Aplikasi Graf Bipartite dalam *Hash* Sistem *Music Recognition* (Aplikasi *Shazam*)

David Setyanugraha | 13511003[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]david.setya@students.itb.ac.id

***Abstrak*— Dewasa ini, banyaknya jumlah musik yang bermunculan telah menimbulkan suatu kesulitan bagi manusia untuk mengidentifikasi suatu judul musik ketika berada di suatu tempat yang memutarkan lagu yang asing baginya. Dengan suatu aplikasi bernama *"Shazam"*, kini masalah tersebut dapat diselesaikan. Setiap orang dapat langsung mengetahui judul lagu sampai nama penyanyi dari sebuah musik yang didengarnya tanpa harus pergi ke toko musik.**

Konsep utama dari aplikasi *Shazam* ini memanfaatkan identitas unik yang dimiliki oleh setiap lagu dan mencocokkannya dengan basis data yang dimilikinya. Identitas unik musik tersebut dikenal dengan nama *Spectogram*. Sistem penyimpanan identitas unik yang dimiliki oleh musik tersebut ternyata memanfaatkan prinsip dari salah satu jenis graf,yaitu graf bipartite.

Makalah ini akan membahas penerapan konsep graf bipartite dalam pencarian identitas music tersebut terutama dengan aplikasi "*Shazam*".

***Kata Kunci*—Aplikasi *Shazam*, *Spectogram*, Graf Bipartite**

I. PENDAHULUAN

Bayangkan jika anda sedang berada pada suatu tempat mendengarkan sebuah lagu yang bagus dan anda ingin mengetahui judul lagunya tetapi anda tidak mengetahui harus bertanya kepada siapa. Berakar dari masalah tersebut, maka dibuatlah suatu aplikasi pencarian identitas dalam musik (seperti judul lagu, nama penyanyi, dan lain-lain) dari sebuah lagu yang sedang diputar. Salah satu aplikasi yang terkenal yaitu aplikasi "*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.[5]

Rahasia keberhasilan Aplikasi *Shazam* ini terletak pada keberhasilan perusahaan *Shazam* dalam menciptakan suatu algoritma untuk membaca suatu audio musik unik dengan memanfaatkan system kerja *spectrogram*. Tentu saja algoritma tersebut tidak di buka untuk umum secara keseluruhan. Namun secara garis besar, system kerjanya telah dibahas di dalam sebuah *paper* oleh salah satu pengembang *Shazam*.[2] Kali ini kita tidak akan membahas secara mendalam algoritma yang dipakainya. Namun kita akan membahas system kerja unik pada pembacaan identitas musik (*spectrogram*) yang ternyata memanfaatkan prinsip kerja dari graf.

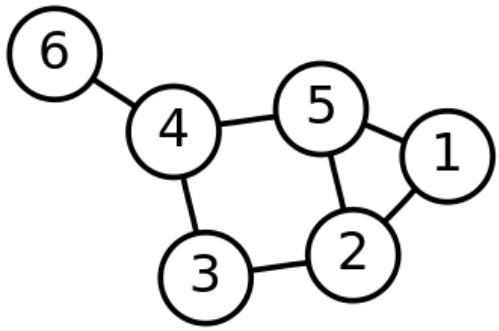
Spectogram adalah suatu representasi gelombang signal yang dimiliki oleh setiap suara yang keluar.[6] Dengan melakukan pencocokan antara spectogram tersebut dengan basis data *server* musik yang dimiliki oleh perusahaan *Shazam*, sebuah ponsel dapat mengetahui properti-properti seperti judul lagu dan penyanyi dari lagu yang sedang didengarnya. Perusahaan *Shazam* mengklaim bahwa perusahaannya telah memiliki jumlah pengguna sebanyak 250 juta orang dan akan terus menerus bertambah seiring bertambahnya waktu. **[5]**

Jadi, jika lebih diperinci lagi alur kerja dari aplikasi *Shazam* ini dibagi menjadi 4 bagian utama yaitu Pertama, Perusahaan *Shazam* telah menyimpan terlebih dahulu identitas unik dari tiap lagu yang dimilikinya dalam sebuah basis data yang dimilikinya. Kedua, ketika seorang pengguna ingin mencari tahu segala hal mengenai lagu yang sedang didengarnya, ia kemudian mengaktifkan aplikasi tersebut. Aplikasi tersebut akan mengambil sampel spectrogram selama 10 sekon dari audio yang ada di lingkungan. Ketiga, aplikasi tersebut kemudian akan meng*upload* identitas unik tersebut ke dalam suatu system yang dimiliki oleh perusahaan *Shazam*. Pada system tersebut, kemudian akan dijalankan suatu algoritma khusus dalam mencocokkan sampel musik yang ada dengan basis data yang dimiliki oleh *Shazam*. **[2]**

II. DASAR TEORI

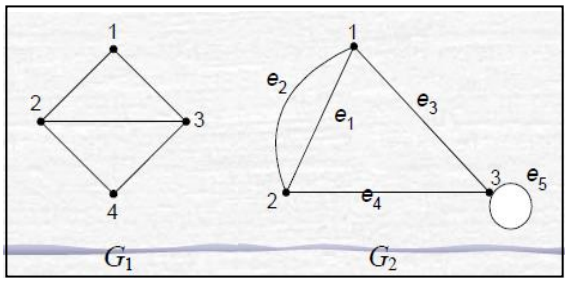
II.A Graf dan Jenis-jenisnya

Graf adalah himpunan benda-benda yang disebut simpul (*vertex* atau *node*) yang terhubung oleh sisi (edge). Graf memiliki banyak sekali aplikasi terutama dalam merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Representasi graf sederhana ditunjukkan pada gambar 1. Graf pada gambar 1 terdiri dari 6 node dan 7 buah sisi. **[3]**



Gambar 1. Contoh representasi Graf [3]

Untuk memahami prinsip kerja graf dalam algoritma *Shazam*, maka kita perlu memahami terlebih dahulu beberapa properti mengenai suatu graf. Berikut akan dijelaskan beberapa terminologi penting yang berhubungan dalam graf yaitu :



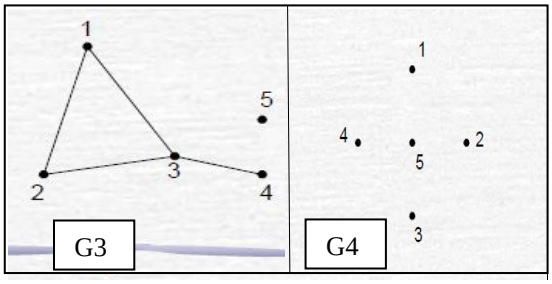
Ganbar 2 Representasi Ketetanggaan dan Bersisian[3]

1. Ketetanggaan (adjacent)
   Tinjau graf G1 pada Gambar 2. Dua buah simpul (vertex) dikatakan bertetangga jika ada suatu garis yang menghubungkan kedua simpul tersebut. Simpul 1 dan Simpul 2 dikatakan berketenggaan karena ada sisi yang menghubungkan kedua simpul tersebut. [3]

2. Bersisian
   TInjau graf G2.pada Gambar 2. Untuk sembarang sisi e= (e1,e2), dikatakan e bersisian dengan simpul 1 atau 2. [3]

3. Simpul terpencil
   Simpul terpencil adalah suatu simpul yang tidak memiliki sisi yang bersisian dengannya. TInjau graf G3 pada Gambar 3. Node 5 merupakan salah satu contoh simpul terpencil. [3]

4. Graf Kosong
   Graf kosong adalah graf yang himpunan simpulnya tidak ada yang bersisian. Tinjau graf G4 pada Gambar 3. Graf G4 dikatakan graf kosong karena tidak ada garis yang menghubungkan simpul-simpul pada graf, [3]

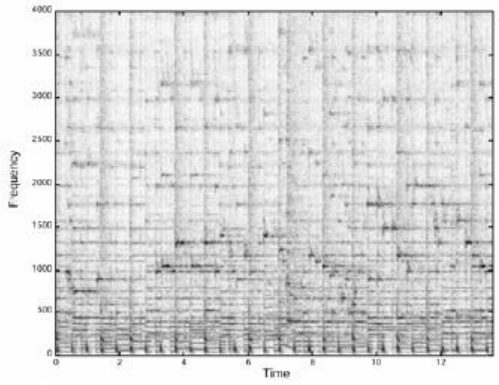


Gambar 3 Representasi Simpul Terpencil dan Graf Kosong [3]

5. Graf Bipartite
   Graf Bipartite adalah suatu graf yang himpunan simpulnya dapat dipisah menjadi dua himpunan bagian V1 dan V2, sedemikian sehingga setiap sisi pada graf menghubungkan sebuah simpul di V1 ke sebuah simpul di V2. Graf Bipartite ini dapat dilihat contohnya seperti pada gambar 7 dan 8. [3]

II.B *Spectogram* dan *Fourier Transform*

Spectogram adalah sebuah representasi sederhana yang digunakan dalam analisis densitas spectral signal terjadap waktu. Singkatnya, berbagai suara memiliki spectralnya masing-masing. Kita ambil salah satu contoh representasi grafik Spectogram pada gambar 4.



Gambar 4.Contoh Grafik Spectogram Frekuensi-waktu [2]

Spectogram telah banyak digunakan dalam bidang musik computer untuk membantu dalam pembuatan karya musik dalam industri musik. Spectogram dapat dibuat dengan 2 cara yaitu berdasarkan perkiraan *filterbank* (penyimpanan frekuensi) yang dihasilkan dari sejumlah *filter bandpass* (metode zaman sebelum ditemukan cara pemrosesan signal secara digital) atau melalui perhitungan signal waktu dengan menggunakan *Short-Time Fourier transform (STFT).* [1]

Metode STFT inilah yang sering digunakan pada industri musik sekarang. Perusahaan *Shazam* berhasil membuat suatu Algoritma dalam menangani cara menganalisis suatu identitas musik dari metode *STFT* ini.

Kita tidak akan membahas secara mendalam Algoritma *Fourier Transform* pada *paper* ini karena fokus utama paper ini adah menganalisis metode penanganan frekuensi

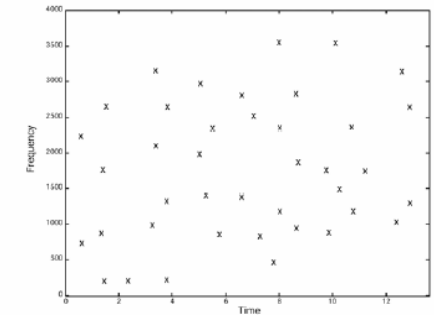
signal yang dihasilkan dari Fourier Transform ini memanfaatkan prinsip dari graf bipartite. Untuk pemahaman mendalam Fourier Transform ini, anda dapat merujuk pada buku-buku lain yang khusus membahas. Secara garis besar Fourier Transform ini digunakan mentranslasikan suatu musik ke dalam plot representasi spectogram Frekuensi-Waktu diatas untuk analisis lebih lanjut.

III. ANALISIS GRAF TERHADAP SISTEM KERJA *MUSIC DETECTOR* DALAM APLIKASI *SHAZAM*

IIIA. Plot Representasi grafik Spektogram yang dianalisis aplikasi *Shazam*

Seperti yang sudah dibahas sebelumnya pada bagian Dasar Teori, Aplikasi *Shazam* menganalisis identitas suatu musik dari grafik yang dihasilkan dari algoritma transformasi Fourier. Salah satu contoh representasi grafik spectrogram itu seperti yang ada pada gambar 4.

Akan tetapi, Analisa yang dilakukan Perusahaan *Shazam* itu tidak berhenti sekedar pada grafik gambar 4 karena gambar 4 masih terlalu abstrak. Perlu adanya suatu grafik baru yang lebih lanjut untuk dianalisis oleh Perusahaan *Shazam*. Grafik yang dimaksud dapat dilihat pada grafik gambar 5.

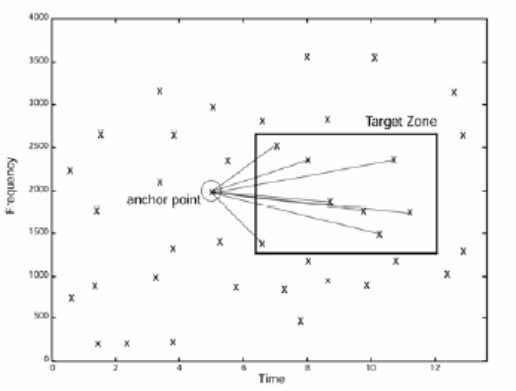


Gambar 5 Representasi titik-titik spectrogram [2]

Jika kita lihat dari grafik gambar 5, masih membutuhkan waktu yang lama untuk mencocokkan setiap frekuensi pada grafik tersebut. Maka kita perlu mengembangkan metode hanya mengambil beberapa sampel titik saja. Dari representasi titik-titik pada grafik 5 tersebut, perusahaan *Shazam* berhasil menemukan suatu cara dalam merepresentasikan sejenis identitas unik yang dimiliki oleh setiap musik dan ternyata hal tersebut berkaitan erat dengan konsep graf. Penjelasan lebih lanjut cara yang dimaksud dibahas dalam bagian IIIB.

IIIB. Hubungan graf dengan representasi grafik spectogram

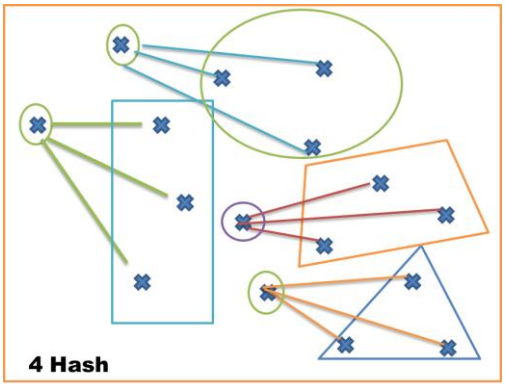
Seperti yang sudah dijelaskan sebelumnya, perusahaan *Shazam* memiliki cara dalam merepresentasikan setiap titik dari grafik spectrogram. Caranya ditunjukkan pada grafik 6.



Gambar 6 Grafik penyimpanan kode aplikasi *Shazam*[2]

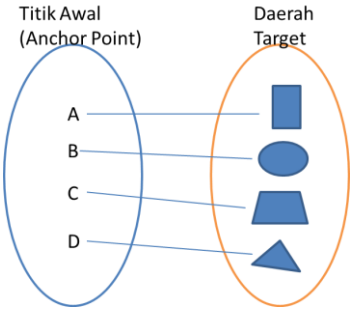
Cara kerja dari sistem pada gambar 6 adalah pada awalnya akan ditentukan suatu sampel titik awal *(anchor point)*. Titik awal ini kemudian akan menunjuk pada suatu daerah target yang juga terdiri dari titik lain. Sampel Titik awal ini tidak hanya satu, tetapi akan ada banyak titik awal yang menunjuk pada daerah yang berbeda. Gabungan satu titik awal dan titik-titik pada daerah targetnya kemudian akan membentuk suatu *hash*. Setiap *Hash* pada musik ini dapat diproduksi secara akurat karena sudah ada penanganan pada *noise* di sekitar lingkungan. Selanjutnya setiap Hash yang ada tersebut akan disimpan dalam memori 32-*Bit unsigned integer*. Data memori sebesar 32-Bit inilah yang digunakan dalam pengecekan dengan Basis Data *Shazam*. Tentu saja *Hash* yang dihasilkan tidak hanya satu tapi banyak sesuai dengan jumlah *hash* yang dibutuhkan.

Jika kita analisis dari sisi teori graf, maka kita dapat menganalogikan sebuah hash yang terdiri dari titik menuju titik lain sebagai sebuah implementasi dari sebuah graf bipartite. Ilustrasi yang lebih jelas dapat dilihat pada gambar 7 dan gambar 8.



Gambar 7 representasi spectrogram 4 titik awal ke 4 daerah

Terlihat pada gambar 7, terdapat 4 buah *Anchor Point* yang sedang menunjuk pada 4 *target area* yang berbeda. Sistem pengelompokkan pasangan-pasangan dengan konsep graf bipartite seperti inilah yang diterapkan oleh perusahaan *Shazam* dalam penanganan titik-titik pada grafik *spectrogram*.

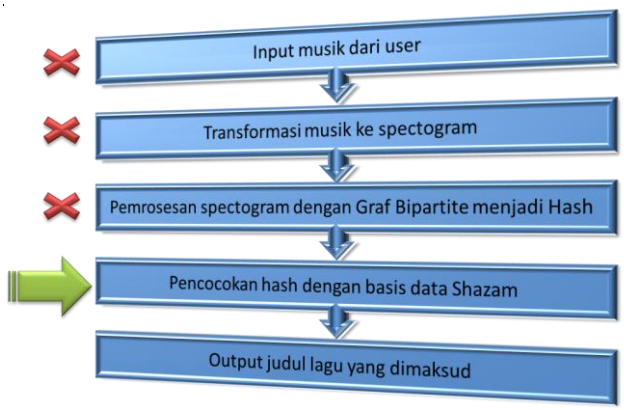


Gambar 8 Fungsi bipartite graf 4 *hash*

Implementasi Graf secara bipartite dalam system pembacaan frekuensi tetentu ini sangat penting manfaatnya dalam penyimpanan frekuensi musik. Bayangkan jika tidak ada graf bipartite ini, maka akan timbul kemungkinan dimana data yang disimpan ganda. Penyimpanan suatu data yang sama jelas tidak akan efisien karena ada data yang sama disimpan dua kali atau lebih sedangkan suatu aplikasi juga menuntut efisiensi penggunaan waktu yang ada. Jika tidak ada graf bipartite ini, mungkin akan butuh waktu lebih dari 10 sekon (waktu normal pemrosesan aplikasi *Shazam*) bagi suatu aplikasi *Shazam* untuk mendapatkan data jumlah frekuensi minimal yang ada.

Selain dari sisi keefisienan pemrosesan membaca, graf bipartite juga berperan dalam mempermudah pengaksesan tiap hash yang ada. Pada aplikasi *Shazam*, tiap hash akan berisikan frekuensi-frekuensi dari poin-poin yang ada serta perbedaan waktu antara titik dalam daerah target dan titik awal. Berkat pengelompokkan graf bipartite, maka pemrosesan terhadap setiap hash akan lebih mudah dilakukan. Misalnya kita ingin memroses titik awal *(Anchor point)* pada sebuah memori yang berisi banyak sekali hash dan poin-poin daerah targetnya. Di dalam graf bipartite, sudah ada pengelompokan jelas yang membedakan antara titik awal dan daerah target. Jelas, Hal ini mempermudah dalam pengaksesan tiap hash yang ada.

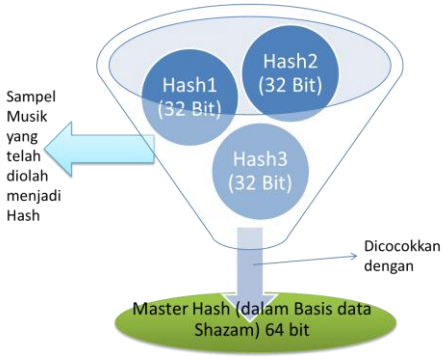
IIIC Pengaruh Graf Bipartite dalam performa pencarian dan pencocokan dengan basis data aplikasi *Shazam*



Gambar 9 alur berpikir

Pada bagian sebelumnya sudah disinggung bahwa graf bipartite telah membuat kecepatan pemrosesan pada aplikasi lebih cepat secara organisasi struktur data. Pada bagian ini kita akan menganalisis lebih *detail* lagi pengaruh konsep bipartite ini dalam algoritma pencarian dan pencocokan pada basis data aplikasi *Shazam* .

Dari *paper* yang dikeluarkan oleh salah satu pengembang aplikasi *Shazam*, ternyata system pencocokan utama lagu yang ada dengan basis data Shazam menggunakan hash yang telah diproses sebelumnya. Atau dengan kata lain proses pencocokan musik akan dilakukan dengan mencocokkan sampel hash yang ada dengan basis data dari perusahaan *Shazam*. Dalam *paper*nya dijelaskan bahwa setiap hash yang disimpan dalam basis data lagu *Shazam* terdiri dari 64-bit data, 32 Bit untuk hash dan 32 Bit untuk representasi *time offset* (digunakan saat pencocokan lagu) serta identitas lagu yang dimaksud. Untuk lebih jelas ilustrasi system kerjanya dapat dilihat pada gambar 10.



Gambar 10 Diagram alir pencocokan *hash*

Dari hasil analisis yang saya lakukan, berkat adanya graf bipartit, akurasi pencocokan sampel hash dengan basis data akan lebih tinggi sebab tidak akan terjadi penumpukan hash yang berlebihan pada saat pencocokan dengan basis data *Shazam* (setiap hash telah dipaket dalam sebuah pasangan grup). Hal ini turut menyumbang pada tingkat keakuratan dalam pengecekan suatu lagu dengan basis data dari *Shazam*.

Kecepatan penemuan judul lagu yang dimaksud juga otomatis meningkat berkat adanya implementasi graf bipartite ini. Sebab setiap data yang akan dicocokkan sampelnya sudah dkelompokkan terlebih dahulu. Kita dapat menganalogikannya seperti kita mengatur sebuah barisan yang sudah teratur dan belum teratur untuk melakukan sesuatu. Jelas akan lebih susah dan membutuhkan waktu yang lama bagi barisan yang belum teratur untuk melakukan hal tersebut. Hal ini juga berlaku sama pada saat pencocokan masing-masing sampel hash dengan hash yang ada di basis data. Dari data yang kami peroleh dari *paper* system kerja aplikasi *Shazam*, dibutuhkan waktu sekitar 5-500 milisekon untuk menemukan hash yang cocok dengan hash dari sampel musik yang sedang didengar. Jika hash pada sampel musik belum di kelompokkan dalam sebuah graf bipartite,

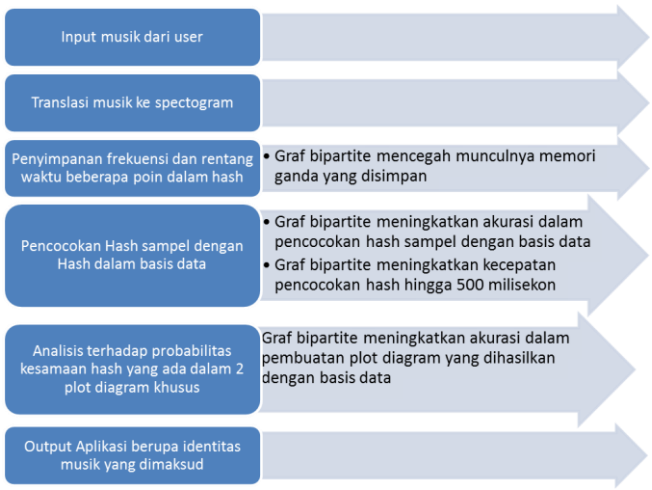
mungkin akan dibutuhkan waktu yang jauh lebih besar dari waktu maksimal 500 milisekon untuk menemukan hash yang dimaksud. Waktu ini belum lagi ditambah dengan analisis terhadap kesamaan hash yang ditemukan dengan hash pada basis data.

Sebagai informasi, pada aplikasi *shazam* walaupun sudah ditemukan *hash* yang mirip dengan hash yang ada di basis datanya, aplikasi tersebut tetap melakukan suatu analisis pada hasil proses pencocokan *hash* yang sudah diperoleh dalam bentuk plot graf. Tentu saja, aplikasi akan mencari probabilitas kemiripan *hash* yang paling besar dari sampel hash yang diperolehnya. Analisa terhadap probabilitas kemiripan *hash* ini penting untuk menanggulangi beberapa lagu yang tidak ada dalam basis data *shazam*. Dari metode yang dijelaskan perusahaan *shazam*, mereka menggunakan plot 2 diagram untuk mencari lagu yang tepat. Aplikasi *Shazam* ini dapat mencari lagu yang paling mirip berkat adanya algortima pintar dalam analisis probabilitas kesamaan Hash ini. Sekali lagi pengolahan *hash* ini menjadi lebih mudah dan cepat dilakukan berkat adanya konsep graf bipartite.

Tidak dapat dipungkiri bahwa Graf bipartite pada sampel *hash* telah membuat sebuah terobosan baru dalam peningkatan kecepatan penanganan data *hash* yang ada. Jadi, dibalik rahasia Perusahaan *Shazam* dapat menghasilkan suatu aplikasi yang kecepatan pemrosesan datanya cepat, perusahaan ini telah menerapkan salah satu prinsip graf yaitu graf bipartite.

IV. DIAGRAM ALIR PENGARUH GRAF BIPARTITE PADA PERFORMA APLIKASI SHAZAM

Berikut adalah diagram alir rangkuman pengaruh secara keseluruhan graf bipartite pada proses kerja aplikasi shazam :



Gambar 11. Diagram Alir pengaruh Graf Bipartite

V. KESIMPULAN

1. Konsep Graf Bipartite berperan penting dalam meningkatkan kemangkusan penyimpanan data sampel *hash* musik agar tidak ada memori yang ganda disimpan.
2. Konsep Graf Bipartite membantu meningkatkan akurasi dalam pencocokan hash sampel dengan basis data karena tidak adanya sampel hash yang bertumpukan.
3. Konsep Graf Bipartite membantu meningkatkan kecepatan pencocokan hash sampel dengan basis data karena sampel hash yang sudah teratur berkat graf bipartite.
4. Konsep Graf Bipartite berperan penting dalam meningkatkan akurasi pembuatan plot diagram untuk dianalisis oleh aplikasi

DAFTAR PUSTAKA

[1] Bracewell, R. N. (2000), *The Fourier Transform and Its Applications* (3rd ed.), Boston: McGraw-Hill. < http://books.google.com/books/about/The_Fourier_transform_and_its_applicatio.html?id=ZNQQAQAAIAAJ >

[2] Columbia. *Paper An Industrial-Strength Audio Search Algorithm* .15 Desember 2012 (19.00) < http://www.ee.columbia.edu/~dpwe/papers/Wang03-shazam.pdf >

[3] Munir,Rinaldi. 2008. *Diktat Kuliah IF 2091 Struktur Diskrit edisi 4*. Program Studi Teknik Informatika STEI ITB.

[4] S. Haykin, editor, *Advances in Spectrum Analysis and Array Processing*, volume.1, Prentice-Hall, 1991. < http://books.google.com/books/about/Advances_in_spectrum_analysis_and_array.html?id=V-1SAAAAMAAJ >

[5] Shazam. *Who is Shazam?*. 15 Desember 2012 (18.00) http://www.shazam.com/music/web/about.html >

[6] Stanford. *Spectograms*. 16 Desember 2012 (8.00) < https://ccrma.stanford.edu/~jos/mdft/Spectrograms.html >

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, 15 Desember 2012

David Setyanugraha | 13511003