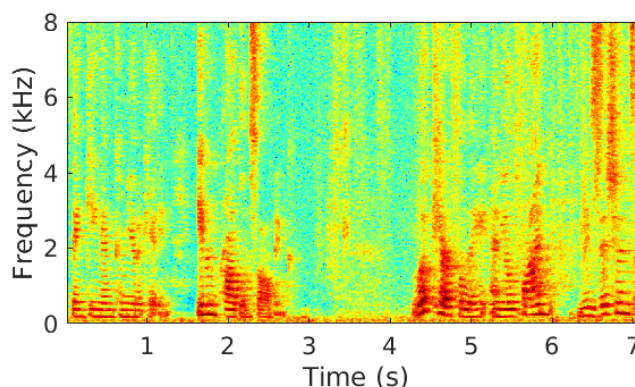


Graf Bipartite, sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf>

## C. Spektrogram dan Short Time Fourier Transform

Spektrogram adalah cara visual untuk merepresentasikan kekuatan sinyal, atau "kenyaringan" sinyal dari waktu ke waktu dari berbagai frekuensi yang ada dalam bentuk gelombang tertentu. Dengan kata lain, ini adalah grafik tiga dimensi. Jika kita melihat contoh di bawah ini, kita dapat melihat bahwa sumbu mewakili frekuensi dan waktu, sedangkan nilai ketiga (amplitudo) diwakili oleh intensitas warna setiap titik pada gambar. Untuk mengetahui lebih lanjut coba perhatikan gambar berikut.



Contoh Spektrogram, sumber:

[https://www.researchgate.net/figure/Example-of-spectrogram-for-a-signal-from-the-Dynamic-dataset-a-close-talk-clean\\_fig4\\_329841571](https://www.researchgate.net/figure/Example-of-spectrogram-for-a-signal-from-the-Dynamic-dataset-a-close-talk-clean_fig4_329841571)

Spektrogram sering digunakan untuk menganalisis suara dan musik serta menentukan karakteristik suara yang dihasilkan oleh suatu benda atau hewan, atau untuk mengidentifikasi pola dalam sinyal audio yang kompleks. Spektrogram dapat dibentuk melalui perhitungan signal waktu dengan menggunakan *Short-Time Fourier transform (STFT)*.

*Short-Time Fourier Transform (STFT)* adalah sebuah teknik yang digunakan untuk menganalisis suatu sinyal dalam domain frekuensi dengan memecah sinyal menjadi beberapa bagian yang lebih kecil (biasanya disebut sebagai "time-frames") dan melakukan transformasi Fourier pada masing-masing bagian tersebut. Ini berguna untuk menganalisis sinyal yang berubah-ubah atau memiliki struktur frekuensi yang kompleks dalam waktu yang singkat. STFT sering digunakan dalam bidang audio dan musik untuk menganalisis dan memodifikasi sinyal suara.

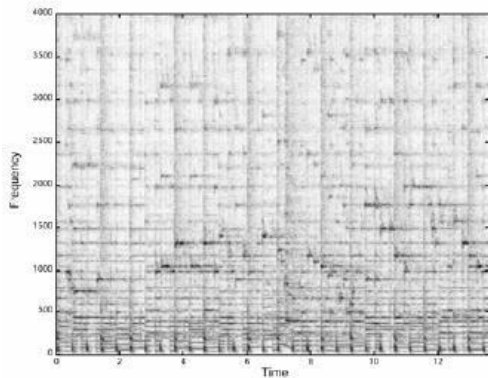
## III. METODOLOGI

### A. Sistem Kerja Music Recognition pada Aplikasi Shazam

Seperti yang sudah dibahas sebelumnya, *Shazam* mengidentifikasi suara dengan melakukan beberapa langkah. Langkah pertama adalah melakukan *sampling*, yaitu proses mengubah suara analog menjadi digital. Mikrofon mengubah suara disekitar menjadi sinyal diskrit, tetapi bentuknya tetap kontinu agar bisa diproses. Transformasi dari analog ke digital ini dilakukan menggunakan Transformasi Fourier. Transformasi dilakukan dalam batasan waktu tertentu, misal 0,1 *second*, karena batasan dalam mengubah dari kontinu menjadi diskrit. Setelah mendapatkan frekuensi dalam rentang waktu tertentu, maka terbentuklah spektrogram dengan contoh seperti gambar 3.1. Puncak spektrogram mewakili frekuensi terkuat dalam sinyal. Agar puncak frekuensi dapat digunakan dalam sidik jari audio, penting agar puncak frekuensi ditempatkan secara merata melalui spektrogram.

Sangatlah penting agar puncak-puncak itu memiliki jarak waktu yang sama sehingga sistem dapat mengenali bagian mana pun dari lagu tersebut. Misalnya, jika semua puncak ada di awal

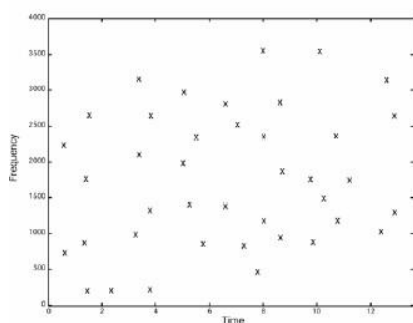
lagu, sidik jari tidak akan menutupi bagian selanjutnya.



Gambar 3.1 Grafik spektrogram<sup>[3]</sup>

Kita dapat mengakses informasi pengenalan dalam frekuensi yang ada dalam trek dengan menganalisis spektrogram, yaitu dengan mengubah semua informasi itu menjadi sidik jari unik untuk trek. Sidik jari audio ini bergantung pada penemuan puncak dalam spektrogram. Puncak ini adalah frekuensi paling keras pada suatu saat dalam lagu. Karena keras, kemungkinan besar mereka akan bertahan saat mengalami kebisingan atau distorsi lainnya.

Namun, mempertimbangkan *Shazam* memiliki katalog jutaan lagu, tidak mungkin untuk menyimpan seluruh spektrogram dan juga membandingkan seluruh spektrogram dari semua lagu di database dengan spektrogram dari lagu yang diunggah pengguna. Apalagi grafik spektrogram tersebut masih terlalu abstrak untuk dianalisis lebih lanjut, maka dibutuhkan grafik baru. Spektrogram rumit tersebut direduksi menjadi sekumpulan koordinat jarang yang disebut peta konstelasi seperti gambar berikut. Titik koordinat tersebut dipilih berdasarkan kriteria kepadatan untuk memastikan bahwa strip waktu frekuensi untuk file audio memiliki cukup cakupan yang seragam.



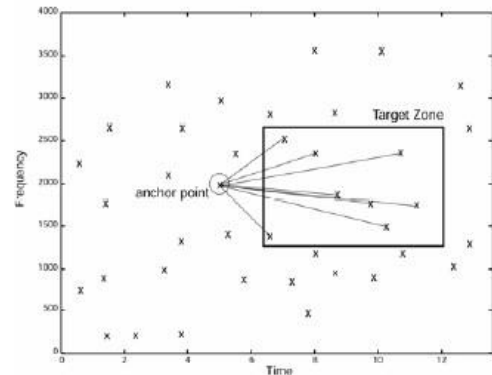
Gambar 3.2 Peta konstelasi<sup>[3]</sup>

Perhatikan bahwa bintang-bintang gelap pada gambar 3.1 (spektrogram) cocok dengan persilangan pada gambar 3.2.

Langkah awal dari sistem ini adalah memilih beberapa titik dari dalam spektrogram yang disederhanakan, disebut “anchor point”, lalu setiap *anchor point* tersebut akan berkaitan dengan titik-titik lain disekitarnya yang disebut “zona target”.

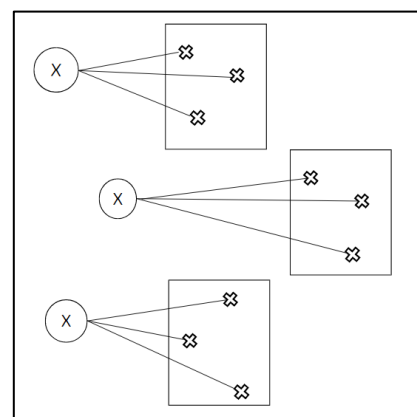
Langkah selanjutnya, setiap *anchor point* dipasangkan secara berurutan dengan titik-titik didalam zona targetnya dan akan menghasilkan dua komponen frekuensi. Gabungan dari setiap *anchor point* dan titik-titik lain pada zona targetnya akan membentuk suatu hash yang akan menjadi agregasi dari yang

berikut: frekuensi di mana titik jangkar berada ( $f_1$ ) + frekuensi di mana titik di zona target berada ( $f_2$ ) + perbedaan waktu antara waktu saat titik di zona target berada di lagu ( $t_2$ ) dan waktu saat titik jangkar berada di lagu ( $t_1$ ) +  $t_1$ . Untuk menyederhanakan:  $hash = (f_1+f_2+(t_2-t_1))+t_1$ . Kemudian, *hash* yang terbentuk akan disimpan dalam memori 32-Bit *unsigned integer*. Data yang disimpan inilah yang akan digunakan dalam pengecekan dengan database *Shazam*. Kombinasi *hash* dari setiap titik dalam spektrogram yang disederhanakan merupakan sidik jari dari sampel audio.

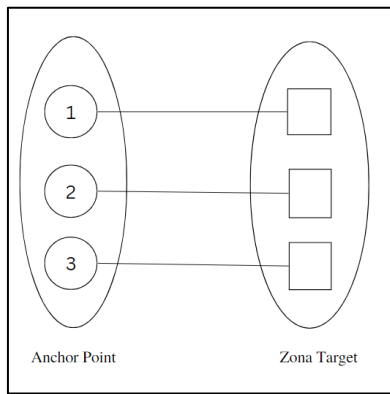


Gambar 3.3 Grafik kombinasi hash<sup>[3]</sup>

Jika kita analisis lebih lanjut menggunakan teori graf, sebuah *hash* yang terbentuk dari *anchor point* dan titik-titik lain tersebut menggunakan prinsip dari graf bipartite. Graf ini dirasa cocok untuk menyelesaikan masalah ini karena sifatnya yang seakan membagi dua kumpulan simpul. Jadi, tidak akan ada sisi yang menghubungkan simpul dari zona yang sama. Ini mencegah dari kemungkinan adanya data ganda yang disimpan.



Gambar 3.4 Representasi spektrogram 3 titik jangkar ke 3 zona target

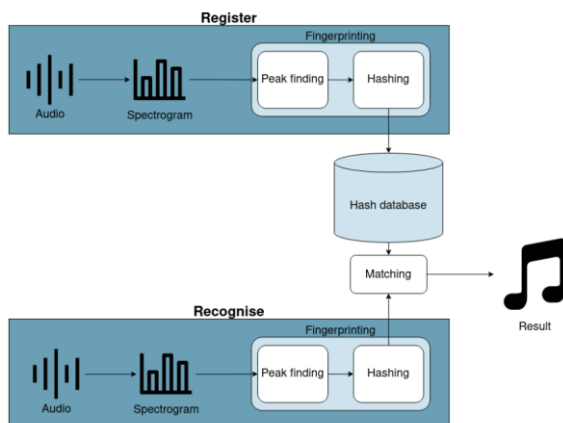


Gambar 3.5 Implementasi graf bipartite pada hash

Pada kasus ini, zona pertama dapat dianalogikan sebagai zona *anchor point* dan zona yang satunya merupakan zona target. Kemudian, sisi-sisi yang menghubungkan kedua zona tersebut merupakan representasi dari fungsi *hash* yang dibentuk. Implementasi graf bipartite pada kasus ini sangat penting agar penyimpanan data tertata dengan efisien sehingga sistem dapat menghemat memori serta waktu yang digunakan. Apabila tidak menggunakan graf ini, bisa jadi akan memakan waktu yang lebih lama.

Setiap *hash* akan berisi frekuensi *anchor point* dari zona target dan frekuensi titik yang *hash* nya telah dihitung, serta perbedaan waktu antara *anchor point* dan titik yang berada di dalam zona target. Pemrosesan terhadap setiap *hash* menjadi lebih mudah dilakukan berkat adanya pengelompokkan graf bipartite yang jelas, membedakan antara *anchor point* dan zona target. Hal ini menyebabkan lebih sedikit tabrakan *hash* yang terjadi dan mempercepat pencarian katalog beberapa kali lipat dengan memanfaatkan waktu pencarian.

Lantas, bagaimana cara mereka menemukan lagu berdasarkan sampel yang direkam? Ya, mereka mengulangi sidik jari yang sama juga ke sampel yang direkam. Setiap *hash* yang dihasilkan dari sampel suara, akan dicari kecocokannya di database. Jika kecocokan ditemukan, kita akan memiliki waktu *hash* dari sampel ( $th1$ ), waktu *hash* dari lagu di database ( $th2$ ) dan secara implisit identitas lagu yang cocok dengan *hash* tersebut. Pada dasarnya,  $th1$  adalah waktu sejak awal sampel hingga waktu sampel *hash* dan  $th2$  adalah waktu sejak awal lagu dan waktu *hash* lagu. Berikut merupakan gambar *overview* dari cara kerja Shazam.



Gambar 3.6 System Overview [7]

## B. Resonansi Graf Bipartite dalam music recognition pada Aplikasi Shazam

Dalam jurnal salah satu tim pengembang Aplikasi Shazam dijelaskan bahwa mereka mencocokkan lagu dengan *database* yang mereka miliki dengan menggunakan setiap *hash* di *database* yang cocok dengan *hash* di sidik jari yang baru saja diproses. Setiap *hash* disimpan dalam *database* yang terdiri dari 64 bit data, 32 Bit alokasi untuk *hash* dan 32 Bit sisanya alokasi untuk representasi *offset time* serta identitas mengenai informasi lagu yang dimaksud.

Dengan adanya penerapan graf bipartite, keakuratan pencocokan sampel *hash* ke *database* semakin tinggi karena tidak ada *hash* yang menumpuk berlebihan atau disimpan lebih dari satu kali saat mencocokkan ke *database* Shazam. Untuk memudahkan proses cepat, 64-Bit diurutkan berdasarkan nilai token *hash*. Dengan adanya implementasi graf bipartite, kecepatan pencarian informasi lagu ini secara otomatis meningkat karena setiap sampel data yang akan dicocokkan dikelompokkan terlebih dahulu membagi menjadi dua bagian. Pasti akan lebih mudah untuk mencari suatu informasi dari data yang tertata rapi dan teratur. Dan sebaliknya, pasti akan memakan waktu lebih lama apabila datanya tidak beraturan. Ini juga berlaku ketika setiap sampel *hash* dicocokkan dengan *hash* basis data. Berdasarkan jurnal salah satu tim pengembang dari Shazam, dibutuhkan sekitar 5-500 milidetik (tergantung pada pengaturan parameter dan aplikasi) untuk menemukan *hash* yang cocok dengan *hash* sampel musik yang kita dengarkan. Bayangkan apabila sampel musik tidak dikelompokkan dalam dua bagian terlebih dahulu, tentu saja proses pencocokan *hash* ini dapat memakan waktu lebih lama dari waktu sebelumnya.

Meskipun menemukan *hash* yang mirip dengan *hash* di *database* nya, aplikasi Shazam tersebut tetap melakukan analisis terhadap hasil pencocokan *hash* dalam bentuk histogram. Tentu saja, aplikasi akan mencari kemungkinan kesamaan *hash* tertinggi di antara sampel *hash* yang diterimanya. Analisis kemungkinan kesamaan *hash* ini penting untuk menilai beberapa lagu apabila tidak ada di basis data Shazam. Dalam metode yang dijelaskan oleh tim Shazam, mereka menggunakan 2 plot graf lagu untuk menemukan lagu yang tepat. Aplikasi Shazam ini dapat mencari lagu yang paling mirip berkat algoritma cerdas dalam analisis probabilitas kemiripan *hash*. Proses *hash* ini menjadi lebih mudah dan lebih cepat berkat konsep grafik bipartite. Tidak dapat disangkal bahwa graf bipartite pada sampel *hash* telah banyak membantu dalam meningkatkan kecepatan pemrosesan data *hash* yang ada.

## IV. KESIMPULAN

Berdasarkan pembahasan sebelumnya, dapat disimpulkan bahwa Shazam melakukan langkah berikut untuk mengidentifikasi lagu:

1. Menghitung spektrogram lagu
2. Mengekstrak dan memproses puncak dari spektrogram itu

3. Memasangkan puncak itu menjadi sebuah *hash*
4. Menyimpan koleksi *hash* untuk sebuah lagu sebagai sidik jari
5. Menghitung sidik jari sampel audio
6. Menemukan *hash* yang cocok dengan sidik jari itu di *database*
7. Menghitung waktu lacak untuk setiap *hash* yang cocok
8. Mengembalikan lagu dengan skor kecocokan tertinggi kepada pengguna

Dalam berbagai langkah diatas, prinsip graf bipartite berperan penting dalam meningkatkan akurasi pencocokan *hash* sampel dengan *database* dan berperan penting dalam keefektifan penyimpanan data, prinsip graf bipartite juga membantu meningkatkan kecepatan dan akurasi pembuatan plot graf untuk dianalisis lebih lanjut oleh aplikasi.

## V. UCAPAN TERIMA KASIH

Puji syukur penulis panjatkan kepada Allah SWT atas berkat, rahmat, dan hidayat-Nya, dapat terselesaikannya tugas makalah ini dengan baik. Pada kesempatan ini, penulis ingin mengucapkan terima kasih kepada Dr. Ir. Rinaldi Munir, M.T. selaku Dosen Mata Kuliah IF2120 Matematika Diskrit Kelas 03 atas bimbingan dan ilmu yang telah diberikan. Penulis juga ingin menyampaikan terima kasih kepada kedua orang tua atas dukungan yang selalu diberikan kepada penulis. Penulis juga mengucapkan terima kasih kepada teman-teman yang senantiasa memberikan bantuan dalam penyusunan makalah ini.

## REFERENSI

- [1] Munir, Rinaldi (2003). Graf (Bag.1) Bahan Kuliah IF2120 Matematika Diskrit, URL: <<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Graf-2020-Bagian1.pdf>> Diakses: 9 Desember 2022
- [2] Ana Haris. *How does Shazam work?*, URL: <<https://medium.com/@anaharris/how-does-shazam-work-d38f74e41359>> Diakses: 10 Desember 2022
- [3] Columbia. *Paper An Industrial-Strength Audio Search Algorithm* < <http://www.ee.columbia.edu/~dpwe/papers/Wang03-shazam.pdf> > Diakses: 9 Desember 2022
- [4] Christophe. How does Shazam work. URL:< <http://coding-geek.com/how-shazam-works/> > Diakses: 9 Desember 2022
- [5] Stanford. *Spectograms*. < <https://ccrma.stanford.edu/~jos/mdft/Spectograms.html> > Diakses: 9 Desember 2022
- [6] Jovanic, Jovan. < <https://www.toptal.com/algorithms/shazam-it-music-processing-fingerprinting-and-recognition>> Diakses: 9 Desember 2022
- [7] MacLeod, Cameron. *abracadabra: How does Shazam work?* < <https://www.toptal.com/algorithms/shazam-it-music-processing-fingerprinting-and-recognition>> Diakses: 10 Desember 2022

## 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



Muhhamad Syauqi Jannatan  
NIM : 13521014