

PERBANDINGAN *DISTANCE MEASURE* PADA *K-MEANS CLUSTERING* UNTUK PENGELOMPOKAN MUSIK TERHADAP SUASANA HATI

I Gede Harsemadi

Program Studi Sistem Informasi, STMIK STIKOM Bali
Jl. Raya Puputan No. 86 Renon, Denpasar, Bali
Email : harsemadi@stikom-bali.ac.id¹⁾

Abstrak

Dalam bidang *information retrieval*, *MIR (Music Information Retrieval)* merupakan bagian bidang keilmuan yang berhubungan dengan pencarian informasi dalam media berupa musik dengan penerapan algoritma *data mining* untuk pengelompokan/*clustering* data musik.

Penelitian ini diawali dengan tahap pra-pengolahan data yaitu mendapatkan nilai fitur pada musik dengan *feature extraction*. Tahapan *feature extraction* terdiri dari tiga proses, yaitu pengambilan sampel lagu (bagian *refrain*), penerapan *fast fourier transform*, dan penerapan *spectral analysis* untuk mendapatkan nilai *spectral feature* yang menjadi atribut dasar untuk dilakukannya pengelompokan musik terhadap suasana hati.

Proses *clustering/pengelompokan K-Means* ini menggunakan 200 file musik ke dalam 4 jenis suasana hati model Thayer. Dalam proses pengelompokan ini dilakukan perhitungan jarak setiap cluster menggunakan *distance measure* yang terdiri dari *euclidean distance*, *city block distance*, dan *cosine distance*. Selanjutnya dihitung nilai *silhouette coefficient* pada *K-Means* yang menunjukkan seberapa baik dan optimal suatu objek ditempatkan dalam suatu cluster. Akhirnya dari ketiga nilai *distance measure* ini diperoleh nilai *silhouette coefficient* tertinggi yaitu 0,64018 dengan waktu pemrosesan tercepat yaitu 0,0789935 pada *euclidean distance*.

Kata kunci : *MIR, K-Means, Data Mining, Music Mood*

1. Pendahuluan

Dalam beberapa dekade terakhir, era digital telah menciptakan inovasi teknologi yang membawa perubahan menyeluruh pada cara manusia menikmati musik. Faktor-faktor seperti akses internet yang meluas, peningkatan *bandwidth* dalam hal akses file musik atau penggunaan format audio berkualitas tinggi seperti .mp3 berperan besar dalam perubahan tersebut. Adanya penawaran dan permintaan dalam industri musik digital saat ini menunjukkan adanya kebutuhan akan cara otomatis untuk menemukan suatu lagu yang tepat dan relevan diinginkan oleh pendengarnya dalam konteks tertentu dari suatu database musik yang besar. Basis data

musik yang besar ini pun akan berguna jika pengunanya dapat menemukan apa yang mereka cari dengan cepat dan efisien.

Basis data musik digital hingga kini memerlukan organisasi file dan mekanisme pencarian musik yang semakin maju, fleksibel dalam penggunaannya, serta dapat disesuaikan dengan kebutuhan penggunanya. Hal ini berkaitan dengan fungsi musik yang bersifat sosial dan psikis, dan biasanya pengelompokan musik seperti itu akan fokus pada informasi mengenai gaya, jenis, kesamaan musik dan suasana hati yang terkandung dalam musik tersebut [1].

Dalam bidang *information retrieval*, *MIR (Music Information Retrieval)* merupakan bagian bidang keilmuan yang berhubungan dengan pencarian informasi dalam media berupa musik dan hubungan metadata berbagai file musik dalam suatu basis data yang saling terhubung. Hal inilah yang hingga kini menjadikan *MIR* menarik untuk diteliti, selain untuk menganalisa representasi metadata di dalam musik, penelitian dalam bidang *MIR* juga telah dilakukan dengan penerapan algoritma *data mining* untuk klasifikasi [1][2][3][4] dan pengelompokan/*clustering* data musik [5].

Clustering merupakan salah satu metode *data mining* yang bersifat tanpa supervisi (*unsupervised*). Tujuan dari *clustering* ini adalah untuk mengelompokkan data ke dalam suatu kelompok / *cluster*, sehingga objek-objek yang ada pada *cluster* tersebut memiliki kemiripan yang sangat besar dengan objek-objek lainnya yang ada pada *cluster* yang sama, namun memiliki ketidakmiripan yang tinggi dengan objek-objek yang ada pada *cluster* lainnya [5].

Untuk proses *clustering* dalam penelitian ini menggunakan algoritma *K-Means*. *K-Means* merupakan salah satu algoritma yang melakukan pengelompokan data dengan sistem partisi, dimana algoritma *data mining* ini melakukan proses pemodelan tanpa supervisi (*unsupervised learning*) dan berusaha mengelompokkan data yang ada ke dalam beberapa kelompok *k*, dimana data dalam satu kelompok mempunyai karakteristik yang sama satu sama lainnya dan mempunyai karakteristik yang berbeda dengan data yang ada di dalam kelompok yang lain. Dengan kata lain, algoritma ini berusaha untuk meminimalkan variasi antar data yang ada di dalam suatu *cluster* dan memaksimalkan variasi dengan data yang ada di *cluster* lainnya.

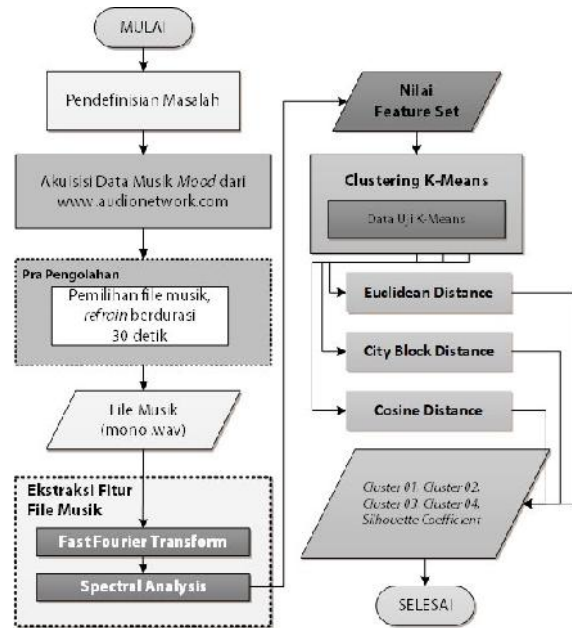
Hal yang menarik dalam algoritma *K-Means* ini adalah untuk mengetahui ciri-ciri *cluster* yang baik atau optimal adalah jika *cluster* yang dihasilkan tersebut memiliki data dengan tingkat kemiripan (*similarity*) yang tinggi pada *cluster* yang sama dan tingkat kemiripan yang rendah pada *cluster* yang berbeda. Penelitian ini dilakukan untuk mengukur kemiripan data dalam suatu *cluster* dapat dilakukan dengan menerapkan metode *Distance Measure* pada perhitungan algoritma *K-Means* dan *Silhouette Coefficient*. *Silhouette Coefficient* digunakan untuk melihat kualitas dan kekuatan *cluster* [6].

Penelitian ini diawali dengan tahap pra-pengolahan data yaitu mendapatkan nilai fitur pada musik dengan *feature extraction* [7] atau ekstraksi ciri dari sebuah file lagu untuk menemukan ciri yang untuk masuk ke tahapan pengolahan data berikutnya. Tahapan *feature extraction* terdiri dari tiga proses, yaitu pengambilan sampel lagu (bagian *refrain*), penerapan *fast fourier transform*, dan penerapan *spectral analysis* untuk mendapatkan nilai *spectral feature* yang menjadi atribut dasar untuk dilakukannya pengelompokan musik terhadap suasana hati.

Proses *clustering*/pengelompokan *K-Means* ini menggunakan 200 file musik ke dalam 4 jenis suasana hati yaitu 1. *Contentment*, 2. *Exuberance*, 3. *Depression*, 4. *Anxious* [8] [Thayer]. Selanjutnya dalam proses pengelompokan ini dilakukan perhitungan jarak setiap *cluster* menggunakan *distance measure* yang terdiri dari *euclidean distance*, *city block distance*, dan *cosine distance*. Setelah hasil perhitungan *distance measure* ini diperoleh, selanjutnya diambil rata-rata nilai tersebut yang digunakan sebagai nilai *silhouette coefficient* pada *K-Means* yang menunjukkan seberapa baik dan optimal suatu objek ditempatkan dalam suatu *cluster*. Akhirnya dari ketiga nilai *silhouette coefficient* ini selanjutnya akan dibandingkan dan diperoleh nilai *silhouette coefficient* tertinggi berdasarkan ketiga metode *distance measure* ini.

2. Metode Penelitian

Dalam menentukan *distance measure* terbaik pada pengelompokan musik terhadap suasana hati ini terdiri dari beberapa tahapan utama yaitu pra-pengolahan data, tahapan ekstraksi fitur, *clustering K-Means*, dan pengujian *Silhouette Coefficient*. Secara keseluruhan dapat dilihat pada gambar 1.



Gambar 1. Alur tahapan penelitian

2.1 Pra-Pengolahan Data

Tahapan ini diawali dengan mendefinisikan permasalahan yaitu bagaimana menghasilkan nilai *distance measure* terbaik pada *K-Means* untuk mengelompokkan musik terhadap suasana hati. Sebagai acuan sumber basis data musik yang digunakan dalam penelitian ini adalah audionetwork.com. didalamnya terdapat 56 jenis anotasi kata sifat dalam mengelompokkan jenis musik berdasarkan *mood* atau suasana hati yang kemudian disederhanakan menjadi 4 jenis *mood* model Thayer yaitu : 1. *Contentment* (menenangkan, relaksasi, damai), 2. *Exuberance* (riuh, bersemangat, bergembira), 3. *Depression* (sedih, murung, depresi, duka), dan 4. *Anxious* (amarah, kacau, konflik). Penelitian ini menggunakan 200 file musik instrumental saja tanpa ada lirik dan vokal di dalamnya.

File musik ini selanjutnya dipilah hanya pada bagian *refrain*. Bagian *refrain* ini biasanya diulang-ulang saat lagu dimainkan, dan merupakan bagian yang paling sering menunjukkan *mood* yang tersirat di dalam musik. Durasi *refrain* klip musik ini ditentukan hanya berdurasi 30 detik [9]. Hasil klip musik ini kemudian disimpan dalam format *.wav* dan *mono audio channel*.

Data musik yang digunakan pada penelitian ini menggunakan *dataset* musik yang sudah dikelompokkan berdasarkan kategori suasana hati dari situs www.audionetwork.com. Situs ini mengelompokkan musik menggunakan 56 anotasi/tag kata sifat untuk pelabelan suasana hati dalam musik, dan label ini telah ditentukan sebelumnya oleh para pakar bidang musik. Sebagai langkah penyederhanaan dalam kata sifat, kami menggunakan jenis kata sifat suasana hati/*mood* model Thayer [8] yaitu yaitu 1) *contentment*/ ketenangan; relaksasi, 2) *exuberance*/ bersemangat; riuh; gembira, 3) *depression*/ depresi; sedih, dan 4) *anxious*/ cemas; kalut; kacau, dimana masing-masing kata sifat tersebut terdiri

dari 50 file musik dan secara keseluruhan terdapat 200 file musik yang digunakan dalam penelitian ini. Keseluruhan file musik ini masuk ke tahapan pra-pengolahan file musik diawali dengan mengambil potongan klip musik berdurasi 30 detik hanya pada bagian *refrain* dari musik tersebut [9]. Hasil klip file musik tersebut disimpan dengan format .wav dengan *mono audio channel*.

2.2 Ekstraksi Fitur Musik

Tahapan ekstraksi fitur file musik yang telah diinput (mono *.wav) akan diproses untuk mendapatkan ciri-ciri khusus yang disebut dengan ekstraksi fitur. Proses ekstraksi dimulai dengan metode FFT. *Fast Fourier Transform* adalah suatu algoritma untuk menghitung transformasi fourier diskrit dengan cepat dan efisien. Karena banyak sinyal-sinyal dalam sistem komunikasi yang bersifat kontinu, sehingga untuk kasus sinyal kontinu seperti sinyal suara dapat menggunakan transformasi fourier. Transformasi Fourier didefinisikan oleh persamaan (1):

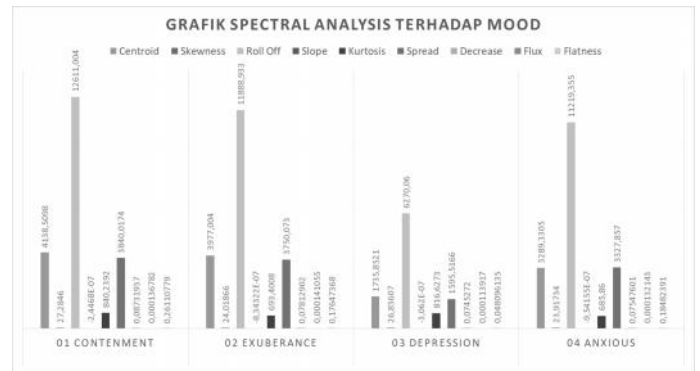
$$s(f) = \int_{-\infty}^{\infty} s(t)e^{-j2\pi ft} dt \quad (1)$$

Dimana $s(f)$ adalah sinyal dalam domain frekuensi (*frequency domain*), $s(t)$ adalah sinyal dalam domain waktu (*time domain*), dan $e^{-j2\pi ft}$ adalah konstanta dari nilai sebuah sinyal, f adalah frekuensi dan t adalah waktu. FFT merupakan salah satu metode untuk transformasi sinyal suara dalam domain waktu menjadi sinyal dalam domain frekuensi, artinya proses perekaman suara disimpan dalam bentuk digital berupa gelombang *spectrum* suara yang berbasis frekuensi sehingga lebih mudah dalam menganalisa *spectrum* frekuensi suara yang telah direkam [10].

Tahapan selanjutnya adalah proses mengekstrak fitur dari sebuah data digital dengan menggunakan metode *spectral analysis* [11]. Fitur-fitur yang dihasilkan ini akan menentukan kelas dari sinyal input yang masuk. Ekstraksi fitur melibatkan analisis input dari sinyal audio. Dalam *Music Information Retrieval*, beberapa peneliti sepakat bahwa ekstraksi fitur memegang peranan yang lebih penting daripada fase lainnya baik untuk tujuan *clustering* musik maupun untuk tujuan klasifikasi musik.

Beberapa metode *spectral analysis* yang digunakan dalam penelitian ini diantaranya [12], pertama berdasarkan pada properti statistik (*statistical property*) dari sinyal audio, dimana fitur audio yang dianalisis berdasarkan panjang blok sinyal audio dan tingkat nada yang diperoleh dari proses ekstraksi. Dalam hal ini nilai fitur audio diperoleh dengan menggunakan analisis *spectral skewness* dan *kurtosis*. Kedua, fitur audio diperoleh berdasarkan bentuk spektral (*spectral shape*) hal ini dapat diketahui berdasarkan *timbre* (warna suara/audio), *pitch* (tinggi-rendah nada) dan *loudness* (kuat-lemah suara). Untuk mendapatkan nilai fitur audio pada *spectral shape* ini diperoleh dengan *spectral*

centroid, *rolloff*, *slope*, *spread*, *decrease*, dan *flux*. Ketiga, fitur audio diperoleh berdasarkan properti sinyal (*signal properties*) audio, dimana fitur audio yang dianalisis berdasarkan nada disepanjang sinyal audio, hal ini menggambarkan keharmonisan dalam musik. Untuk mendapatkan nilai fitur audio berdasarkan *signal properties* ini dengan menggunakan *spectral flatness*. Berikut ini pada gambar 2 adalah rata-rata nilai fitur dari 9 *spectral analysis* yang ada pada data musik terhadap keempat jenis *mood*.

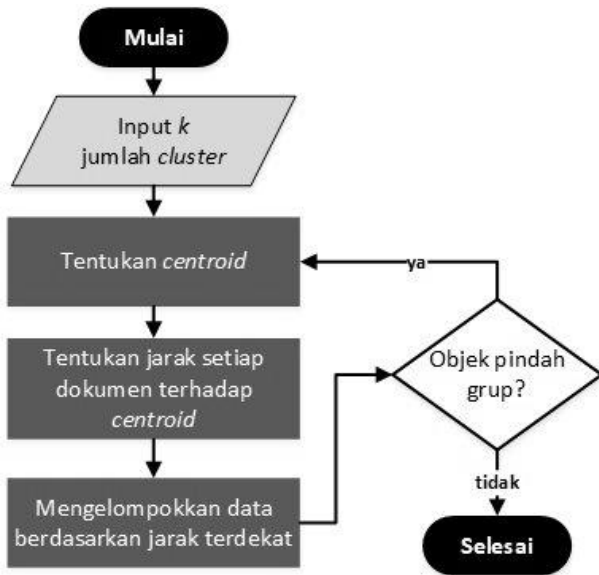


Gambar 2. Grafik spectral analysis dengan mood

Tahapan *spectral analysis* ini menghasilkan seperangkat nilai atau *feature set* yang terdiri dari 9 atribut data untuk setiap file musik. Dan nilai-nilai tersebut yang selanjutnya akan digunakan sebagai nilai input pada *dataset* musik untuk tahapan *clustering K-Means*.

2.3 K-Means Clustering

Tahapan pengelompokan data menggunakan *K-Means* ini diawali dengan penentuan jumlah *cluster* (k) untuk mengelompokkan data uji sebagai data input yang akan dikelompokkan pada sejumlah k *cluster*. Penelitian ini menggunakan 200 baris data dalam *dataset* dan ada 4 *cluster mood* yang perlu dibentuk, maka k adalah 4 *cluster*. Algoritma ini selanjutnya mengambil 4 *record* data secara acak dari *dataset* sebagai pembentuk *cluster* awal. Setiap *cluster* yang terbentuk akan dihitung rata-ratanya (*means*), dimana rata-rata dari suatu *cluster* adalah rata-rata dari semua *record* yang terdapat pada *cluster* tersebut. Selanjutnya data dialokasikan ke *cluster* terdekat, dalam hal ini menggunakan perhitungan *euclidean distance*, *city block distance*, dan *cosine distance*. Selanjutnya menghitung kembali nilai *centroid* dari rata-rata setiap *cluster* yang ada, dan alokasikan kembali data ke *cluster* terdekat. Hal ini dilakukan berulang kali sampai terbentuk *cluster* yang stabil dan prosedur algoritma *K-Means* selesai. *Cluster* yang stabil terbentuk ketika iterasi atau pengulangan perhitungan *K-Means* tidak membuat *cluster* baru dan tidak ada data yang berpindah [5].



Gambar 3. Alur K-Means clustering

Hasil dari pengelompokkan ini dapat dilihat pada Tabel 1, yaitu sejumlah contoh 40 dari 200 baris data yang digunakan dalam penelitian ini yang memiliki 9 nilai *spectral feature* dan sebuah kolom hasil *K-Means clustering* dari setiap data tersebut.

Tabel 1. Hasil K-Means clustering

No.	File Name	Centroid	Skewness	Roll Off	Slope	Kurtosis	Spread	Decrease	Flux	Flatness	Cluster
1	C-03.wav	4178	25,7742	13344,1	0	791,261	4689,57	0,075823	1E-04	0,118937	4
2	C-04.wav	10968,9	25,4988	18627,9	0	767,338	6361,49	0,065905	1E-04	0,861021	2
3	C-05.wav	1845,22	26,9436	7371,52	0	796,103	1232,47	0,050601	2E-04	0,110781	1
4	C-06.wav	1905,5	31,3605	15937,6	0	1070,45	2448,85	0,138777	2E-04	0,546685	4
5	C-08.wav	10849,9	30,3851	14205	0	996,228	5441,32	0,038984	1E-04	0,330043	2
6	C-10.wav	2716,1	27,7774	13150,4	0	828,682	4525,55	0,088979	2E-04	0,121064	4
7	C-11.wav	8555,7	29,6754	11955,9	0	981,934	4436,76	0,077157	1E-04	0,206485	4
8	C-13.wav	2356,23	27,2384	10761,3	0	826,658	3417,73	0,081104	1E-04	0,125012	1
9	C-17.wav	1944,41	29,4717	10879,7	0	912,204	1514,61	0,095023	1E-04	0,108546	1
10	C-19.wav	9481,99	29,846	13914,4	0	925,099	5302,13	0,029553	1E-04	0,15611	2
11	E-103.wav	6387,14	28,0958	11977,4	0	864,943	3914,35	0,119639	2E-04	0,277814	4
12	E-104.wav	1836,26	17,6905	10955,1	-1E-06	469,51	2490,53	0,044728	1E-04	0,117882	1
13	E-105.wav	3511,57	23,0979	12859,8	-1E-06	633,647	5046,12	0,086934	1E-04	0,120199	4
14	E-106.wav	2234,05	17,9433	9362,37	-1E-06	404,46	2675,12	0,071403	1E-04	0,082563	1
15	E-107.wav	4484,37	23,1229	12978,2	-1E-06	646,046	5023,64	0,061833	2E-04	0,233484	4
16	E-108.wav	2918,17	24,3986	12128	-1E-06	683,669	4072,4	0,053988	2E-04	0,104815	4
17	E-06.wav	4215,33	22,6779	10018,8	-1E-06	645,14	3268,82	0,076222	2E-04	0,089479	1
18	E-07.wav	2396,1	20,9309	11493,1	-2E-06	535,688	3309,63	0,059089	1E-04	0,120611	1
19	E-08.wav	2471,96	26,2151	9717,5	-1E-06	806,316	2914,43	0,055935	2E-04	0,137592	1
20	E-09.wav	2347,37	16,8	10729,1	-2E-06	373,396	2736,7	0,04547	1E-04	0,127931	1
21	D-101.wav	399,209	28,8065	1000,81	0	929,427	319,066	0,125342	1E-04	0,030721	3
22	D-102.wav	2228,23	27,2111	7672,84	0	824,934	2597,76	0,059467	1E-04	0,046076	1
23	D-106.wav	775,132	24,3129	3669,62	-1E-06	708,002	1002,87	0,067806	1E-04	0,035686	3
24	D-107.wav	1676,3	29,6833	7791,22	0	933,049	1173,3	0,032754	1E-04	0,075464	1
25	D-110.wav	640,412	23,1811	3303,73	0	648,22	545,971	0,074764	1E-04	0,039886	3
26	D-27.wav	1200,04	20,5434	7931,11	-1E-06	531,016	1764,95	0,045152	9E-05	0,069012	1
27	D-29.wav	1093,02	22,4031	5768,08	0	595,604	984,206	0,104797	8E-05	0,052198	3
28	D-31.wav	979,724	25,3797	7608,27	0	728,399	1805,34	0,058774	9E-05	0,077983	1
29	D-32.wav	1490,27	22,5742	13537,8	0	609,305	3108,54	0,04703	8E-05	0,115481	4
30	D-34.wav	2617,62	30,025	5606,66	0	937,588	1350,21	0,206274	1E-04	0,197179	3
31	A-07.wav	9147,76	23,0119	15647	0	626,755	4983,5	0,059494	1E-04	0,717173	2
32	A-100.wav	3564,88	24,5331	11611,5	-1E-06	720,802	4248,92	0,069276	2E-04	0,218426	4
33	A-101.wav	6191,79	28,0768	13182,7	0	892,705	5320,99	0,084985	2E-04	0,180203	4
34	A-103.wav	2484,85	26,5977	9642,17	-1E-06	801,707	2922,19	0,065734	9E-05	0,09365	1
35	A-105.wav	3413,17	32,4388	18541,8	0	1199,05	6120,2	0,065595	1E-04	0,813736	4
36	A-108.wav	1932,84	19,8781	9577,6	-2E-06	470,847	2483,98	0,042847	1E-04	0,09096	1
37	A-109.wav	2814,54	21,6677	9362,37	-1E-06	597,842	2757,53	0,039746	1E-04	0,075916	1
38	A-110.wav	4165,07	29,6385	11955,9	-1E-06	917,051	3874,07	0,056889	1E-04	0,138223	4
39	A-62.wav	2176,97	20,1303	9760,54	-1E-06	569,554	2588,24	0,10107	2E-04	0,083585	1
40	A-06.wav	3923,34	23,6317	9297,8	-1E-06	683,313	2934,5	0,078913	1E-04	0,132413	1

2.4 Distance Measure

K-Means clustering melakukan pengelompokkan data yang memiliki kemiripan nilai atribut yang cukup baik,

namun perlu beberapa pengukuran untuk menentukan kemiripan antara objek satu dengan lainnya. Untuk menentukan kemiripan data tersebut dapat menggunakan pengukuran *distance measure* [13]. Berikut ini terdapat tiga jenis *distance measure* yang digunakan dalam algoritma *K-Means* sebagai berikut :

1. Euclidean Distance

Euclidean Distance adalah matriks yang paling sering digunakan untuk menghitung kesamaan dua vektor. Persamaan *Euclidean Distance* adalah akar dari kuadrat perbedaan 2 vektor, dapat dilihat pada persamaan (2).

$$D_{L2}(X_2, X_1) = \|X_2 - X_1\|_2 = \sqrt{\sum_{j=1}^P (X_{2j} - X_{1j})^2} \quad (2)$$

Keterangan :

P = Dimensi Data

X₁ = Posisi titik 1

X₂ = Posisi titik 2

2. City Block Distance

City Block Distance ini juga dikenal sebagai *Manhattan Distance*. Analagi pengukuran ini diilustrasikan seperti jarak antara titik-titik blok kota. Jarak blok kota dihitung sebagai jarak di x ditambah jarak y, yang mirip dengan cara bergerak di kota (seperti kota Manhattan) di mana kita harus berkeliling bangunan yang ada di kota daripada langsung melintasinya. Persamaannya dapat dilihat pada persamaan (3)

$$D_{L1}(X_2, X_1) = \|X_2, X_1\|_1 = \sum_{j=1}^P (X_{2j} - X_{1j})^2 \quad (3)$$

Keterangan :

P = Dimensi Data

|.| = Nilai absolut

X₁ = Posisi titik 1

X₂ = Posisi titik 2

3. Cosine Distance

Cosine Distance digunakan untuk menghitung jarak kosinus antara dua variable. Biasa dikenal dengan kemiripan kosinus, jarak sudut kosinus, dan kesamaan kosinus. Dirumuskan dalam persamaan (4)(5) :

Diawali dengan menghitung *cosine similarity* :

$$\text{Cosine Similarity} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}} \quad (4)$$

Untuk menghitung *cosine distance* :

$$\text{Cosine Distance} = 1 - \text{Cosine Similarity} \quad (5)$$

2.5 Pengujian *Silhouette Coefficient*

Silhouette Coefficient digunakan untuk melihat kualitas dan kekuatan *cluster*, seberapa baik suatu objek ditempatkan dalam suatu *cluster*. Metode ini merupakan gabungan dari metode *cohesion* dan *separation*. Tahapan perhitungan *Silhouette Coefficient* adalah sebagai berikut:

1. Hitung rata-rata jarak dari suatu data misalkan i dengan semua data lain yang berada dalam satu *cluster*

$$a(i) = \frac{1}{|A|-1} \sum_{j \in A, j \neq i} d(i, j) \quad (6)$$

2. Hitung rata-rata jarak dari data i tersebut dengan semua data di *cluster* lain, dan diambil nilai terkecilnya.

$$d(i, C) = \frac{1}{|A|} \sum_{j \in C} d(i, j) \quad (7)$$

dengan $d(i, C)$ adalah jarak rata-rata data i dengan semua objek pada *cluster* lain C dimana $A \neq C$.

$$b(i) = \min_{C \neq A} d(i, C) \quad (8)$$

3. Nilai *Silhouette Coefficient*-nya adalah :

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (9)$$

Nilai *silhouette coefficient* dapat bervariasi antara -1 hingga 1. Hasil *clustering* dikatakan baik jika nilai *silhouette coefficient* bernilai positif ($a(i) < b(i)$) dan $a(i)$ mendekati 0, sehingga akan menghasilkan nilai *silhouette coefficient* yang maksimum yaitu 1 saat $a(i)=0$. Dengan demikian, jika $s(i)=1$ berarti objek i sudah berada dalam *cluster* yang tepat. Jika nilai $s(i)=0$ maka objek i berada diantara dua *cluster* sehingga objek tersebut tidak jelas harus dimasukkan ke dalam *cluster* A atau *cluster* B. Akan tetapi, jika $s(i)=-1$ artinya struktur *cluster* yang dihasilkan *overlapping*, sehingga objek i lebih tepat dimasukkan ke dalam *cluster* yang lain [6]. Nilai rata-rata *silhouette* dari tiap objek dalam suatu *cluster* adalah suatu ukuran yang menunjukkan seberapa ketat/fit dan optimal sekelompok data berada pada *cluster* tersebut. Ukuran nilai *silhouette coefficient* :

$0.7 < SC \leq 1$ *strong structure*

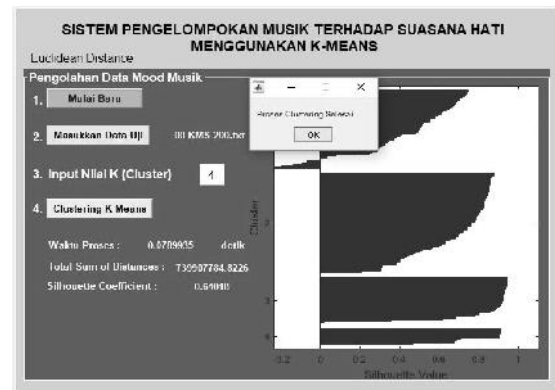
$0.5 < SC \leq 0.7$ *medium structure*

$0.25 < SC \leq 0.5$ *weak structure*

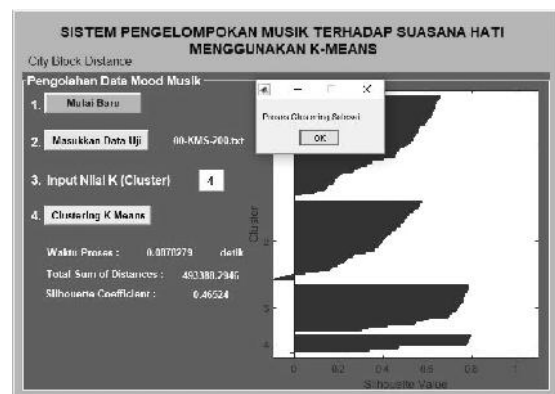
$SC \leq 0.25$ *no structure*

3. Hasil dan Pembahasan

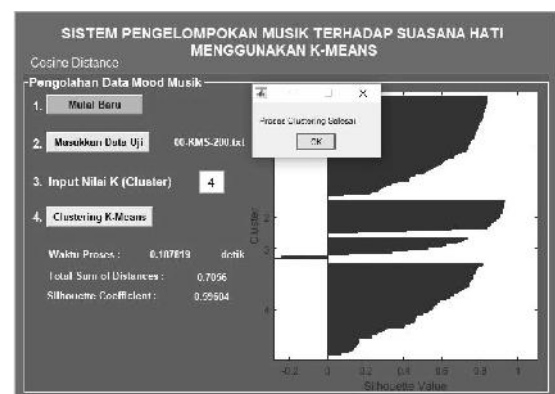
Untuk menghasilkan nilai *silhouette coefficient* terbaik berdasarkan perbandingan nilai *distance measure*, maka telah dikembangkan sistem untuk menerapkan *K-Means* untuk mengelompokkan musik terhadap *mood* pada Gambar 4, 5, dan 6. Dengan tahapan kerja sistem sebagai berikut :



Gambar 4. Pengukuran euclidean distance



Gambar 5. Pengukuran city block distance



Gambar 6. Pengukuran cosine distance

Data *input* untuk proses *clustering* ini dibentuk dari data set berjumlah 200 baris data nilai *spectral feature* dengan format *.txt yang diperoleh pada tahapan pra-pengolahan *mood* musik. Berikut ini pada Gambar 4 menunjukkan antarmuka proses *clustering* menggunakan algoritma *K-Means*. Terdapat tombol “Mulai Baru” yang digunakan untuk memulai baru proses *clustering*. Kemudian dilanjutkan dengan memasukkan data uji yang dalam format data *.txt, sistem akan membuka window untuk memasukkan file data uji. Selanjutnya akan ditentukan berapa jumlah k *cluster* yang akan

terbentuk dari data uji ke dalam kelompok-kelompok. Dalam tahapan ini nilai k yang digunakan adalah 4, yang mewakili jumlah kategori *mood* musik.

Tahapan selanjutnya adalah memulai proses clustering yaitu dengan menekan tombol “Clustering K-Means”. Sistem akan secara otomatis melakukan *clustering* menggunakan *K-Means* dan ketiga metode *distance measure* juga dihitung di masing-masing antarmuka sistem. Hasilnya pengelompokan data yang divisualisasikan dalam bentuk histogram, waktu yang diperlukan untuk proses klasifikasi, dan menghasilkan nilai *silhouette coefficient*.

Berikut ini pada Tabel 2 dan Gambar 7, disajikan hasil perbandingan nilai *silhouette coefficient* dan waktu pemrosesan *clustering*.

Tabel 2. Perbandingan Silhouette Coefficient dan Waktu Proses K-Means

No	Jenis Distance	Silhouette Coefficient	Waktu pemrosesan
1	Euclidean Distance	0,64018	0,0789935 dtk
2	City Block Distance	0,46524	0,0878279 dtk
3	Cosine Distance	0,59604	0,187819 dtk



Gambar 7. Perbandingan Silhouette Coefficient dan Waktu Proses K-Means

4. Kesimpulan

Berdasarkan hasil analisa dan pengujian terhadap sistem pengelompokan musik terhadap *mood* menggunakan algoritma *K-Means* dengan membandingkan hasil dari 3 metode *distance measure* dapat disimpulkan bahwa :

1. Telah dibangun sistem untuk mengelompokkan musik terhadap suasana hati menggunakan algoritma *K-Means*.
2. Hasil *distance measure* paling optimal digunakan dalam pengelompokan musik terhadap *mood* adalah *Euclidean Distance* yaitu 0,64018 (tergolong *medium structure*) dan juga menghasilkan waktu pemrosesan *clustering* tercepat yaitu 0,0789935 detik.

Daftar Pustaka

- [1] Song, Y. *et al.* Evaluation of Musical Features for Emotion Classification. *Proceedings of the 13th International Society for Music Information Retrieval Conference*. Porto, Portugal. 2012
- [2] Vallabha Hampiholi. *A method for Music Classification based on Perceived Mood Detection for Indian Bollywood Music*. World Academy of Science, Engineering And Technology Vol : 6. 2012
- [3] Braja Gopal Patra, Dipankar Das. *Automatic Music Mood Classification of Hindi Songs*. Proceedings of the 3rd Workshop on Sentiment Analysis where AI meets Psychology (SAAIP) IJCNLP pages 24–28, Nagoya, Japan. 2013
- [4] Mudiana Binti Mokhsim, *et al.* *Automatic Music Emotion Classification Using Artificial Neural Network Based on Vocal and Instrumental Sound Timbre*. Journal of Computer Science 10(12) : 2584-2592. 2014
- [5] Setiawan, Arif. *Analisis Klasifikasi Suara Berdasarkan Gender dengan Format WAV Menggunakan Algoritma K-Means*. Lembaga Penelitian Universitas Muria Kudus. 2009
- [6] Kaufman L., and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Hoboken, NJ : John Wiley & Sons, Inc. 1990
- [7] Samira Pouyanfar, Hossein Sameti. *Music Emotion Recognition Using Two Level Classification*. International Conference on Intelligent System (ICIS). 2014
- [8] Thayer. *The biopsychology of mood and arousal*. Oxford University Press. 1989.
- [9] Seungwon, Oh., Minsoo Hahn, Jinsul Kim. *Music Mood Classification Using Intro and Refrain Parts of Lyrics* (ICISA). 2013
- [10] Reonaldo Y. S. *Simulasi Sistem Pengacak Sinyal dengan Metode FFT (Fast Fourier Transform)*. E-Journal Teknok Elektro dan Komputer ISSN 2301-8402. 2014.
- [11] Ricky Aurelius N.D., IKG Darma Putra, NMAE Dewi W. *Majalah Ilmiah Teknologi Elektro*. Vol. 13 No. 2, pages : 36-39. 2014
- [12] Lerch, Alexander. *An introduction to Audio Content Analysis – Applications in Signal Processing and Music Informatics*. IEEE Press. 2012
- [13] Jyoti Bora Dibya, Kumar A.G. 2014. *Effect of Different Distance Measures on the Performance of K-Means Algorithm: An Experimental Study in Matlab*. International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 5 (2) , 2501-2506. 2014

Biodata Penulis

I Gede Harsemadi, memperoleh gelar Sarjana Komputer (S.Kom), Program Studi Sistem Informasi STMIK STIKOM Bali, lulus tahun 2009. Memperoleh gelar Magister Teknik (M.T) pada Program Pasca Sarjana Megister Teknik Elektro Universitas Udayana, lulus tahun 2016. Saat ini sebagai staf dan pengajar bidang multimedia di STMIK STIKOM Bali.