

Klasterisasi Jenis Musik Menggunakan Kombinasi Algoritma Neural Network, K-Means dan Particle Swarm Optimization

Alhaji Sheku Sankoh¹, Ahmad Reza Musthafa², Muhammad Imron Rosadi³, Agus Zainal Arifin⁴

Program Pasca Sarjana Teknik Informatika, Institut Teknologi Sepuluh Nopember

Kampus ITS Keputih, Sukolilo, Surabaya 60111, Jawa Timur

E-mail: ¹alhajisheku82@hotmail.com, ²rezaaltraz@yahoo.com, ³imron_uyp@yahoo.com, ⁴agusza@cs.its.ac.id

Masuk: 16 Desember 2014; Direvisi: 16 Januari 2015; Diterima: 26 Januari 2015

Abstract. *Having a number of audio files in a directory could result to unstructured arrangement of files. This will cause some difficulties for users in sorting a collection of audio files based on a particular category of music. In some previous studies, researchers used a method conducting to group documents on a web page. However, those studies were not carried out on file containing documents such as audio files; relatively they were conducted on files that contain text documents. In this study, we develop a method of grouping files using a combination of pre-processing approach, neural networks, k-means, and particle swarm optimization to obtain a form of audio file collections that are group based on the types of music. The result of this study is a system with improved method of grouping audio files based on the type of music. The pre-processing stage has therefore produced the best results on this approach based on spectrum analysis melody and bass guitar, which offers a value precision 95%, 100% recall and an F-Measure 97.44%.*

Keywords: Cluster, Music, NN, K-Means, PSO

Abstrak. *Banyaknya file audio pada suatu direktori membuat susunan file tidak terstruktur. Hal ini akan menyulitkan pengguna untuk mengurutkan bahkan memilah kumpulan file audio berdasarkan kategori tertentu, khususnya kategori berdasarkan jenis musik. Pada penelitian sebelumnya, dilakukan pengelompokan dokumen pada suatu halaman website. Namun hal tersebut tidak dilakukan pada file selain dokumen, seperti file audio. Penelitian ini bertujuan untuk mengembangkan metode pengelompokan file berupa kombinasi pendekatan pre-processing, neural network, k-means, dan particle swarm optimization dengan masukan berupa file audio sehingga diperoleh keluaran berupa kumpulan file audio yang telah terkelompok berdasarkan jenis musik. Hasil dari penelitian ini yaitu berupa suatu sistem dengan pengembangan metode dalam pengelompokan file audio berdasarkan jenis musik. Metode pada tahap pre-processing memiliki hasil terbaik pada pendekatan berdasarkan analisa spectrum melodi gitar dan bass, di mana memiliki nilai precission 95%, recall 100% dan F-Measure 97,44%.*

Kata kunci: Klaster, Musik, NN, K-Means, PSO

1. Pendahuluan

Secara umum sebagai salah satu bagian dari seni, musik merupakan suatu kumpulan bunyi atau nada dari suara vokal atau instrumen maupun gabungan keduanya yang disusun sehingga memiliki ritme tertentu yang mampu menghasilkan keindahan dari segi bentuk, harmoni, dan mengandung ide atau nilai perasaan emosi tertentu. Tidak semua suara atau bunyi dapat dikategorikan sebagai musik. Contoh suara yang tidak dapat dikategorikan sebagai musik adalah suara klakson kendaraan di jalanan, suara gesekan daun pepohonan, suara desiran ombak, dan suara tidak beraturan lainnya (Oxford University Press, 2014). Perkembangan musik di jaman modern ini sangatlah pesat. Ada ribuan musik yang diciptakan setiap harinya oleh berbagai pencipta musik untuk memenuhi kebutuhan para penikmat musik.

Efisiensi dan kecerdasan pencarian informasi musik menjadi topik yang sangat menarik sejak abad ke-21. Dua masalah fundamental dalam berurusan dengan data musik adalah klasifikasi dan klasterisasi (Li, dkk.2009: 477). Klasterisasi data musik adalah mengelompokkan data musik berekstensi MP3 berdasarkan karakteristik yang terkandung dalam data musik dengan tujuan untuk membagi berbagai data musik berdasarkan karakteristik yang serupa atau mirip. Pada saat ini klasifikasi musik menurut *genre*, seperti *Alternative*, *Christian/Gospel*, *Country/Folk*, *Dance/Electronic*, *Eclectic*, *Indie*, *Jazz Blues*, *Pop*, *Rock*, atau *World* (Meyers, 2004: 16). Kalau di Indonesia jenis musik yang termasuk populer yaitu dangdut.

Penelitian klasterisasi data musik telah dilakukan oleh beberapa peneliti, misalnya penelitian yang dilakukan oleh Yang, dkk (2009: 876). Pada penelitian tersebut dilakukan klasterisasi data musik menggunakan metode *K-means*, *spectral*, dan *affinity propagation*. Dengan membandingkan ketiga metode tersebut, diperoleh nilai akurasi berturut-turut 83,2%, 85,4%, dan 82,1%. Penelitian lain yang dilakukan dengan metode yang berbeda menggunakan metode KPSO, yaitu suatu algoritma optimasi evolusi baru hibrida didasarkan pada penggabungan *K-means* dan algoritma PSO. Melakukan uji coba dengan membandingkan metode *K-means*, PSO, PSOK, KPSO, KPSOK untuk klasterisasi dokumen *web*. Hasilnya kombinasi *K-means* dan PSO lebih unggul daripada *K-means* dan PSO secara individu (Jaganathan and Jaiganesh, 2013: 772).

Seleksi fitur merupakan sebuah tahapan penting dalam proses klasifikasi, karena fitur yang terseleksi sangat mempengaruhi tingkat akurasi dari klasifikasi. Pada *dataset* yang memiliki banyak fitur membutuhkan proses untuk mereduksi fitur sebanyak mungkin. (Saputra, 2011:1) mengusulkan sebuah metode seleksi fitur menggunakan *Ensemble Random Forest* dan *Neural Network*.

Berdasarkan hasil klasterisasi data menggunakan berbagai metode berbeda, maka pada penelitian ini peneliti melakukan klasterisasi data musik dengan jumlah klaster yang optimal menggunakan kombinasi algoritma *Neural Network*, *K-means* dan PSO. Diharapkan dengan penelitian ini, dapat mengklasterisasi berbagai macam musik dengan baik dan mendapatkan kelas-kelas yang tepat sesuai dengan fitur yang dimilikinya. Setelah terbagi menjadi beberapa kelas yang sesuai, akan ditetapkan labelisasi terhadap kelas tersebut. Dari hasil labelisasi ini, pengguna dapat mengetahui jenis musik yang terkandung dalam kelas-kelas musik yang telah terbagi.

2. Pendekatan dan Tinjauan Pustaka

2.1. Pemilihan File Audio

Musik terdiri dari enam elemen yaitu modus, tempo, *pitch (register)*, irama, harmoni dan melodi (Meyers, 2004: 24). Pada penelitian ini, dilakukan pengambilan fitur dengan memanfaatkan spektrum melodi gitar dan bass dari setiap *file* audio. Melodi gitar dan bass memiliki *range* frekuensi antara 80-1000 Hz. Setiap *file* audio memiliki rata-rata durasi antara tiga sampai empat menit. Format audio yang digunakan pada penelitian ini yaitu berformat MP3 yang memiliki *sampling rate* 44100 Hz. Dengan besarnya nilai *sampling rate* tersebut, maka dibutuhkan proses untuk menyederhanakan data menjadi fitur yang lebih spesifik. Hal tersebut dapat dilakukan pada tahap-tahap selanjutnya, yaitu pada tahap praproses dan ekstraksi fitur.

2.2. Praproses Sinyal Audio

Praproses sinyal suara digunakan untuk meningkatkan kualitas sinyal suara. Praproses sinyal suara merupakan langkah penting untuk membentuk sistem pengenalan suara atau identifikasi pembicara yang handal dan efisien. Langkah praproses yang digunakan pada penelitian ini yaitu *preemphasis* dan *endpoint* (Keerio, dkk., 2009: 216). Fungsi dari *preemphasis* sama dengan *high-pass filter* yaitu untuk menekan komponen frekuensi tinggi dari sinyal suara. Misalkan $s(n)$ adalah sinyal suara maka, *preemphasis* dapat dinyatakan dengan Persamaan 1, dimana $y(n)$ adalah sinyal *preemphasis* dan α adalah konstanta yang bernilai 0,94.

$$y(n) = s(n) - \alpha \cdot s(n - 1), \quad (1)$$

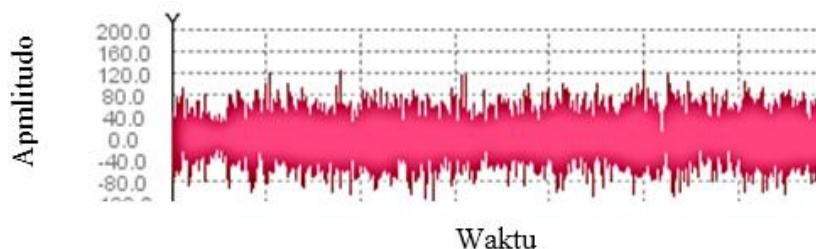
Deteksi *endpoint* digunakan untuk membedakan antara bagian *speech* dan *nonspeech* di sinyal suara. Deteksi *endpoint* yang digunakan pada penelitian ini berdasarkan *short-time energy*. *Short-time energy* menggambarkan kuantitas sinyal suara yang dinyatakan dengan kuadrat dari amplitudo. *Short-time energy* dinyatakan dengan Persamaan 2, dimana E_k adalah energi pada frame ke- k , N adalah jumlah sampel dalam satu *frame*, $w(m)$ adalah fungsi *window*, dan $x_k(m)$ adalah sinyal suara.

$$E_k = \sum_{m=0}^{N-1} x_k^2(m) = \sum_{m=0}^{N-1} [w(m)x(k+m)]^2, \quad 0 \leq m \leq N-1, \quad (2)$$

2.3. Pemrosesan Sinyal Audio dan Ekstraksi Fitur

Pemrosesan sinyal audio dan ekstraksi fitur digunakan untuk mengubah sinyal audio menjadi beberapa tipe representasi *parametric*, sehingga memudahkan pada proses selanjutnya. Hal ini sering dikaitkan dengan pengolahan sinyal. Ada banyak cara untuk merepresentasikan sinyal audio secara parametris sehingga dapat diproses lebih lanjut, antara lain dengan menggunakan *Linier Predictive Coding*, *Mel Frequency Cepstrum Coefficients* (MFCC) dan masih ada banyak lagi (Gupta, dkk., 2013: 102).

MFCC didasarkan pada variasi *bandwith* frekuensi dari telinga manusia, atau bagaimana manusia mendengar. *Linear* pada frekuensi rendah, dan logaritmik pada frekuensi tinggi. Teknik ini baik digunakan untuk menangkap karakteristik dari suara manusia. Hal ini diekspresikan dalam skala *mel-frequency*, yang *linear* pada frekuensi di bawah 1000 Hz, dan logaritmik untuk frekuensi di atas 1000 Hz. Gupta, dkk. (2013) melakukan ekstraksi fitur musik dengan langkah-langkah yang akan dijelaskan selanjutnya. Gambar 1 merupakan grafik masukan sinyal audio.



Gambar 1. Grafik Masukan Sinyal Audio (Sumber: Grafik aplikasi pada penelitian ini)

Langkah 1: *Frame Blocking*. Sinyal suara dibagi menjadi blok-blok kecil yang disebut *frame*. Antara *frame* satu dengan yang lain terjadi tumpang tindih (*overlapping*). Tiap *frame* terdiri dari N sampel dan tumpang tindih sebanyak M sampel. Pada penelitian ini panjang *frame* yang digunakan adalah ~30 milidetik atau $N = 256$ sampel. Sedangkan panjang tumpang tindih adalah ~10 milidetik atau $M = 100$ sampel. Tujuan *frame blocking* adalah membagi sinyal suara menjadi *frame* kecil yang cukup sampel untuk mendapatkan cukup informasi. Jika ukuran *frame* terlalu kecil, informasi yang didapat tidak cukup untuk dipercaya. Sebaliknya jika terlalu besar, informasi di dalam *frame* sering berubah-ubah.

Langkah 2: *Windowing*. Proses *windowing* dilakukan untuk tiap *frame*. *Windowing* digunakan untuk meminimalkan ketidaksinambungan pada awal dan akhir *frame*. Dengan menggunakan *window*, sinyal suara pada awal dan akhir *frame* menjadi runcing. Hasil dari perhitungan *windowing* ditunjukkan pada Persamaan 3, dimana $w(m)$ adalah fungsi *window* dan $x(m)$ adalah sinyal suara. Pada penelitian ini fungsi *window* yang digunakan adalah *hamming window* yang dinyatakan dengan Persamaan 4.

$$y(m) = w(m)x(m), \quad 0 \leq m \leq N - 1, \quad (3)$$

$$w(m) = 0.54 - 0.46 * \cos\left(\frac{2\pi m}{N-1}\right), \quad 0 \leq m \leq N - 1, \quad (4)$$

Langkah 3: Fast Fourier Transform (FFT). Fast Fourier Transform (FFT) merubah sinyal suara dari domain waktu ke domain frekuensi. FFT adalah algoritma tercepat yang mengimplementasikan *Discrete Fourier Transform* (DFT). DFT dinyatakan dengan Persamaan 5, dimana j adalah bilangan imajiner yang bernilai $\sqrt{-1}$ dan $x(m)$ adalah sinyal suara. Hasil dari proses ini disebut sebagai spektrum.

$$X(k) = \sum_{m=0}^{N-1} x(m)e^{-j2\pi km/N}, \quad k = 0,1,2, \dots, N-1, \quad (5)$$

Langkah 4: Mel-frequency Filtering. Besaran frekuensi hasil FFT dipetakan menggunakan *mel-scale* yang dinyatakan dengan Persamaan 6, dimana f adalah frekuensi maksimum. Mel frekuensi kemudian dibagi menjadi L filter. Pada penelitian ini jumlah filter yang digunakan adalah 26 filter.

$$mel(f) = 2595 * \log\left(1 + \frac{f}{700}\right), \quad (6)$$

Langkah 5: Discrete Cosine Transform (DCT). Setelah didapatkan spektrum mel, dilakukan operasi DCT. DCT selain untuk mengembalikan ke domain waktu juga untuk kompresi spektrum. Operasi DCT dapat dinyatakan dengan Persamaan 7, dimana L adalah jumlah mel spektrum filter, $D(m)$ adalah mel spektrum, dan N adalah jumlah koefisien cepstrum. Pada penelitian ini jumlah koefisien cepstrum yang digunakan adalah 13 dan untuk komponen pertama, $C(0)$, tidak dipakai karena mengandung sedikit informasi. Jadi jumlah koefisien cepstrum yang akan dipakai adalah 12 cepstrum.

$$C(k) = \sum_{m=1}^L (\log D(m)) \cos\left[k\left(m - \frac{1}{2}\right)\frac{\pi}{L}\right], \quad k = 0,1,2, \dots, N-1, \quad (7)$$

Langkah 6: Delta & Delta-delta Cepstrum. Pendengaran manusia lebih sensitif terhadap karakteristik dinamik sinyal suara. Turunan pertama (*delta*) dan turunan kedua (*delta-delta*) menggambarkan karakteristik dinamik dari sinyal suara. Nilai *delta cepstrum* didapatkan dari turunan pertama koefisien cepstrum yang dinyatakan dengan Persamaan 8, dimana $C(k)$ adalah koefisien cepstrum dan T bernilai konstan. Pada penelitian ini konstanta T bernilai dua. Sedangkan *delta-delta cepstrum* didapatkan dari turunan pertama *delta cepstrum*.

$$d(k) = \frac{\sum_{t=1}^T t(C(k+t) - C(k-t))}{2 \sum_{t=1}^T t^2}, \quad (8)$$

Langkah 7: ZCR. ZCR: Sebuah *zero-crossing* dikatakan terjadi jika sampel berturut-turut memiliki tanda-tanda yang berbeda. *Zero-crossing Ratio* adalah jumlah *time-domain zero-crossings* dan jumlah sampel dalam *frame* (Wang, dkk., 2011: 392). *Zero-crossing Rate* (ZCR) adalah fitur lain akustik dasar yang dapat dihitung dengan mudah. Ini adalah tingkat tanda perubahan sepanjang sinyal, tingkat dimana perubahan sinyal dari positif ke negatif dan kembali. Fitur ini telah digunakan di kedua pengenalan suara dan musik pencarian informasi (Wang, dkk., 2011). Pada Persamaan 9 ZCR adalah sinyal diskrit *sampling*, N berarti number poin *sampling*.

$$ZCR = \frac{1}{2N-1} \sum_{n=1}^{N-1} \text{sgn}[x(n+1) - \text{sgn}(n)] \quad (9)$$

2.4. Klasterisasi

Klasterisasi adalah mengelompokkan obyek-obyek berdasarkan karakteristik yang dimilikinya. Tujuan klasterisasi adalah untuk membuat benda-benda dalam suatu kelompok menjadi serupa atau berhubungan satu sama lain dan berbeda dengan benda-benda di kelompok lainnya. Klaster dapat mengklasifikasi obyek sehingga setiap obyek yang paling dekat kesamaannya dengan obyek lain berada dalam klaster yang sama.

Sebelum melakukan klasterisasi, peneliti melakukan pemilihan fitur terbaik dengan menggunakan algoritma NN. Setelah mendapatkan fitur terbaik dilakukan klasterisasi menggunakan algoritma *K-means*. Untuk mengoptimalkan hasil klaster pada data musik, peneliti menggunakan PSO.

2.4.1. Neural Network

Proses pemilihan fitur sangat mempengaruhi hasil dari proses klasterisasi. Metode yang digunakan untuk seleksi fitur pada penelitian ini adalah *Neural Network*. *Neural Network* merupakan sebuah mesin klasifikasi yang dimodelkan meniru dari struktur biologi pada saraf manusia. Pada penelitian ini *neural network* yang digunakan termasuk dalam jenis MLP (*Multi Layer Perceptron*) dimana terdapat *hidden layer* sebelum *output* dari *perceptron* di proses. MLP juga dikenal sebagai FFNN (*Feed-Forward Neural Network*). Secara umum model ini bekerja dengan menerima suatu vektor input *I* dan kemudian menghitung suatu respon atau *output O* dengan memproses (*propagating*) *I* melalui beberapa elemen-elemen proses yang saling terkait (Suhartono, 2007: 44).

Pada FFNN elemen-elemen proses tersusun dalam beberapa lapis (*layer*) dan data input mengalir dari satu lapis ke lapis berikutnya secara berurutan. Pada tiap lapisan *input* data ditransformasikan ke dalam lapis berikutnya secara *nonlinear* oleh elemen-elemen proses dan kemudian diproses ke lapis berikutnya. Akhirnya nilai *output O* yang didapat berupa nilai *scalar* atau vektor, yang dihitung pada lapisan *output* (Suhartono, 2007: 45).

Untuk seleksi fitur *Neural Network* akan digunakan sebagai mesin klasifikasi yang kemudian akan di hitung selisih nilai *error* yang dihasilkan. Dapat dikatakan bahwa jika sebuah fitur ketika dihapus membuat nilai *error* semakin bertambah besar maka fitur tersebut merupakan fitur yang baik dan direkomendasikan untuk digunakan, dan sebaliknya (Suhartono, 2007: 45).

2.4.2. K-means

K-means merupakan salah satu metode *clustering* non hirarki yang berusaha mempartisi data yang ada ke dalam bentuk satu atau lebih *cluster*. Metode ini mempartisi data ke dalam *cluster* sehingga data yang memiliki karakteristik yang sama dikelompokkan ke dalam satu *cluster* yang sama dan data yang mempunyai karakteristik yang berbeda di kelompokkan ke dalam *cluster* yang lain. Secara umum algoritma dasar dari *K-means Clustering* adalah sebagai berikut.

Langkah 0: Tentukan jumlah *cluster*. **Langkah 1:** Alokasikan data ke dalam *cluster* secara *random*. **Langkah 2:** Hitung *centroid*/rata-rata dari data yang ada di masing-masing *cluster*. Untuk menghitung jarak antara *centroid* dengan data dapat menggunakan *Euclidean Distance*. Persamaan 10 adalah persamaan dari *Euclidean Distance*. **Langkah 3:** Alokasikan masing-masing data ke *centroid*/rata-rata terdekat. **Langkah 4:** Kembali ke langkah 2, apabila nilai *centroid* belum konvergen.

$$d_{ij} = \sqrt{\sum_{k=1}^P \{x_{ik} - x_{jk}\}^2} \quad (10)$$

dimana: d_{ij} = Jarak objek antara objek *i* dan *j*; *P* = Dimensi data; X_{ik} = Koordinat dari obyek *i*

pada dimensi k ; X_{jk} = Koordinat dari obyek j pada dimensi k .

2.4.3. Particle Swarm Optimization

Particle Swarm Optimization atau biasa disingkat PSO merupakan sebuah algoritma optimasi stokastik. Sejak pertama kali ditemukan, algoritma PSO telah digunakan secara luas untuk memecahkan masalah optimasi (Niknam dan Amiri, 2010: 184). Model matematika untuk PSO dapat dilihat pada persamaan 11.

$$V_i^{(t+1)} = \omega \cdot V_i^{(t)} + c_1 \cdot \text{rand}_1(\cdot) \cdot (P_{best\ i} - X_i^{(t)}) + c_2 \cdot \text{rand}_2(\cdot) \cdot (G_{best} - X_i^{(t)})$$

$$X_i^{(t+1)} = X_i^{(t)} + V_i^{(t+1)}; i = 1, 2, 3, \dots, N_{swarm} \quad (11)$$

Keterangan: i = Indeks dari setiap partikel, t = Jumlah iterasi atau pengulangan, $\text{rand}_1(\cdot)$ dan $\text{rand}_2(\cdot)$ = Nomor acak antara 0 dan 1, $P_{best\ i}$ = Pengalaman terbaik dari partikel ke- i yang tercatat, G_{best} = Partikel terbaik di antara seluruh populasi, N_{swarm} = Jumlah kawanan, Konstanta c_1 dan c_2 = Faktor bobot dari istilah percepatan stokastik yang menarik setiap partikel terhadap posisi $P_{best\ i}$ dan G_{best} , t_{max} = Jumlah maksimum iterasi atau pengulangan, ω = Berat inersia, ω_{max} = Berat inersia maksimum, ω_{min} = Berat inersia minimum, K = Jumlah variabel.

Sebagaimana ditunjukkan dalam Persamaan 11, terdapat tiga parameter penyetelan (*tuning parameter*), yaitu ω , c_1 , dan c_2 yang masing-masing memilikidampak yang besar pada kinerja algoritma. Berat inersia mengontrol sifat eksplorasi algoritma. *Learning factor* c_1 dan c_2 masing-masing menentukan dampak dari $P_{best\ i}$ terbaik dan G_{best} terbaik. Jika $c_1 > c_2$, maka partikel memiliki kecenderungan untuk berkumpul ke posisi terbaik yang ditentukan oleh dirinya sendiri ($P_{best\ i}$) dibandingkan posisi terbaik yang ditemukan oleh populasi (G_{best}), dan sebaliknya. Kebanyakan implementasi menggunakan pengaturan dengan $c_1 = c_2 = 2$.

2.5. Eksperimen dan Evaluasi

Eksperimen dilakukan sebagai uji coba pada penelitian ini dan evaluasi dilakukan sebagai pengukuran keberhasilan dalam penelitian ini.

2.5.1. Eksperimen

Pada bagian ini, kami menerapkan pendekatan yang diusulkan untuk pengelompokan audio. Pertama-tama, kita harus membangun sebuah *audio library* yang terdiri dari musik *pop*, musik dangdut, dan musik *rock*. Data audio yang digunakan dalam percobaan adalah *stereo*, *encoding* dari 64 bit, dan *sampling rate* 44100Hz. Dalam proses penelitian, kami secara acak memilih 68 fragmen masing-masing kelas sebagai set pelatihan, dan sisanya digunakan sebagai set tes.

2.5.2. Evaluasi

Evaluasi kluster digunakan untuk mengetahui kluster yang akurat dengan menggunakan *cohesion*, *separation* dan pengukuran menggunakan *F-Measure*.

2.5.2.1. Cohesion dan Separation

K-means merupakan algoritma *clustering* yang bersifat *partitional* yaitu membagi himpunan objek data ke dalam sub himpunan (*cluster*) yang tidak *overlap*, sehingga setiap objek data berada tepat dalam satu *cluster*. Strategi *partitional-clustering* yang paling sering digunakan adalah berdasarkan kriteria *square error*. Secara umum, tujuan kriteria *square error* adalah untuk memperoleh partisi (jumlah *cluster* tetap) yang meminimalkan total *square error* (Tan, 2007:496).

SSE (Sum Squared of Error) menyatakan total kesalahan kuadrat yang terjadi bila n data x_1, \dots, x_n dikelompokkan ke dalam k *cluster* dengan pusat tiap *cluster* adalah μ_1, \dots, μ_k . Nilai *SSE* tergantung pada jumlah *cluster* dan bagaimana data dikelompokkan ke dalam *cluster-cluster* tersebut. Semakin kecil nilai *SSE*, semakin bagus hasil *clustering*-nya (Tan, 2007: 539).

Adapun rumus *SSE* dapat dilihat pada Persamaan 12.

$$SSE = \sum_{I=0}^K \sum_{x \in C_i} ||x - m_i|| \tag{12}$$

Keterangan: k = Jumlah Klaster, C_i = Set Data C ke- i , x = Nilai set Data, m_i = *Mean* (rata-rata) dari setiap klaster.

Besaran *separation* antar klaster adalah *SSB* (*Sum of Square Between-group*), jumlah kuadrat jarak dari sebuah klaster *centroid* m_i terhadap setiap *cluster centroid* lainnya x . Dengan menjumlahkan *SSB* dari setiap klaster, diperoleh total *SSB* (Persamaan 13) (Tan, 2007:540).

$$SSB = \sum_{I=0}^K \sum_{x \in C_i} n_i ||x - m_i|| \tag{13}$$

Keterangan: k = Jumlah klaster, n_i = Ukuran sampel dari data ke- I , x = Nilai set data, m_i = *Mean* (rata-rata) dari setiap klaster.

Semakin tinggi nilai total *SSB* maka semakin jauh jarak suatu *cluster* terhadap *cluster* yang lainnya. Dalam beberapa kasus, terdapat relasi yang kuat antara *cohesion* dan *separation*. Secara spesifik, jumlah dari total *SSE* dan total *SSB* adalah konstan yaitu sama dengan *total sum of squares* (*TSS*) (Persamaan 14) sehingga meminimalkan *SSE* ekuivalen dengan memaksimalkan *SSB* (Tan, 2007:540).

$$TSS = SSE + SSB \tag{14}$$

2.5.2.2. *F-Measure*

F-Measure adalah nilai yang didapatkan dari pengukuran *precision* dan *recall* antara kelas hasil *cluster* dengan kelas sebenarnya yang terdapat pada data masukan (Liu, 2007:73-74). Tabel 1 adalah tabel kebenaran dari kondisi aktual terhadap *classified*.

Tabel 1. Tabel Kebenaran Kondisi Actual Terhadap Classified

	Classified Positive	Classified Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

Keterangan:

TP : Jumlah klasifikasi yang benar dari contoh positif (*true positive*).

FN : Jumlah okeklasifikasi yang salah contoh positif (*false negative*).

FP : Jumlah klasifikasi yang salah contoh negatif (*false positive*).

TN : Jumlah klasifikasi yang benar dari contoh negatif (*true negative*).

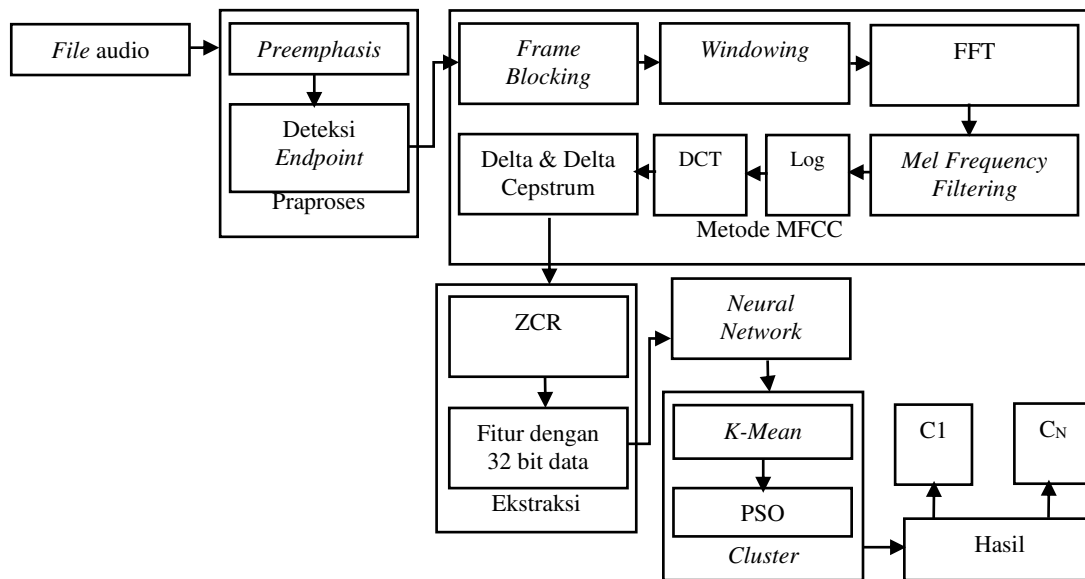
Precision dan *Recall* bisa didapatkan dengan Persamaan 15. Berdasarkan *confusion matrix*, *Precision* (p): jumlah sampel berkategori positif diklasifikasi benar dibagi dengan total sampel yang diklasifikasi sebagai sample positif. *Recall* (r): jumlah sampel diklasifikasi positif dibagi total sampel dalam *testing set* berkategori positif. *F-Measure* ($F1$) adalah *harmonic mean* dari *precision* dan *recall* (Persamaan 16). Semakin besar nilai *F-Measure* semakin baik kualitas *cluster* tersebut.

$$p = \frac{TP}{TP + FP}, \quad r = \frac{TP}{TP + FN} \tag{15}$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{16}$$

3. Metodologi

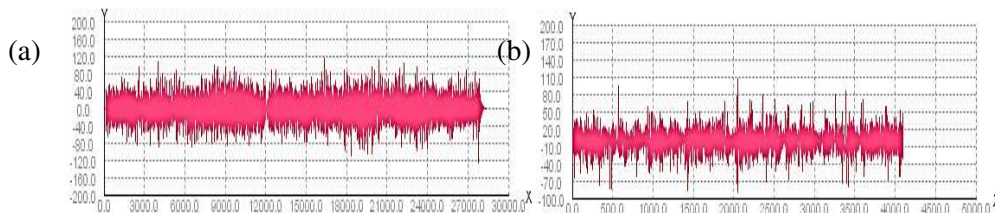
Proses klasterisasi jenis musik pada penelitian ini memiliki beberapa tahap yang harus dilakukan, seperti ditunjukkan pada Gambar 2. Tahap pertama yaitu dilakukan praproses dengan menggunakan *preemphasis* dan deteksi *endpoint*. Tahap kedua dilakukan pemrosesan sinyal audio dengan menggunakan metode MFCC. Tahap ketiga dilakukan ekstraksi fitur menggunakan ZCR. Tahap keempat dilakukan pemilihan fitur terbaik dengan menggunakan algoritma NN. Setelah mendapatkan fitur terbaik, dilakukan klasterisasi menggunakan algoritma *K-means* dan dioptimalisasi menggunakan PSO. Setelah itu dilakukan labelisasi untuk mengetahui jenis musik dari setiap *file* audio.



Gambar 2. Desain Sistem Kluster Dokumen Musik

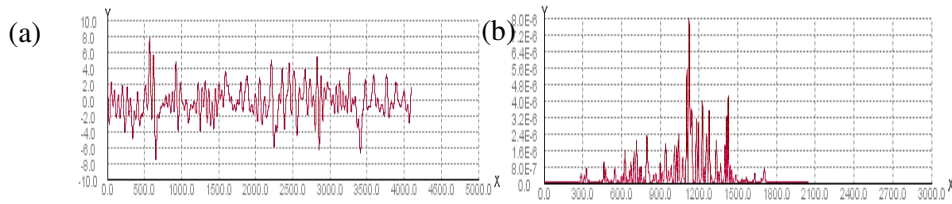
4. Hasil dan Pembahasan

Pada tahap praproses, pemrosesan sinyal audio, dan ekstraksi fitur pada penelitian ini ditunjukkan pada ilustrasi Gambar 3. Sinyal audio akan di-*capture* menggunakan audio *streamreader* dari aplikasi yang dikembangkan pada penelitian ini seperti ditunjukkan pada Gambar 3(a). Setelah itu sinyal audio akan dilakukan *sampling* dalam jarak dan durasi satu menit seperti ditunjukkan pada Gambar 3(b). Setelah itu, dengan memanfaatkan metode MFCC dilakukan pemrosesan sinyal audio berupa *filtering* dan penguatan spektrum sinyal audio seperti ditunjukkan pada Gambar 4(a) dan Gambar 4(b). Setelah itu, dilakukan ekstraksi fitur dengan memanfaatkan ZCR sehingga panjang fitur menjadi 32 bit data, seperti ditunjukkan pada Gambar 5.

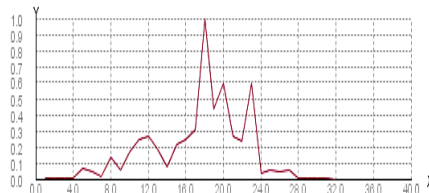


Gambar 3 (a) Sinyal Audio yang Dibaca (b) *Sampling* Sinyal Audio Dalam Jarak dan Durasi 1 Menit

(Sumber: Grafik aplikasi pada penelitian ini)

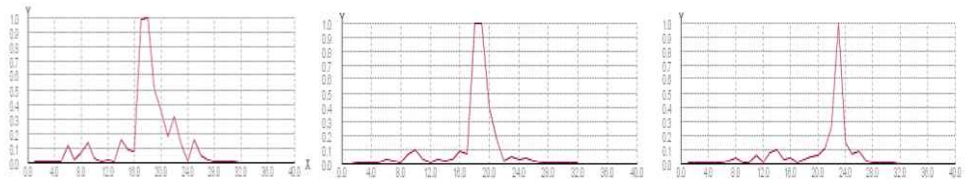


Gambar 4 (a) Sinyal *Sampling* Audio yang Telah Di-filter (b) Penguatan Spektrum Sinyal Audio (Sumber: Grafik aplikasi pada penelitian ini)

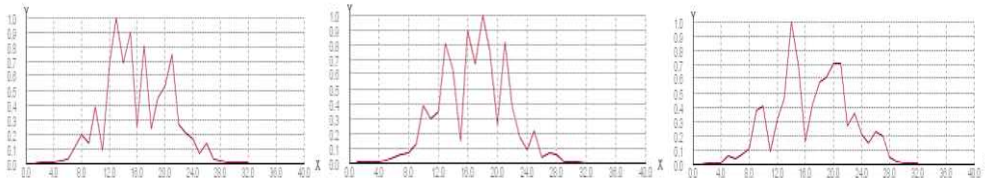


Gambar 5. Fitur Audio yang Dihasilkan, Dengan Panjang Fitur 32 Bit Data. (Sumber: Grafik aplikasi pada penelitian ini)

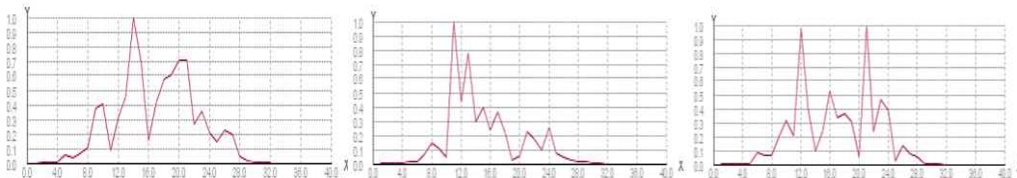
Berdasarkan analisa spektrum melodi dan bass dari jenis musik, diperoleh fitur-fitur yang dominan dari musik tersebut. Fitur-fitur yang dominan dari musik *pop*, *rock*, dan dangdut ditunjukkan pada Gambar 6, Gambar 7, dan Gambar 8.



Gambar 6. Karakteristik Fitur Musik Jenis Pop (Sumber: Grafik aplikasi pada penelitian ini)



Gambar 7. Karakteristik Fitur Musik Jenis Rock (Sumber: Grafik aplikasi pada penelitian ini)



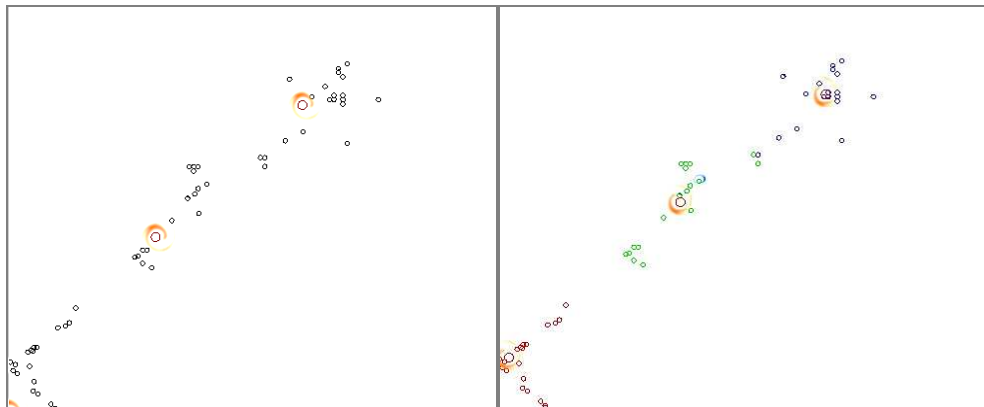
Gambar 8. Karakteristik Fitur Musik Jenis Dangdut (Sumber: Grafik aplikasi pada penelitian ini)

Fitur musik jenis *pop* memiliki karakteristik dengan spektrum yang berbentuk satu puncak yang tinggi dan puncak lain yang tidak dominan. Fitur musik jenis *rock* memiliki spektrum yang terdiri dari beberapa puncak yang tinggi. Fitur dangdut memiliki fitur yang hampir sama dengan fitur *pop*, namun memiliki rata-rata spektrum yang lebih rendah dari *rock*. Berdasarkan fitur yang diperoleh dari setiap *file* audio, selanjutnya akan dilakukan pemrosesan menggunakan algoritma NN. Fitur-fitur tersebut akan digunakan sebagai masukan NN.

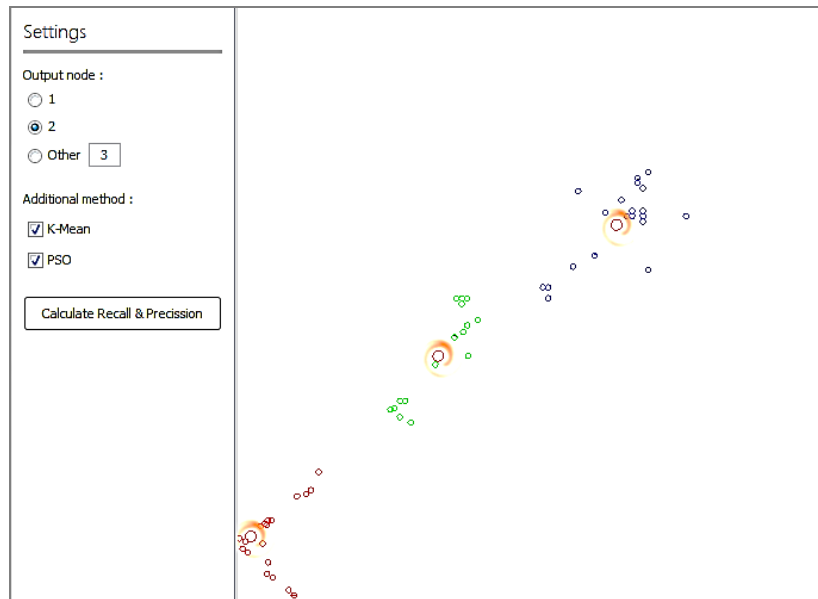
Arsitektur NN yang digunakan terdiri dari 32 titik masukan, 64 titik *hidden*, dua titik keluaran, dan tiga target keluaran. Pemilihan data untuk *training* NN dilakukan dengan melihat kemiripan terhadap karakteristik bentuk spektrum. Kemiripan tersebut dapat dicari dengan memanfaatkan prosentase *error* antara target keluaran dengan hasil keluaran. Nilai *error* dari suatu fitur yang melebihi 20% tidak digunakan sebagai data *training*. Fitur yang sangat berbeda dari karakteristik musik tidak digunakan pada proses *training* agar tidak merusak data serta sebagai alasan untuk mempercepat proses komputasi.

Algoritma NN yang digunakan pada penelitian ini yaitu menggunakan algoritma *backpropagation*, dikarenakan terdapat fase *backward training* dan memiliki hasil yang akurat. Pada tahap pengujian, data yang digunakan berupa *file* audio secara acak pada suatu direktori. Data tersebut akan digunakan sebagai masukan NN sehingga akan dihasilkan keluaran seperti ditunjukkan pada Gambar 9(a). Setelah itu, dilakukan klusterisasi dengan memanfaatkan algoritma *K-means* seperti ditunjukkan pada Gambar 9(b).

Hasil klusterisasi antara Gambar 9(b) dan Gambar 10 memiliki perbedaan. Pada hasil klusterisasi menggunakan algoritma *K-means clustering* memiliki kelemahan, yaitu keakuratan yang tidak terjamin seperti terdapat suatu anggota dari suatu *cluster* yang tidak masuk ke dalam *cluster* yang seharusnya walaupun memiliki jarak yang dekat. Dengan menggunakan algoritma PSO, hasil klusterisasi dioptimalisasi sehingga anggota *cluster* tersebut masuk ke dalam *cluster* yang seharusnya. PSO memanfaatkan masukan dari hasil akhir nilai *centroid K-means* pada proses sebelumnya. PSO memiliki algoritma yang hampir samadengan *K-means*, khususnya pada tahap mencari nilai SSE. Namun pada tahap *update* nilai *centroid* memiliki perbedaan, pada PSO hal tersebut memanfaatkan nilai kecepatan (*velocity*) dari partikel yang telah dibangun. Algoritma PSO akan mencari nilai *global optimum* dari setiap *centroid* sehingga diperoleh hasil klusterisasi yang lebih optimal.



Gambar 9 (a) Hasil Keluaran NN (b) Klusterisasi dengan *K-means*
(Sumber: Grafik aplikasi pada penelitian ini)



Gambar 10. Optimalisasi Hasil *K-means* menggunakan PSO
(Sumber: Grafik aplikasi pada penelitian ini)

Tabel 2 menunjukkan perbedaan terhadap hasil klusterisasi musik pada *file* audio dengan menggunakan beberapa pendekatan metode. Data tes yang digunakan pada pengujian tersebut yaitu sebanyak 30 *file* audio pada suatu direktori. Semakin banyak data tes, semakin terlihat perbedaan hasil klusterisasi yang dihasilkan.

Tabel 2. Perbedaan Hasil Klusterisasi

Metode	Presisi (%)	Recall (%)	SSE	SSB	TSS	F-Measure (%)
NN	75	100	5.51232	5.862951	11.37527	85.71
NN+ <i>K-means</i>	90	100	5.51232	5.86295	11.3752	94.73
NN+ <i>K-means</i> +PSO	95	100	6.531133	6.351476	12.88261	97.44

Tabel 3 menunjukkan hasil pengujian kombinasi metode NN, *K-means*, dan PSO pada parameter *cohesion*, *separation*, dan *F-Measure* dengan jumlah kluster yang sama yaitu tiga, yang mempunyai arti bahwa setiap satu parameter kelompok mendapatkan label jenis musik. Hasil uji coba menunjukkan bahwa presisi terbaik di lakukan dengan *dataset* sebanyak 20 yaitu 95% dan untuk *recall* setiap uji coba hasilnya sama yaitu 100% dan *F-Measure* terbaik ditemukan di penggunaan *dataset* sebanyak 20 yaitu 97,44%.

Tabel 3. Nilai Rata-rata Hasil Uji Coba

Percobaan	Jumlah Dataset	Presisi (%)	Recall (%)	SSE	SSB	TSS	F-Measure (%)
1	10	90.00	100	3.046219	2.576869	5.618906	94.74
2	12	91.67	100	4.117232	2.393627	7.053504	95.65
3	14	92.86	100	4.900875	3.675933	8.576809	96.30
4	16	93.75	100	5.193481	4.394548	9.58803	96.77
5	18	94.44	100	5.3586	5.298674	10.65727	97.14
6	20	95.00	100	6.531133	6.351476	12.88261	97.44
7	22	90.91	100	6.577217	7.096729	13.67395	95.24
8	24	91.67	100	7.129259	7.613228	14.74249	95.65
9	26	88.46	100	7.725598	8.193825	15.91942	93.88
10	28	89.29	100	7.789668	8.983771	16.77344	94.34
11	30	90.00	100	7.856915	9.891201	17.74812	94.74

Nilai yang diperoleh tersebut juga dipengaruhi oleh penentuan *threshold* pada NN untuk seleksi fitur pada proses *training*. Semakin kecil prosentase *threshold error* yang digunakan, maka semakin akurat keluaran yang dihasilkan.

5. Kesimpulan dan Saran

Metode yang peneliti usulkan dalam melakukan klusterisasi data musik dengan jumlah kluster yang optimal yaitu dengan menggunakan kombinasi algoritma NN, *K-means*, dan PSO. Hasil penelitian ini diharapkan dapat melakukan klusterisasi berbagai macam jenis musik dengan baik, khususnya dalam mendapatkan kelas yang sesuai dengan fitur yang dimilikinya. Hasil akhir penelitian ini memiliki nilai presisi 95%, *recall* setiap uji coba hasilnya sama yaitu 100%, dan *F-Measure* terbaik ditemukan di penggunaan *dataset* sebanyak 20 yaitu 97,44%.

Penelitian kedepannya akan menggunakan berbagai macam format *file* audio, tidak terbatas pada *file* audio dengan format MP3. Selain itu, juga akan dilakukan penelitian dengan menggunakan pendekatan lain di samping menggunakan pendekatan analisa terhadap *spectrum* melodi dan bass pada tahap pemrosesan sinyal dan ekstraksi fitur.

Referensi

- Gupta, S., Jaafar, J., Ahmad, W.F., dan Bansal, A. 2013. Feature Extraction Using MFCC. *Signal & Image Processing: An International Journal (SIPIJ)*, Vol. 4, No. 4:101-108.
- Jaganathan, P., and Jaiganesh, S. 2013. An improved K-means algorithm combined with Particle Swarm Optimization approach for efficient web document clustering. *IEEE*. 978-1-4673-6126-2/13: 772-776.
- Keerio, A., Mitra, B.K., Birch, P., Young, R., dan Chatwin, C. 2013. On Preprocessing of speech signal. *International Journal of Signal Processing*, Vol. 5, No.3: 216-222.
- Li, Tao. and S. Zhu. 2009. Music Clustering With Features From Different Information Sources. *IEEE Transactions on Multimedia*. VOL. 11, NO. 3, April 2009: 477-485.
- Liu, Bing. 2007. *Web Data Mining*. Chicago: ACM Computing Classification.
- Meyers, C.O. 2004. *A Mood-Based Music Classification and Exploration System*. McGill University: Program in Media Arts and Sciences.
- Niknam, T., and Amiri, B. 2010. An Efficient Hybrid Approach Based on PSO, ACO And K-Means for Cluster Analysis. *Applied Soft Computing*, 10 (2010): 183–197.
- Oxford University Press. 2014. *Definition of music in English*. (<http://oxforddictionaries.com/definition/english/music>, diakses 1 Desember 2015).
- Saputra, Wahyu S. J, Sujatmika Arif Rahman, Arifin Agus Zainal. 2011. Seleksi Fitur Menggunakan Random Forest Dan Neural Network. *Industrial Electronics Seminar*: 93-97.
- Suhartono. 2007. *Feedforward Neural Network Untuk Pemodelan Runtun Waktu*. Distertasi. Yogyakarta: Universitas Gadjah Mada.
- Tan, P-N., Steinbach, M., Kumar, V. 2007. *Cluster Analysis: Basic Concepts and Algorithms*. In: *Introduction to Data Mining*. Boston: Pearson Addison Wesley. P.
- Wang, Yunhui, Yu Xiaoqing. 2011. The research of audio clustering with Gaussian mixture based on em algorithm. *CCWMC*: 389-393.
- Yang, Y.H, Liu, C.C., and Chen, H.H. 2009. *Clustering For Music Search Result*. Taipei: National Taiwan University.