

# Music Information Retrieval Menggunakan $k$ -NN dan Cosine Similarity

Salam Kurniawan<sup>1</sup>, Surya Agustian<sup>2</sup>

Teknik Informatika UIN Sultan Syarif Kasim Riau  
Jl. H.R. Soeberantas km. 11,5 Simpang Baru, Panam, Pekanbaru  
e-mail: salamkurniawan@gmail.com<sup>1</sup>, sagustian@gmail.com<sup>2</sup>

## Abstrak

Berbagai macam jenis, genre dan ukuran musik/lagu dapat ditemukan di jaringan internet pada saat ini, untuk dimainkan dengan komputer dan gadget. Namun, untuk mengetahui informasi tentang musik seperti judul lagu dan pencipta dari potongan musik yang dimainkan, menjadi masalah tersendiri apabila kita tidak mengetahuinya. Penelitian ini mengusulkan sebuah sistem pencarian informasi musik (music information retrieval) yang dapat mengenali fiturnya dari input kueri berupa potongan musik yang singkat. Fitur musik diekstraksi menggunakan Echonest API. Proses retrieval berdasarkan similarity fitur menggunakan metode kNNCS (K-Nearest Neighbour dan Cosine Similarity). Eksperimen dilakukan dengan pengaturan kombinasi beberapa parameter untuk mencari model terbaik. Hasilnya diterapkan pada data pengujian berupa potongan musik, menghasilkan performa berdasarkan akurasi, precision, dan recall dengan nilai masing-masing 0.6778, 0.9661, dan 0.6785. Dari segi akurasi, sistem ini telah menghasilkan nilai yang sangat baik, Namun dari segi recall, masih harus ditingkatkan agar kandidat musik yang tepat dapat berada pada peringkat teratas.

**Kata kunci:**  $k$ -NN, cosine similarity, sistem pencari musik, fitur musik, echonest

## Abstract

Various types, genres and sizes of music or songs can easily be found on the internet today, to be played with computers and gadgets. However, a problem raised in finding about information of the music such as the song's title and the composer from a piece of played music, if we don't know it. This study proposes a music information retrieval that can recognize music features from the input query in the form of short pieces of music. Music features are extracted using the Echonest API. The retrieval process is based on a feature similarity using the kNNCS (k-Nearest Neighbor and Cosine Similarity) method. Experiments were conducted to seek the best model with combination of k-NN parameters. Then, this model is applied to the test query data in the form of short music pieces, yielding performance in terms of accuracy, precision, and recall with values of 0.6778, 0.9661, and 0.6785, respectively. Regarding the precision, this system has produced very good scores. However, in terms of recall and accuracy, it still has to be improved so that the right music candidate can be found in the top ranking.

**Keywords:**  $k$ -NN, cosine similarity, music information retrieval, music feature, echonest

## 1. Pendahuluan

Saat ini, beragam data musik dengan jumlah yang sangat besar tersimpan di internet, mulai dari musik-musik terbaru sampai klasik yang telah direkonstruksi menjadi format digital. Setiap orang dapat dengan mudah mencari musik atau lagu yang disukainya berdasarkan judul maupun lirik di dalam lagu tersebut, melalui sebuah sistem *music retrieval*.

Namun ketika seseorang hanya memiliki potongan musik berformat \*.mp3 yang disukai tetapi tidak mengetahui informasi dari musik tersebut, misalnya apa judulnya dan siapa artisnya, tentunya ini akan menjadi sebuah permasalahan tersendiri. Mungkin seseorang ingin mencari lirik dari lagu tersebut agar dapat ikut menyanyikannya. Tanpa mengetahui judul lagu dan artisnya, maka akan sulit menemukan lirik yang dimaksud. Pada saat suatu sistem (misalnya Youtube) ingin mendeteksi apakah suatu potongan musik dalam suatu video disebar dengan melanggar hak cipta yang dimiliki seseorang, maka proses mencari sumber aslinya sangat diperlukan. Bagaimana menemukan musik yang sama atau mirip dengan potongan musik yang dimiliki untuk mengetahui informasi musik tersebut merupakan masalah yang ingin dipecahkan dalam penelitian *music information retrieval (MIR)* ini.

Beberapa hasil penelitian di bidang *MIR* dilaporkan dengan berbagai fokus dan tujuan penelitian. Antara lain identifikasi musik untuk mengklasifikasi *genre* menggunakan fungsi *similarity* [1], memberikan hasil yang efektif. Penelitian lain untuk klasifikasi *genre* musik dengan *support vector machine* menghasilkan akurasi sebesar 68,75 % dengan 112 data pengujian [2]. *Mood, genre* dan tempo digunakan sebagai fitur untuk *music information retrieval* pada aplikasi *MusicMoo* [3]. Fitur *NLP* pada *Jukebox*, di mana sebuah model *deep learning* membangkitkan kata-kata dari suara audio di dalam sebuah lagu yang di-*encode* secara kontinu, dimanfaatkan di dalam [4] untuk digunakan dalam aplikasi *MIR*. Sedangkan [5] memfokuskan pada pengenalan fitur tempo dan *chord* untuk membedakan *genre* musik *metal* dan *rock* pada aplikasi *MIR*.

Tipe data yang berbeda dari suatu file musik atau audio, dibandingkan dengan file dokumen teks atau citra, menyebabkan fitur yang dapat digunakan untuk representasi musik menjadi beragam, yang dapat berasal dari intrinsik musik seperti *loudness*, tempo, *chord*, *ensemble* frekuensi, maupun informasi ekstrinsiknya seperti meta data, dan sebagainya.

*Echonest* adalah sebuah perusahaan musik yang menyediakan *web service* untuk musik. Banyak aplikasi berbasis musik dibangun menggunakan platform *Echonest* [6]. *Echonest* menyediakan kunci *API* yang bisa digunakan untuk mendapatkan data-data musik dari server *Echonest* secara gratis. Data-data tersebut dapat dipergunakan sebagai fitur yang merepresentasikan musik digital dalam bentuk vektor data. Di samping itu, server *Echonest* menyimpan ribuan data musik sehingga bisa dijadikan korpus atau koleksi dokumen untuk sistem *MIR*.

Penelitian ini bertujuan untuk mengembangkan aplikasi pencarian informasi musik dari input potongan pendek musik atau lagu dengan memanfaatkan fitur *Echonest*. Penulis memanfaatkan *API (Application Programming Interface)* dalam pencarian musik dan data pada *Echonest* sebagai *knowledgebase*-nya. Proses *retrieval* berdasarkan *ranking* hasil perhitungan kemiripan antara fitur musik pada koleksi *Echonest* dan fitur pada input kueri, berdasarkan *k-NN* dan *Cosine Similarity*. Alasan *k-NN* dan *cosine similarity* digunakan karena implementasi yang mudah dan cepat dalam menghitung jarak dan kemiripan dari dua vektor berorde banyak. Untuk mendapatkan kandidat dokumen dari suatu kueri, suatu sistem *information retrieval* memerlukan komputasi *similarity* yang banyak, yaitu menghitung kemiripan antara kueri dengan setiap dokumen di dalam koleksi (korpus). Untuk korpus yang berukuran sangat besar, maka mempersingkat proses tersebut dengan membatasi jumlah kandidat yang dihitung jaraknya dengan vektor kueri akan mengurangi proses komputasi kemiripan oleh *cosine similarity*.

Bagian selanjutnya membahas mengenai metode penelitian *music information retrieval* ini. Bagian ketiga menerangkan eksperimen dan analisa hasil eksperimen, ditutup dengan bagian kesimpulan, yang berisi kesimpulan hasil penelitian serta saran untuk penelitian selanjutnya, yang dapat dikembangkan dari penelitian ini.

## 2. Metode Penelitian

*Music Information Retrieval (MIR)* adalah fasilitas pencarian informasi yang mengenai musik, yang merupakan bit-bit di dalam sistem penyimpanan digital yang sangat besar, yang menyimpan milyaran data tanpa struktur yang jelas [7]. Oleh karena itu, untuk berbagai keperluan analisis dan pengolahan di bidang *music retrieval* maupun *music information retrieval*, maka data musik perlu direpresentasikan dalam format yang lebih pendek dan struktur yang sederhana berdasarkan fitur-fitur yang dimilikinya.

### 2.1. Fitur Data Musik dari *Echonest*

Sebagai dokumen di dalam koleksi (korpus), musik perlu diwakili oleh suatu representasi yang dapat dikalkulasi. Bentuk berkas (file) data musik yang berbeda dari teks menyebabkan teknik untuk membangkitkan fitur representasi juga harus disesuaikan, terutama karena data musik tersusun atas ensemble berbagai frekuensi bunyi dan suara, yang berubah menurut waktu. Frekuensi menjadi salah satu fitur yang sangat populer dalam pengolahan data audio. Elemen lainnya selain frekuensi untuk dijadikan fitur adalah energi, kenyaringan (*loudness*), *key feature*, *mode* dan *tempo* [8, 9].

Dari input kueri berupa file musik (atau potongan pendek) berekstensi *mp3*, *API Echonest* akan membangkitkan data sebagai berikut [6], yang sebagiannya akan dipilih sebagai vektor representasi dari suatu musik dalam penelitian ini:

- a. *Meta data* : informasi dari data musik.
- b. *Track data* : berisikan;
  1. *Time signature*: estimasi dari *signature* waktu secara keseluruhan dari suatu *track*.
  2. *Key*: estimasi dari kunci keseluruhan dari *track*.
  3. *Mode*: mengindikasikan *modality* (*major* atau *minor*) dari suatu *track*, yaitu tipe skala di mana isi melodi dari musik diperoleh.
  4. *Tempo*: estimasi tempo keseluruhan dari suatu *track* dalam satuan *bpm* (*beats per minutes*).
  5. *Loudness*: informasi kenyaringan secara keseluruhan dari suatu *track* dalam *decibels* (*dB*).
  6. *Duration*: lamanya suatu *track* dalam detik dihitung secara tepat oleh suatu decoder audio.
  7. *End of fade in*: waktu akhir dari *intro fade-in* suatu *track* musik dalam satuan detik.
  8. *Start of fade out*: waktu mulai dari *fade out* pada akhir suatu *track* musik, dalam satuan detik.
  9. *Codestring, Echoprintstring*: representasi dari dua *fingerprint audio* yang dihitung pada *track audio* dan digunakan oleh layanan *Echonest* lainnya untuk mendeteksi lagu.
- c. *Timbre, pitch, dan loudness*:
  1. *loudness\_start*: mengindikasikan level kenyaringan pada permulaan segmen musik
  2. *loudness\_max\_time*: *offset* dari *segment* yang mengandung titik maksimum kenyaringan
  3. *loudness\_max*: puncak nilai kenyaringan di dalam segmen
- d. *Sequenced data*: data audio yang dipecah ke dalam elemen-elemen dengan interval dan urutan waktu tertentu.

## 2.2. *k-Nearest Neighbor (k-NN)*

*K-Nearest Neighbor* adalah suatu metode pembelajaran mesin *supervised* untuk *task* klasifikasi, dimana hasil dari kueri input yang baru, diklasifikasi berdasarkan mayoritas dari kelas pada sejumlah *k* data yang terdekat dengan kueri [10]. Algoritma *k-NN* bekerja berdasarkan jarak terpendek dari kueri input ke sampel data tetangga.

Pada proses *training*, data musik diproyeksikan ke ruang berdimensi *n* dalam bentuk vektor fitur *Echonest*. Pada fase klasifikasi, fitur yang sama untuk data *testing* ditransformasikan ke dalam vektor fitur. Jarak (*Euclidean distance*) dari vektor input *x* terhadap seluruh vektor data *training* *y* dihitung menurut persamaan (1), dan sejumlah *k* *sample* yang jaraknya paling dekat dipilih sebagai kandidat.

$$Distance(x, y) = \sqrt{\sum_{i=0}^N (x_i - y_i)^2} \quad (1)$$

dengan *i* adalah nomor indeks dari elemen penyusun vektor fitur musik dari *Echonest*.

## 2.3. *Cosine Similarity*

*Cosine Similarity* merupakan salah satu metode untuk mengukur kemiripan antara dua vektor berdasarkan sudut yang dibentuknya [11]. Perhitungan pada *cosine similarity* didasarkan pada *dot product* di antara dua vektor (vektor data kueri *q* dan data dokumen *d*), berdasarkan persamaan (2). Kandidat dokumen musik (terpilih oleh proses *k-NN*) diurutkan berdasarkan nilai *cosine* yang paling besar (nilai *cosine* maksimum adalah 1). Rasionalnya adalah bila nilai *cosine* adalah 1, sama artinya dengan kedua vektor adalah searah atau berhimpit (sudut antar vektor adalah 0), sehingga dokumen yang ditemukan, sama dengan kueri input.

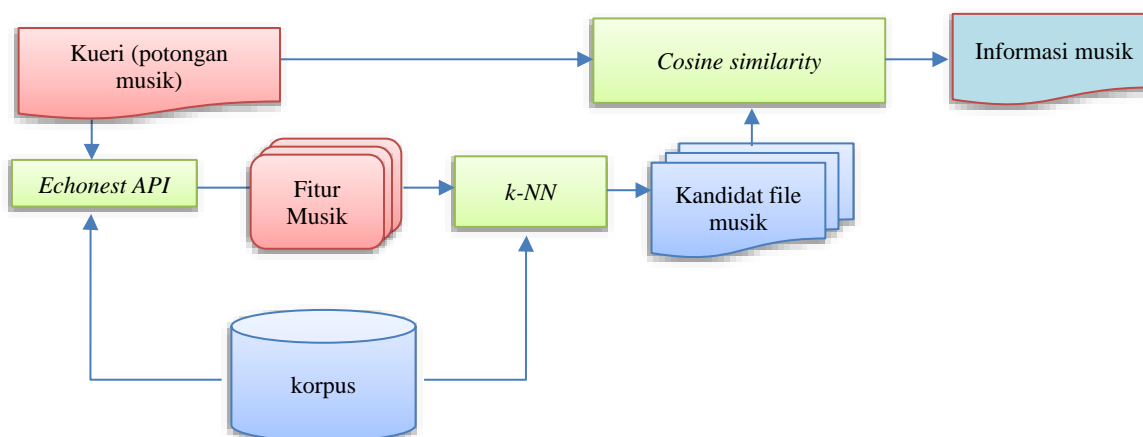
$$cosine(\vec{d}, \vec{q}) = \frac{\vec{d} \cdot \vec{q}}{|\vec{d}| \cdot |\vec{q}|} = \frac{\sum_i^N (w_{id} \cdot w_{iq})}{\sqrt{\sum_i^N w_{id}^2} \cdot \sqrt{\sum_i^N w_{iq}^2}} \quad (2)$$

dengan *w* adalah bobot dari elemen vektor ke-*i*, dalam hal ini adalah nilai dari fitur yang dibangkitkan oleh *Echonest API*.

## 2.4. Metode *Music Information Retrieval* yang Diusulkan

Aplikasi pencarian informasi tentang musik (*MIR*) menerima input kueri berupa potongan pendek musik, yang diinputkan ke dalam system yang dikembangkan berbasis web. Langkah-langkah *retrieval*-nya sesuai Gambar 1 adalah sebagai berikut:

1. Korpus dibangun dengan membentuk vektor fitur dari koleksi musik pada server *Echonest*. Fitur-fitur yang dipilih untuk vektor representasi dibangkitkan dari *Echonest API*, yang dipilih 5 fitur utama, yaitu *key*, *mode*, *loudness*, *energy*, dan *tempo*.
2. Kueri (dapat berupa potongan atau musik lengkap) diubah ke dalam bentuk vektor fitur dengan cara yang sama, menggunakan *Echonest API*.
3. Tahapan *retrieval* untuk mencari informasi musik pada kueri, membandingkan jarak *Euclidean* dan *cosine similarity* antara vektor kueri dan vektor data musik pada korpus.
  - a. Jarak *Euclidean* dihitung untuk menemukan *k* kandidat terdekat dengan input kueri.
  - b. *Cosine similarity* dihitung untuk menemukan 1 data musik yang paling relevan dari *k* kandidat hasil dari *k-NN*.
4. Output dari sistem adalah informasi lengkap tentang musik atau lagu dari kueri potongan musik, yang ditemukan pada korpus.

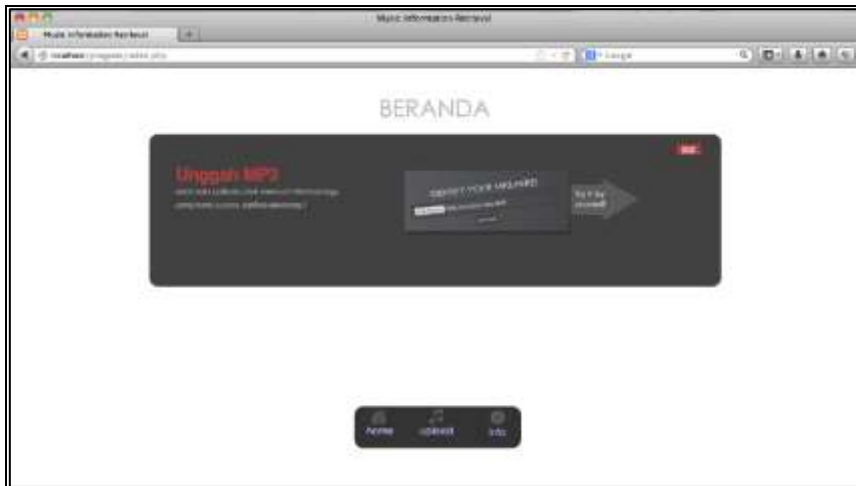


Gambar 1. Skema Tahapan *Music Information Retrieval*

## 3. Hasil dan Pembahasan

### 3.1. Antar Muka Sistem

Pada halaman beranda, sistem *MIR* ini memberikan informasi singkat tentang sistem (Gambar 2a). Pada saat link “*Try it by yourself*” diklik, akan masuk ke menu “halaman *upload*”, yaitu pencarian menggunakan track atau potongan musik (Gambar 2b). Hasil pencarian ditampilkan pada halaman info (Gambar 2c). User dapat memasukkan input berupa potongan *track* musik maupun file musik lengkap.



(a)



(b)



(c)

Gambar 2. Halaman antar muka sistem

### 3.2. Skema Pengujian

Dari data musik yang tersimpan di *Echonest*, dibentuk korpus dengan cara *download* sebanyak 280 lagu atau musik lengkap dengan informasi yang tersedia. Informasi ini dapat dibangkitkan melalui *Echonest API*. Semua lagu yang *download* memiliki *bit rate* 64 kbit/s.

Dari seluruh lagu di dalam korpus tersebut, dipilih secara acak 30 lagu atau musik, dan kemudian *bit rate*-nya dikonversi menjadi 32 bit/s. Hal ini untuk mensimulasikan bahwa dalam dunia nyata, user tidak mengetahui atau mempertimbangkan masalah *bit rate* dari potongan musik yang diupload ke sistem sebagai kueri. Sehingga ada kemungkinan *bit rate*-nya berbeda dari yang tersedia di dalam korpus.

Pengujian dilakukan dengan data uji yang *bit rate*-nya 32 kbit/s. Pengujian dilakukan sebanyak 30 judul berbeda (sesuai dengan data *bit rate* lagu yang sudah dikonversi). Lima buah sampel data uji sebelum dan sesudah konversi *bit rate* beserta fitur-fiturnya dapat dilihat pada Tabel 1 dan 2 berikut ini. Dari kedua tabel terlihat bahwa *fitur key* dan *mode* tidak mengalami perubahan, sedangkan fitur *loudness*, energi dan *tempo* mengalami perubahan, walaupun kecil.

Tabel 1. Sampel Data Uji dengan 5 Fitur dan *Bit Rate* 64 kbit/s

No	Judul	Penyanyi	Key	Mode	Loudness	Energi	Tempo
1	Afterlife	Avenged Sevenfold	2	0	-6.951	0.807124	110.382
2	Girlfriend	Avril Lavigne	7	1	-6.336	0.921211	163.983
3	Lua	Bright Eyes	2	1	-18.674	0.13019	105.369
4	Is This Love	Bob Marley	6	0	-9.065	0.558251	122.215
5	I Wanna Go	Britney Spears	5	1	-8.205	0.531336	130.018

Tabel 2. Sampel Data Uji dengan 5 Fitur dan *Bit Rate* 32 kbit/s

No	Judul	Penyanyi	Key	Mode	Loudness	Energi	Tempo
1	Afterlife	Avenged Sevenfold	2	0	-4.558	0.955007	110.001
2	Girlfriend	Avril Lavigne	7	1	-6.207	0.908809	163.964
3	Lua	Bright Eyes	2	1	-18.111	0.126862	104.552
4	Is This Love	Bob Marley	6	0	-8.345	0.56503	122.184
5	I Wanna Go	Britney Spears	5	1	-7.956	0.536442	129.98

Disebabkan oleh nilai besaran yang tidak berimbang di antara kelima fitur tersebut, maka diterapkan normalisasi menggunakan *min-max* untuk menghasilkan bobot  $w$  sebagai elemen fitur musik sebagaimana persamaan (3), dengan  $f$  adalah nilai besaran fitur yang akan dinormalisasi  $f_{min}$  adalah nilai minimal dari fitur  $f$ , dan  $f_{max}$  adalah nilai maksimum dari fitur  $f$  untuk seluruh dokumen di dalam koleksi (280 file musik).

$$w = \frac{f - f_{min}}{f_{max} - f_{min}} \quad (6)$$

Akurasi dihitung sebagai metrik pengukuran performa, apakah benar hasil pen- ranking pertama sesuai dengan lagu atau musik yang diinginkan dalam kueri. Hasil tidak ditampilkan dalam bentuk daftar *ranking* lagu, melainkan informasi mengenai musik (judul lagu dan artis). Berbeda dengan *music retrieval* (temu kembali musik) yang input kuerinya adalah teks yang mengandung judul lagu dan/atau dapat ditambah dengan informasi artis yang mempopulerkannya, maka sistem MIR adalah kebalikannya. Oleh karena itu, metrik akurasi dipakai untuk memeriksa apakah sistem dapat menemukan judul lagu yang tepat atau tidak.

### 3.3. Hasil Pengujian

Dari kueri berupa file lagu atau musik, akan dicari informasi mengenai musik tersebut dari dalam koleksi. Tahap pertama adalah perhitungan kemiripan berdasarkan jarak antara kueri terhadap koleksi, menggunakan *k-NN*. Diambil  $k=\{5, 7, 10\}$  tetangga yang memiliki jarak terdekat dengan kueri sebagai kandidat hasil. Kemudian dari tetangga tersebut, barulah dihitung *cosine similarity* antara kueri dengan kandidat, dan yang menempati ranking pertama, akan ditampilkan

informasinya (judul lagu, artis, fitur dan *genre*). Apabila *ranking* pertama tidak mencapai nilai *threshold*, maka tidak ada hasil yang ditampilkan. Artinya informasi tentang lagu atau musik tidak ditemukan di dalam koleksi. Nilai *threshold* dipilih sangat tinggi, karena kita ingin menemukan musik yang benar-benar cocok atau sama dengan kueri.

Tabel 3. Hasil Perhitungan *k-NN* dan *Cosine Similarity*

Pasangan (Kueri, Dokumen)	Judul	Penyanyi	k-NN	Cosine sim
1,1	Afterlife	Avenged Sevenfold	2.427649	0.999769
1,2	Girlfirend	Avril Lavigne	54.25144	0.995119
1,3	Lua	Bright Eyes	14.91299	0.986454
1,4	Is This Love	Bob Marley	13.62543	0.994436
1,5	I Wanna Go	Britney Spears	20.59515	0.994969

Sebagai contoh, kueri adalah dokumen 1 dari Tabel 2 (32 kbit/s), yaitu lagu/musik berjudul "After life" oleh *Avenged Sevenfold*. Perhitungan *k-NN* dan *cosine similarity* terhadap 5 dokumen di tabel 1 (64 kbit/s) adalah sebagai mana terlihat pada Tabel 3. Dari nilai *cosine similarity*, maka informasi lagu "Afterlife" ditampilkan pada antarmuka sebagai hasil pencarian, karena memiliki ranking pertama dan *cosine similarity* melebihi *threshold*.

Tabel 4. Hasil Pengujian

BitRate	k	Threshold	Akurasi	Precision	Recall
32	5	0.997	0.6667	0.9508	0.6823
32	5	0.998	0.6667	0.9508	0.6823
32	5	0.999	0.6778	0.9661	0.6785
32	7	0.997	0.6667	0.9508	0.6823
32	7	0.998	0.6667	0.9508	0.6823
32	7	0.999	0.6778	0.9661	0.6785
32	10	0.997	0.6667	0.9508	0.6823
32	10	0.998	0.6667	0.9508	0.6823
32	10	0.999	0.6778	0.9661	0.6785

Untuk keseluruhan pengujian (30 data uji) dengan skema variasi pengujian berupa  $k=\{5,7,10\}$ , dan  $threshold=\{0.997, 0.998, 0.999\}$ , diperoleh hasil seperti ditunjukkan pada Tabel 4. Akurasi terbaik diperoleh dengan kombinasi  $k=5$  dan  $threshold 0.999$ , yaitu nilai akurasi 0.6778 dengan *precision* dari nilai akurasi tertinggi adalah 0.9661 dan nilai *recall* adalah 0.6785. Walaupun untuk semua  $k$ ,  $threshold 0.999$  memberikan nilai yang sama, namun memilih  $k$  yang lebih kecil dapat menghemat waktu komputasi karena mengurangi jumlah perhitungan *cosine similarity*.

#### 4. Kesimpulan

*Music Information Retrieval* adalah sistem temu kembali informasi musik dengan kueri berupa potongan musik atau musik dengan *bit rate* sembarang. Untuk menemukan informasi yang sesuai, maka kueri dan koleksi berupa musik digital perlu direpresentasikan oleh suatu vektor angka, agar dapat diukur kemiripan maupun jarak antara vektor keduanya. Fitur yang dibangkitkan oleh *Echonest API* dapat digunakan sebagai representasi vektor untuk memenuhi fungsi *retrieval*.

Pengujian dengan menerapkan 5 fitur dari sejumlah fitur *Echonest* menggunakan pengukuran jarak dengan *k-NN* dan *cosine similarity* untuk dataset yang kecil, telah memberikan hasil yang tinggi dari segi *precision*, yaitu di atas 95%. Sedangkan dari segi akurasi, dan *recall*, masih di kisaran 67-68%. Jumlah  $k=5$  sudah cukup untuk menghasilkan akurasi yang sama

dibandingkan menggunakan  $k$  yang lebih besar, sehingga nilai  $k$  yang lebih kecil dapat dipilih untuk menghemat *cost* komputasi.

Saran untuk penelitian selanjutnya, yaitu menambah fitur lainnya sebagai elemen vektor fitur. Selain itu, informasi frekuensi suara dengan *MFCC* sebagai representasi dari dokumen dan kueri berdasarkan analisis intrinsik lagu juga dapat dipertimbangkan.

#### Daftar Pustaka

- [1] Anan, Yoko., Hatano, Kohei., Bannai, Hideo., dan Takeda, Masayuki (2011). Music Genre Classification Using Similarity Functions. *ISMIR 2011. Department of Informatics: Kyushu University*. 2011.
- [2] Pramudita, Krisnaldi Eka., Khodra, Masayu Leylia., dan Santoso, Oerip S. (2012) Klasifikasi Genre Musik Digital Simbolik Dengan Support Vektor Mechine. *Sekolah Teknik Elektro dan Informatika: Institut Teknologi Bandung*. 2012.
- [3] Ridoean, Johannes Andre (2017) Rancang Bangun Aplikasi Musicmoo Dengan Metode MIR (Music Information Retrieval) Pada Modul Mood, Genre Recognition, Dan Tempo Estimation. *Undergraduate thesis, Institut Teknologi Sepuluh Nopember*.
- [4] Rodrigo Castellon and Chris Donahue and Percy Liang (2021), Codified audio language modeling learns useful representations for music information retrieval, In *arXiv: 2107.05677*
- [5] Szymon Janusz Szczepański dan Michał Stanisław Szyca (2014), Music Information Retrieval: A case study of MIR in modern rock and metal music, *Master thesis of Blekinge Institute of Technology, Sweden*.
- [6] Jehan, Tristan. The Echonest Analyze Documentation. [online] Available [http://developer.echonest.com/docs/v4/\\_static/AnalyzeDocumentation.pdf](http://developer.echonest.com/docs/v4/_static/AnalyzeDocumentation.pdf), 2010.
- [7] Jehan, Tristan. Creating Music by Listening. M.S. Media Arts and Sciences: *Massachusetts Institute Of Technology*. 2005.
- [8] Astawa, I Gede Suta Lascarya., Muliantara, Agus., dan Dewi, Kadek Cahya. Identifikasi Lagu Menggunakan Algoritma K-Nearest Neighbour – Cosine Similarity (KNNCS). *PSNTI 2012. Program Studi Teknik Informatika: Universitas Udayana*. 2012.
- [9] Florentine, Mary., Popper, Arthur N., and Fay, Richard R. Loudness (pp.1-12). ISBN: 978-1-4419-6711-4. *Springer. London, UK : New York Dordrecht Heidelberg*. 2010.
- [10] Batista, Gustavo E. A. P. A., and Monard, Maria Carolina. A Study of K-Nearest Neighbour as an Imputation Method. *Brazil: University of S\*ao Paulo*. 2011.
- [11] Manning, D. Christopher., Raghavan, Prabhakar., and Schutze, Hinrich. Introduction to Information Retrieval. ISBN-13 978-0-511-41405-3. *Cambridge, UK : Cambridge University Press*. 2008.