



KLASIFIKASI GENRE LAGU DENGAN FITUR AKUSTIK MENGUNAKAN METODE *K-NEAREST NEIGHBOR*

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Husein Abdulbar

NIM: 155150200111193



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2019

PENGESAHAN

KLASIFIKASI GENRE LAGU DENGAN FITUR AKUSTIK MENGGUNAKAN
METODE K-NEAREST NEIGHBOR

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Husein Abdulbar
NIM: 155150200111193

Skripsi ini telah diuji dan dinyatakan lulus pada
18 Juli 2019

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II



Putra Pandu Adikara, S.Kom, M.Kom
NIP: 19850725 200812 1 002



Sigit Adinugroho, S.Kom., M.Sc
NIK: 2016078807011001

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 18 Juli 2019



Husein Abdulbar

NIM: 155150200111193



PRAKATA

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufik, dan hidayah-Nya sehingga laporan skripsi dengan judul “Klasifikasi Genre Lagu dengan Fitur Akustik Menggunakan Metode *K-Nearest Neighbor*” dapat terselesaikan.

Penulis menyadari bahwa skripsi ini tidak akan berjalan dengan baik tanpa bantuan dari banyak pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terimakasih kepada:

1. Bapak Putra Pandu Adikara, S.Kom, M.Kom dan Bapak Sigit Adinugroho S.Kom., M.Sc selaku Dosen Pembimbing skripsi yang telah memberikan arahan dan semangat kepada penulis dengan penuh kesabaran sehingga skripsi ini dapat terselesaikan.
2. Bapak Agus Wahyu Widodo, S.T, M.Sc selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
4. Ayahanda Choirul Fasih dan Ibunda Nanik Hidayati serta saudara dan saudari penulis yang tanpa dukungan, didikan, dan saran dari mereka skripsi ini tidak akan terselesaikan.
5. Teman-teman POROS yang telah mendukung dan memberikan banyak ilmu yang bermanfaat kepada penulis.
6. Seluruh civitas akademik Teknik Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penyelesaian skripsi ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan , sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 18 Juli 2019

Penulis

huseinabdulbar@gmail.com



ABSTRAK

Husein Abdulbar, Klasifikasi Genre Lagu dengan Fitur Akustik Menggunakan Metode *K-Nearest Neighbor*

Pembimbing: Putra Pandu Adikara, S.Kom, M.Kom dan Sigit Adinugroho, S.Kom., M.Sc

Lagu adalah suatu kegiatan yang tidak bisa lepas dari aktivitas sehari-hari manusia. Ketika mendengarkan lagu manusia dapat lebih fokus pada aktivitas yang sedang dilakukannya. Sebuah lagu minimal memiliki satu genre yang memiliki fungsi untuk mengelompokkan jenis-jenis lagu. Pesatnya perkembangan informasi multimedia dan perangkat elektronik menyebabkan peningkatan dramatis pada apresiasi dan penciptaan musik. Pada suatu sisi peningkatan ini mendorong masyarakat untuk lebih menikmati lagu. Tapi di sisi lain, peningkatan ini memaksa pengembangan-pengembangan teknologi baru untuk kenyamanan mendengarkan lagu. Contohnya adalah bagaimana seseorang ingin mencari lagu berdasarkan lagu yang telah didengarkan. Klasifikasi genre lagu adalah salah satu cara yang dapat memudahkan pengguna untuk mencari lagu yang cocok, dengan klasifikasi pengguna dapat menemukan lagu berdasarkan genre dari lagu yang disukainya.

Penelitian ini menggunakan metode *K-Nearest Neighbor* (K-NN) sebagai teknik pengklasifikasian genre lagu. Untuk mengukur kedekatan dua buah lagu digunakan persamaan *normalized cross correlation* (NCC) yang menggantikan persamaan penghitungan jarak pada metode K-NN. Ini dilakukan karena NCC dapat mengukur kemiripan dua buah sinyal lebih baik dibandingkan persamaan penghitungan jarak pada K-NN. Fitur yang diekstraksi dari sebuah lagu adalah *zero crossing rate*, *spectral centroid*, *spectral rolloff*, dan *energy*. Data yang didapat dari hasil ekstraksi fitur kemudian dinormalisasi menggunakan persamaan *z-score*.

Penelitian ini berhasil mengembangkan sistem yang dapat mengklasifikasi lagu menggunakan empat fitur akustik. Hasil pengujian menunjukkan evaluasi terbaik didapatkan ketika durasi bernilai 10, *offset* bernilai 120, dan K dari K-NN bernilai 10. *Precision*, *recall*, dan *f-measure* yang didapatkan pada penelitian ini adalah *precision* bernilai 0,637, *recall* bernilai 0,633, dan *f-measure* bernilai 0,635.

Kata kunci: genre, lagu, klasifikasi, *K-nearest neighbor*, *cross correlation*



ABSTRACT

Husein Abdulbar, Klasifikasi Genre Lagu dengan Fitur Akustik Menggunakan Metode K-Nearest Neighbor

Supervisors: Putra Pandu Adikara, S.Kom, M.Kom and Sigit Adinugroho, S.Kom., M.Sc

Song cannot be separated from humans daily activities. When listening to songs humans can focus more on their activities. A song has at least one genre that used to classify the types of songs. The rapid development of information on multimedia and electronic devices has led to a dramatic increase in music appreciation and creation. On the one hand this increase encourages people to enjoy songs more. But on the other hand, this increase forced the development of new technologies for the convenience of listening to songs. An example is how someone wants to find a song based on a song that has been heard. Genres classification is one way that can help users to find suitable songs, with that users can find songs by genre of song they like.

This study will use the K-Nearest Neighbor (K-NN) method as a genre classification technique for songs. To measure the similarity of two songs, a normalized cross correlation (NCC) equation is used to replace the distance calculation equation in the K-NN method. This is done because the NCC can measure the similarity of two signals better than the distance calculation equation in the K-NN. The features that extracted from a song are zero crossing rate, spectral centroid, spectral rolloff, and energy. Data obtained from feature extraction will be normalized using the z-score equation.

This research successfully developed a system that can classify songs using four acoustic features. The test results show that the best evaluation is obtained when the duration is 10, the offset is 120, and K in K-NN is 10. Precision, recall, and f-measure that obtained in this study are precision with a value of 0.637, recall with a value of 0.633, and f-measure with a value of 0.635.

Keywords: genre, song, classification, K-nearest neighbor, cross correlation

**DAFTAR ISI**

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka.....	5
2.2 Landasan Teori.....	6
2.2.1 Lagu.....	6
2.2.2 Genre.....	6
2.2.3 Fitur Akustik.....	7
2.2.3.1 <i>Zero Crossing Rate</i>	7
2.2.3.2 <i>Spectral Centroid</i>	7
2.2.3.3 <i>Spectral Rolloff</i>	7
2.2.3.4 <i>Energy</i>	8
2.2.4 Normalisasi.....	8



2.2.5 Klasifikasi.....	8
2.2.5.1 <i>K-Nearest Neighbor</i>	9
2.2.5.2 <i>Cross Correlation</i>	9
2.2.6 Evaluasi.....	10
2.2.6.1 <i>Precision</i>	10
2.2.6.2 <i>Recall</i>	10
2.2.6.3 <i>F-measure</i>	11
BAB 3 METODOLOGI PENELITIAN.....	12
3.1 Tipe Penelitian.....	12
3.2 Strategi Penelitian.....	12
3.3 Data Penelitian.....	12
3.4 Perancangan Algoritme.....	13
3.5 Teknik Analisis.....	14
3.6 Kesimpulan dan Saran.....	14
3.7 Peralatan Pendukung.....	14
BAB 4 PERANCANGAN DAN IMPLEMENTASI.....	15
4.1 Perancangan Algoritme.....	15
4.1.1 Proses Utama Sistem.....	15
4.1.2 Ekstraksi Fitur.....	16
4.1.3 Preprocessing.....	17
4.1.4 Perhitungan K-NN.....	18
4.1.5 Perhitungan NCC.....	19
4.2 Perhitungan Manual.....	20
4.3 Perancangan Pengujian.....	27
4.3.1 Pengujian Durasi.....	27
4.3.2 Pengujian <i>Offset</i>	28
4.3.3 Pengujian Jumlah Tetangga.....	28
4.4 Implementasi.....	29
4.2.1 Implementasi Ekstraksi Fitur.....	29
4.2.2 Implementasi Perhitungan K-NN.....	30



BAB 5 HASIL DAN PEMBAHASAN.....	32
5.1 Evaluasi <i>Confusion Matrix</i>	32
5.2 Pengujian Durasi.....	33
5.3 Pengujian <i>Offset</i>	34
5.4 Pengujian <i>Tetangga Terdekat</i>	35
BAB 6 PENUTUP.....	38
6.1 Kesimpulan.....	38
6.2 Saran.....	38
DAFTAR REFERENSI.....	39
LAMPIRAN A HASIL KELUARAN PROGRAM.....	41
LAMPIRAN B TABEL CONFUSION MATRIX.....	43

**DAFTAR TABEL**

Tabel 4.1 Data lagu fitur <i>zero crossing rate</i>	20
Tabel 4.2 Data lagu fitur <i>spectral centroid</i>	21
Tabel 4.3 Data lagu fitur <i>spectral rolloff</i>	22
Tabel 4.4 Data lagu fitur <i>energy</i>	23
Tabel 4.5 Data lagu kelas genre.....	23
Tabel 4.6 Data uji fitur <i>zero crossing rate</i>	24
Tabel 4.7 Data uji fitur <i>spectral centroid</i>	24
Tabel 4.8 Data uji fitur <i>spectral rolloff</i>	24
Tabel 4.9 Data uji fitur <i>energy</i>	24
Tabel 4.10 Data hasil perhitungan NCC.....	25
Tabel 4.11 Data tetangga terdekat.....	27
Tabel 4.12 Pengujian durasi.....	28
Tabel 4.13 Pengujian <i>offset</i>	28
Tabel 4.14 Pengujian jumlah tetangga.....	28
Tabel 5.1 <i>Confusion matrix</i>	32
Tabel 5.2 Hasil pengujian durasi.....	33
Tabel 5.3 Hasil pengujian <i>offset</i>	34
Tabel 5.4 Hasil pengujian jumlah tetangga.....	35
Tabel 5.5 Hasil klasifikasi.....	37



DAFTAR GAMBAR

Gambar 3.1 Tahap perancangan algoritme.....	13
Gambar 4.1 Diagram alir proses algoritme.....	15
Gambar 4.2 Diagram alir proses ekstraksi fitur.....	16
Gambar 4.3 Diagram alir proses <i>preprocessing</i>	17
Gambar 4.4 Diagram alir metode <i>k-nearest neighbor</i>	18
Gambar 4.5 Diagram alir persamaan <i>normalized cross correlation</i>	19
Gambar 4.6 Hasil Program.....	31
Gambar 5.1 Evaluasi <i>macro average</i> durasi.....	33
Gambar 5.2 <i>Waveform</i> sebuah lagu berdurasi 30 detik.....	34
Gambar 5.3 Evaluasi <i>macro average</i> offset.....	35
Gambar 5.4 Evaluasi <i>macro average</i> tetangga terdekat.....	36



DAFTAR LAMPIRAN

LAMPIRAN A HASIL KELUARAN PROGRAM.....	38
LAMPIRAN B TABEL CONFUSION MATRIX.....	40
B.1 Confusion Matrix Durasi 10, Offset 120, K 10.....	40
B.2 Confusion Matrix Durasi 20, Offset 120, K 10.....	40
B.3 Confusion Matrix Durasi 30, Offset 120, K 10.....	40
B.4 Confusion Matrix Durasi 10, Offset 30, K 10.....	41
B.5 Confusion Matrix Durasi 10, Offset 60, K 10.....	41
B.6 Confusion Matrix Durasi 10, Offset 90, K 10.....	42
B.7 Confusion Matrix Durasi 10, Offset 150, K 10.....	42
B.8 Confusion Matrix Durasi 10, Offset 120, K 5.....	42
B.9 Confusion Matrix Durasi 10, Offset 120, K 15.....	43



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Lagu adalah sesuatu yang tidak bisa lepas dari aktifitas sehari-hari manusia. Iringan Lagu dapat memberikan nuansa positif pada setiap aktifitas yang dilakukan. Seperti yang dikatakan Al-Qutt et al. (2011), Musik dan lagu adalah seni yang dapat menembus batas dari bahasa, perbedaan adat, dan budaya. Lagu memiliki perbedaan dengan musik, pada lagu terdapat lirik berupa puisi yang dilagukan. Isi dari sebuah lagu biasanya didapat dari peristiwa-peristiwa yang dialami oleh penulisnya, seperti permasalahan pada kehidupannya atau sekitarnya, hal-hal lucu ataupun peristiwa-peristiwa bahagia. Peristiwa-peristiwa ini akan penulis ubah menjadi lirik lagu dengan imajinasinya.

Pada lagu terdapat genre yang digunakan untuk mengelompokkan jenis-jenis dari lagu. Sebuah lagu bisa saja memiliki lebih dari satu jenis genre. Beberapa contoh dari genre adalah Rock, Pop, dan Jazz. Jumlah genre yang ada di dunia ini akan terus bertambah, karena seiring dengan terus berkembangnya lagu maka genre-genre baru akan terus bermunculan.

Seperti yang dikatakan oleh Ren, Fan, dan Ming (2017), pesatnya perkembangan informasi multimedia dan perangkat elektronik menyebabkan peningkatan dramatis pada apresiasi dan penciptaan musik. Pada suatu sisi peningkatan ini mendorong masyarakat untuk lebih menikmati lagu. Tapi di sisi lain, peningkatan ini memaksa pengembangan-pengembangan teknologi baru untuk kenyamanan mendengarkan lagu. Contohnya adalah bagaimana seseorang ingin mencari lagu berdasarkan lagu yang telah didengarkan. Klasifikasi genre lagu adalah salah satu teknik *machine learning* yang dapat mengelompokkan lagu berdasarkan fungsinya. Teknik ini dapat dijadikan fungsi dalam suatu sistem untuk mendukung fungsi-fungsi lainnya, seperti perekomendasi lagu, pencarian kata-kata khusus, atau pencarian lagu yang mirip.

Aplikasi atau *software* musik telah banyak bermunculan dan setiap aplikasi biasanya memiliki berbagai macam fitur. Fitur-fitur umum yang ada pada aplikasi musik adalah *playlist*, *favorite*, klasifikasi lagu dan fitur lainnya. Klasifikasi lagu adalah bagaimana sistem dapat mengelompokkan lagu-lagu berdasarkan tipenya. Pengelompokkan tipe ini dapat berbentuk genre, *mood*, frekuensi, ataupun bahasa. Biasanya klasifikasi lagu ini menggunakan data yang didapat dari *meta data* sebuah *file* musik seperti informasi album, artis, atau genre lagu tersebut, tetapi pada penelitian ini data yang diambil berasal dari fitur akustik sebuah lagu. Hasil ekstraksi fitur akustik ini digunakan untuk membandingkan performa dari dua buah fitur lagu (Costa et al., 2012). Fitur akustik ini adalah representasi dari informasi lagu yang didapatkan dengan menjalankan algoritme-algoritme tertentu.

Terdapat penelitian yang berkaitan dengan klasifikasi genre lagu, yaitu Klasifikasi Lagu Berdasarkan Genre pada Format WAV yang dilakukan oleh



Tamatjita dan Harjoko (2014), yang menggunakan fitur *zero crossing rate*, *average energy*, dan *silent ratio*. Penelitian ini memiliki hasil terbaik sebesar 96,67% ketika genre yang digunakan berjumlah tiga. Tetapi terjadi penurunan akurasi di setiap perubahan genre ketika pengujian. Penelitian lainnya adalah penelitian yang dilakukan oleh Guo et al. (2012) yang menggunakan metode klasifikasi *Support Vector Machine* untuk mengelompokkan data menjadi dua kelas. Pengklasifikasian ini dapat dikembangkan lagi menjadi *multiclass classification* sehingga sistem dapat mengelompokkan data dengan lebih detail.

Penelitian yang terkait dengan metode K-NN adalah penelitian tentang pencarian *chord* yang dapat membantu penciptaan lagu dengan menggunakan algoritma *Fast Fourier Transform* dan metode K-NN. Dalam penelitian ini dilakukan pengujian *chord*, yang didapatkan menggunakan lagu tanpa musik dan lagu dengan musik. Hasil akurasi yang didapat adalah 82,35% untuk lagu tanpa musik dan 22,22% untuk lagu dengan musik (Fetra dan Irsyad, 2015). Pada penelitian lainnya dilakukan pengklasifikasian suara yang digunakan pada penerapan kunci otomatis menggunakan metode K-NN. Dalam penelitian ini dihasilkan akurasi 88,88% saat sistem menguji suara langsung pengguna (Apriansyah, Ilhamsyah dan Rismawan, 2016). Dari kedua penelitian ini didapatkan bahwa metode K-NN memiliki nilai akurasi yang baik ketika mengklasifikasikan sinyal suara, ditambah nilai K dalam K-NN dapat meningkatkan nilai akurasi yang didapat dalam sistem.

Berdasarkan uraian di atas dan penelitian-penelitian sebelumnya, maka dibuatlah penelitian untuk menentukan klasifikasi lagu terbaik menggunakan metode K-NN, data yang digunakan adalah sinyal suara dari lagu yang didapatkan menggunakan *music information retrieval*. Penggunaan sinyal suara sendiri dipilih untuk memberikan nilai kemiripan yang tinggi antara lagu dengan klasifikasi lagunya.

1.2 Rumusan Masalah

Berdasarkan latar belakang penelitian ini, maka dibuatlah rumusan masalah sebagai berikut:

1. Bagaimana rancangan algoritme dan sistem menggunakan metode K-NN dengan fitur akustik dapat mengklasifikasi lagu berdasarkan genre?
2. Bagaimana hasil akurasi yang didapatkan dari pengujian metode K-NN pada pengklasifikasian genre lagu menggunakan fitur akustik?

1.3 Tujuan

Berdasarkan dari rumusan masalah yang telah ditetapkan, maka tujuan dari penelitian ini adalah:

1. Merancang sistem yang dapat mengklasifikasi lagu berdasarkan genre dengan metode K-NN.



2. Mendapatkan nilai akurasi dari penelitian Penentuan Klasifikasi Genre Lagu Menggunakan metode K-NN.

1.4 Manfaat

Manfaat yang diharapkan dari penelitian ini adalah:

1. Membantu pengguna untuk menemukan genre yang cocok dari lagu yang disukainya.
2. Membantu programmer yang ingin membuat aplikasi musik dengan klasifikasi genre lagu yang lebih akurat.

1.5 Batasan Masalah

Penelitian ini memiliki beberapa batasan, antara lain:

1. Penelitian ini hanya akan menggunakan lagu dengan bahasa Inggris sebagai datanya.
2. Kualitas *bitrate* seluruh *file* lagu yang digunakan pada penelitian ini akan dikonversi menjadi 128 kBit/s.
3. Format lagu yang digunakan pada penelitian ini adalah .mp3 dan .m4a.

1.6 Sistematika Pembahasan

Sistematika penulisan laporan bertujuan untuk memberikan uraian serta gambaran dari penulisan laporan skripsi ini yang meliputi beberapa bab, sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini berisi bahasan mengenai latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan dari penelitian Klasifikasi Genre Lagu dengan Fitur Akustik Menggunakan Metode *K-Nearest Neighbor*.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi bahasan mengenai kajian pustaka dan landasan teori yang berhubungan dengan penelitian yang akan dibuat. Kajian pustaka berisi penelitian-penelitian sebelumnya yang berkaitan dengan penelitian yang akan dibuat. Landasan teori merupakan teori-teori yang berhubungan dengan metode dan objek yang digunakan dalam penelitian Klasifikasi Genre Lagu dengan Fitur Akustik Menggunakan Metode *K-Nearest Neighbor*.

BAB 3 METODOLOGI

Bab ini berisi bahasan mengenai metodologi serta langkah kerja yang berkaitan dengan penelitian. Langkah kerja terdiri dari tipe penelitian, strategi penelitian, data penelitian, implementasi algoritme, teknik analisis, lingkungan



pengembangan sistem, kesimpulan dan saran dalam penelitian Klasifikasi Genre Lagu dengan Fitur Akustik Menggunakan Metode *K-Nearest Neighbor*.

BAB 4 PERANCANGAN DAN IMPLEMENTASI

Bab ini berisi bahasan mengenai perancangan, perhitungan manual, serta ulasan perancangan dari penelitian dan juga implementasi sistem beserta penjelasannya yang berkaitan dengan metode *K-Nearest Neighbor*.

BAB 5 HASIL DAN PEMBAHASAN

Bab ini berisi bahasan mengenai hasil pengujian dan analisis penelitian yang dilakukan untuk mendapatkan akurasi dari sistem yang telah dibuat.

BAB 6 PENUTUP

Bab ini berisi ulasan mengenai kesimpulan dan saran dari penulis. Kesimpulan berisi analisis dan hasil pengujian yang telah dilakukan. Saran berisi masukan-masukan untuk pengembangan selanjutnya dalam penelitian Klasifikasi Genre Lagu dengan Fitur Akustik Menggunakan Metode *K-Nearest Neighbor*.



BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini akan dijelaskan penelitian-penelitian apa saja yang menjadi referensi pada penelitian ini. Dijelaskan pula landasan teori dari objek, metode dan persamaan yang digunakan pada penelitian Klasifikasi Genre Lagu dengan Fitur Akustik Menggunakan Metode *K-Nearest Neighbor*.

2.1 Kajian Pustaka

Kajian pustaka pada penelitian ini didasarkan pada beberapa penelitian yang sebelumnya pernah dilakukan berkaitan dengan metode klasifikasi *K-Nearest Neighbor* seperti penelitian yang berkaitan dengan klasifikasi genre yang dilakukan oleh Tamatjita dan Harjoko (2014) penelitian ini mengklasifikasi genre menggunakan lagu dengan format WAV. Fitur yang digunakan pada penelitian tersebut adalah *zero crossing rate*, *average energy*, dan *silent ratio*. Pengujian dilakukan berdasarkan jumlah genre dengan nilai 3, 6, 9, dan 12. Didapatkan akurasi terbaik bernilai 96,67% yang menggunakan tiga genre dan akurasi terendah bernilai 33,33% ketika menggunakan dua belas genre. Penelitian lainnya adalah penelitian yang dilakukan oleh Guo et al. (2012) yang menggunakan metode klasifikasi *Support Vector Machine* untuk mengelompokkan data menjadi dua kelas yaitu, nyanyian atau senandung. Pengklasifikasian dengan dua kelas ini dirasa cukup bermasalah, karena ketika data senandung masuk kedalam kelas nyanyian maka data ini akan mengakibatkan *noise*. Ini disebabkan data senandung tidak memiliki informasi lirik di dalamnya sedangkan kelas nyanyian membutuhkan informasi lirik. Pengklasifikasian ini dapat dikembangkan lebih lanjut lagi menjadi *multiclass classification* yang berfungsi untuk mengelompokkan data dengan kelas lebih dari dua jenis.

Penelitian sebelumnya yang berkaitan dengan K-NN adalah penelitian Aplikasi Pencarian *Chord* dalam Membantu Penciptaan Lagu Menggunakan Algoritma FFT dan Metode Klasifikasi K-NN. Dalam penelitiannya dilakukan proses penghitungan durasi suatu *chord*, pemotongan lagu, pencarian frekuensi menggunakan algoritme FFT, terakhir dilakukan pencarian *chord* menggunakan metode K-NN Berdasarkan data pelatihannya. Hasil keluaran yang didapat pada penelitian ini adalah *chord* yang menjadi saran untuk pembuatan lagu. Terdapat dua pengujian yang dilakukan pada penelitian ini yaitu pengujian menggunakan lagu tanpa musik yang menghasilkan akurasi sebesar 82,35% dan pengujian menggunakan lagu dengan musik yang menghasilkan akurasi sebesar 22,22% (Fetra dan Irsyad, 2015). Terdapat penelitian lain yang melakukan klasifikasi pada penerapan kunci otomatis menggunakan metode K-NN. Parameter yang digunakan adalah *average energy*, *zero crossing rate*, dan *spectral centroid*. Pengujian dilakukan menggunakan tiga cara yaitu, pengujian menggunakan suara pengguna berhak, pengujian menggunakan suara pengguna tidak berhak, dan pengujian menggunakan suara rekaman pengguna berhak. Hasil yang didapat



dari tiga pengujian ini adalah 88,88% untuk pengujian suara pengguna berhak, 100% untuk pengujian suara pengguna yang tidak berhak, dan 100% untuk pengujian suara rekaman pengguna berhak (Apriansyah, Ilhamsyah dan Rismawan, 2016).

Dari penelitian-penelitian yang telah dilakukan sebelumnya maka pada penelitian ini peneliti mengusulkan untuk membangun sistem klasifikasi lagu dengan judul “Klasifikasi genre lagu dengan fitur akustik menggunakan metode *K-Nearest Neighbor*”. Pada penelitian ini digunakan MIR untuk mengambil data dan fitur pada sebuah lagu digital, data ini akan diproses sehingga nantinya dapat diolah menggunakan metode K-NN menggunakan persamaan *cross correlation*. Metode K-NN digunakan karena data yang didapat dari pengambilan data dan fitur lagu digital berupa data numerik, yang membuat metode K-NN dapat menentukan hasil terbaik dari perbandingan jarak dari data latih dan data ujinya. Kemudian, persamaan jarak yang ada dalam metode K-NN akan diganti dengan persamaan *cross correlation* karena persamaan *cross correlation* dapat mengetahui seberapa identik dua buah sinyal (Seppänen, 1999).

2.2 Landasan Teori

Pada subbab ini akan dijelaskan teori-teori dari metode, objek, dan persamaan yang digunakan dalam penelitian ini. Terdapat tiga teori utama yaitu lagu, klasifikasi, dan evaluasi.

2.2.1 Lagu

Lagu adalah gubahan seni nada, ragam nada, suara yang berirama, atau suara yang memiliki urutan, kombinasi, dan hubungan temporal untuk menghasilkan gubahan musik yang mempunyai kesatuan dan kesinambungan. Setiap lagu memiliki informasi tersendiri seperti genre, nama album, penyanyi, dan penulis. Informasi ini dapat membantu masyarakat untuk memiliki lagu sesuai ketertarikan mereka. Informasi ini juga berguna bagi komputer untuk memutar lagu berdasarkan genre atau album atau mencari lagu yang mirip dari informasi suatu lagu.

2.2.2 Genre

Genre adalah salah satu label musik yang digunakan untuk mengelompokkan musik ataupun lagu berdasarkan teknik, musik, gaya, konteks, dan tema musik. Schedi, Yang, dan Herrera-Boyer (2016) mengatakan bahwa genre adalah salah satu label musik yang paling banyak digunakan dalam rekomendasi musik dan memiliki performa yang baik digunakan. Umumnya setiap genre akan memiliki subgenre lagi yang mendefinisikan genre lebih detail lagi, seperti *subgenre* Bossa nova yang berasal dari genre Jazz.



2.2.3 Fitur Akustik

Fitur akustik merupakan informasi bernilai numerik yang didapatkan dari data sebuah audio lagu. Hasil ekstraksi fitur akustik digunakan untuk membandingkan performa dari dua buah fitur lagu (Costa et al., 2012). Namun kebanyakan fitur akustik tidak tertulis secara eksplisit sehingga membutuhkan algoritme-algoritme tertentu untuk mengambil fitur akustik tersebut. Fitur-fitur yang diekstraksi pada penelitian Klasifikasi Genre Lagu dengan Fitur Akustik Menggunakan Metode *K-Nearest Neighbor* adalah *zero crossing rate*, *spectral centroid*, *spectral rolloff*, dan *energy* yang akan dijelaskan pada subbab 2.2.1.1-2.2.1.4.

2.2.3.1 Zero Crossing Rate

Zero crossing rate adalah ukuran berapa kali sinyal melewati sumbu horizontal. Sinyal bisa berubah dari tanda positif ke nol lalu ke negatif atau dari negatif ke nol lalu ke positif (Apriansyah, Ilhamsyah dan Rismawan, 2016).

$$ZCR = \frac{i}{2N} \sum_{n=0}^{N-1} |sgn[x_i(n)] - sgn[x_i(n-1)]| \quad (2.1)$$

dengan

$$sgn[x_i(n)] = \begin{cases} 1, & x_i(n) \geq 0 \\ -1, & x_i(n) < 0 \end{cases} \quad (2.2)$$

Keterangan:

N = panjang sinyal

$x_i(n)$ = sampel dari *frame* i ($0, 1, \dots, N-1$)

2.2.3.2 Spectral Centroid

Spectral Centroid menunjukkan di frekuensi mana suatu energi dari spektrum terpusat. *Spectral Centroid* digunakan pula untuk memprediksi kecerahan suatu timbral (Apriansyah, Ilhamsyah dan Rismawan, 2016).

$$C(i) = \frac{\sum_{k=0}^{N-1} k |X_i(k)|}{\sum_{k=0}^{N-1} |X_i(k)|} \quad (2.3)$$

Keterangan:

$x_i(k)$ = urutan koefisien *discrete fourier transform* ($0, 1, \dots, N-1$)

2.2.3.3 Spectral Rolloff

Spectral Rolloff memiliki kemiripan dengan *Spectral Centroid* salah satunya adalah nilainya yang memiliki *range* yang sama. *Spectral Rolloff* menunjukkan frekuensi di bagian bawah dari total *spectral energy* (Stevetjoa, 2014).



$$\sum_{k=0}^{k=i} |X_i(k)| = \frac{P}{100} \sum_{k=0}^{N-1} |X_i(k)| \quad (2.4)$$

Keterangan:

$x_i(k)$ = urutan koefisien *discrete fourier transform* (0, 1, ..., N-1)

P = nilai yang ditentukan ($80 \leq P \leq 100$)

2.2.3.4 Energy

Energy adalah fitur yang merefleksikan variasi amplitudo pada sebuah sinyal. *Energy* didapat dengan mengkuadratkan setiap *sample* yang ada dalam suatu sinyal kemudian menjumlahkan semuanya (Stevetjoa, 2014).

$$E = \sum_n |x(n)|^2 \quad (2.5)$$

Keterangan:

$x(n)$ = sampel pada *index* n

2.2.4 Normalisasi

Normalisasi dilakukan untuk menyamaratakan *range* nilai yang dimiliki setiap fitur. Normalisasi juga dapat meningkatkan akurasi dan efisiensi dari algoritme *nearest neighbor*. Normalisasi dihitung menggunakan *mean* dan standar deviasi dari data yang digunakan (Mohabeer, Soyjaudah dan Pavaday, 2011).

$$S_k' = \frac{(S_k - \mu)}{\sigma} \quad (2.6)$$

Keterangan:

S_k = data

μ = *mean*

σ = standar deviasi

2.2.5 Klasifikasi

Klasifikasi adalah salah satu teknik pengelompokan yang digunakan untuk menentukan kategori atau kelas benda berdasarkan karakteristiknya secara otomatis (Bello, 2014). Penentuan klasifikasi kategori ini dilakukan dengan cara membandingkan data uji dan data latih atau biasa disebut *supervised learning* (Murugappan, 2011). semakin banyak data latih yang mirip dengan data uji pada suatu kelas, maka data uji tersebut akan masuk kedalam kelas tersebut.



2.2.5.1 K-Nearest Neighbor

K-Nearest Neighbor (K-NN) adalah metode klasifikasi yang menggunakan algoritme supervised dengan data uji yang akan diklasifikasikan berdasarkan mayoritas kelas. algoritme ini dikembangkan oleh Fix dan Hodges dan termasuk algoritme pengklasifikasian yang sederhana dengan akurasi yang baik.

algoritme ini bekerja dengan cara mengklasifikasi data uji dengan mempelajari kedekatannya dengan data latih. Kelas data uji ini didapatkan dari mayoritas kelas terbanyak pada K data, K adalah jumlah data dengan jarak terdekat (tetangga terdekat) dari data uji (Fetra dan Irsyad, 2015).

$$E(x, y) = \sqrt{\sum_{i=1}^k (x_i - y_i)^2} \quad (2.7)$$

Keterangan:

- E = Jarak
- x = data latih
- y = data uji
- k = dimensi data
- i = variabel data

2.2.5.2 Cross Correlation

Cross correlation adalah sebuah teknik statistik yang digunakan untuk mengukur tingkat kemiripan dari dua buah sinyal yang berhubungan (Seppänen, 1999). Pada penelitian Klasifikasi Genre Lagu dengan Fitur Akustik Menggunakan Metode *K-Neareast Neighbor* digunakan *normalized cross correlation (NCC)* yang dapat mengukur tingkat kemiripan dua buah sinyal yang memiliki rentang nilai yang cukup jauh berbeda.

$$nCrossCorrelation = \frac{\sum_{n=0}^{n-1} x[n] * y[n]}{\sqrt{\sum_{n=0}^{n-1} x[n]^2 * \sum_{n=0}^{n-1} y[n]^2}} \quad (2.8)$$

Keterangan:

- x = frame data latih
- y = frame data uji
- n = jumlah frame yang ada pada data



2.2.6 Evaluasi

Evaluasi penelitian dilakukan untuk mengetahui keberhasilan implementasi metode pada suatu masalah. Pada penelitian ini digunakan evaluasi *confusion matrix multiclass problem* yang mengambil tiga tipe evaluasi yaitu, *precision*, *recall*, *f-measure* (Asch, 2013).

2.2.6.1 Precision

Evaluasi *precision* adalah tingkat ketepatan informasi yang dikeluarkan sistem, *precision* didapat dengan mencari jumlah data relevan dari *output* yang diterima. Pada Persamaan 2.4 *precision* didapat dari jumlah data *output* yang relevan dibagi dengan jumlah *output* yang didapat. Setelah didapatkan nilai *precision* dari setiap kelas maka akan dilakukan pencarian rata-rata *precision* menggunakan persamaan *micro average precision* yang ditunjukkan pada Persamaan 2.5 dan persamaan *macro average precision* yang ditunjukkan pada Persamaan 2.6.

$$Precision = \frac{TP}{TP+FP} \quad (2.9)$$

$$MicAvgP = \frac{TP_1 + \dots + TP_N}{TP_1 + FP_1 + \dots + TP_N + FP_N} \quad (2.10)$$

$$MacAvgP = \frac{Precision_1 + \dots + Precision_N}{N} \quad (2.11)$$

Keterangan:

TP = (*True Positive*) jumlah keluaran data yang sesuai dengan kelas aslinya

FP = (*False Positive*) jumlah keluaran data yang salah pada kelas N

N = jumlah data uji yang digunakan

2.2.6.2 Recall

Evaluasi *recall* adalah tingkat keberhasilan kesesuaian informasi yang didapat dari *output* sistem, *recall* didapat dengan mencari jumlah data relevan yang diterima. Pada Persamaan 2.6 di bawah ini, *recall* didapat dari jumlah *output* yang relevan dibagi dengan jumlah data yang relevan. Setelah didapatkan nilai *recall* dari setiap kelas maka akan dilakukan pencarian rata-rata *recall* menggunakan persamaan *micro average recall* yang ditunjukkan pada Persamaan 2.7 dan *macro average recall* yang ditunjukkan pada Persamaan 2.8.



$$\text{Recall} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}} \quad (2.12)$$

$$\text{MicAvgR} = \frac{TP_1 + \dots + TP_N}{TP_1 + FN_1 + \dots + TP_N + FN_N} \quad (2.13)$$

$$\text{MacAvgR} = \frac{\text{Recall}_1 + \dots + \text{Recall}_N}{N} \quad (2.14)$$

Keterangan:

TP = (True Positive) jumlah keluaran data yang sesuai dengan kelas aslinya

FN = (False Negative) jumlah data yang salah pada kelas N

N = jumlah data uji yang digunakan

2.2.6.3 F-measure

Evaluasi *F-measure* adalah bobot *harmonic mean* dari *recall* dan *precision*. Pada persamaan *F-measure*, *precision* akan dilambangkan dengan P dan *recall* akan dilambangkan dengan R . Untuk mendapatkan nilai *f-measure* digunakan Persamaan 2.9. Persamaan ini digunakan pula untuk menghitung nilai *micro average f-measure* ataupun *macro average f-measure*.

$$F - \text{Measure} = \frac{2 * P * R}{P + R} \quad (2.15)$$

Keterangan:

P = Precision

R = Recall



BAB 3 METODOLOGI PENELITIAN

Bab ini menjelaskan tahapan – tahapan pada penelitian “Klasifikasi Lagu dengan Fitur Akustik Menggunakan Metode *K-Nearest Neighbor*”. Di dalamnya terdapat tahapan-tahapan penelitian secara umum yang akan dijelaskan secara menyeluruh. Tahapan-tahapan tersebut terdiri dari tipe penelitian, strategi penelitian, data penelitian, implementasi algoritme, teknik analisis, kesimpulan dan saran, dan lingkungan pengembangan yang digunakan oleh sistem.

3.1 Tipe Penelitian

Tipe penelitian yang digunakan pada penelitian ini adalah non-implementatif dan fokus masalah berupa klasifikasi genre lagu menggunakan metode K-NN dengan hasil akhir berbentuk sistem. Penelitian non-implementatif akan mengamati segala hal-hal yang sedang diteliti dalam penelitian guna dapat memberikan kesimpulan berupa hasil analisis sebagai hasil utama dari penelitian ini. Pendekatan analitik digunakan pada penelitian ini untuk membantu peneliti memahami data, karakteristik fitur, dan juga kelas yang digunakan.

3.2 Strategi Penelitian

Strategi penelitian yang digunakan adalah teknik eksperimen, teknik eksperimen digunakan untuk menemukan keterkaitan metode yang digunakan dengan objek data yang dipakai, dalam kasus ini adalah sistem klasifikasi terhadap genre lagu berdasarkan fitur akustik. Metode yang digunakan pada penelitian ini adalah metode *K-Nearest Neighbor*. Pada penelitian ini digunakan MIR untuk mengekstraksi data dan fitur yang ada pada lagu yang kemudian akan dicari tingkat kemiripan antara data latih dan data uji menggunakan persamaan NCC lalu metode K-NN digunakan untuk menentukan kelas mayoritas dari tetangga sejumlah K yang diambil.

3.3 Data Penelitian

Pada tahap ini dilakukan pengumpulan data yang berkaitan dengan Klasifikasi Lagu dengan Fitur Akustik Menggunakan Metode *K-nearest Neighbor*. Data yang digunakan adalah 100 file lagu yang memiliki format .mp3 atau .m4a dan kualitas *bitrate* dari lagu telah dikonversi menjadi 128 kBit/s. Data ini akan dibagi dua menjadi 70 data untuk data latih dan 30 data untuk data uji.

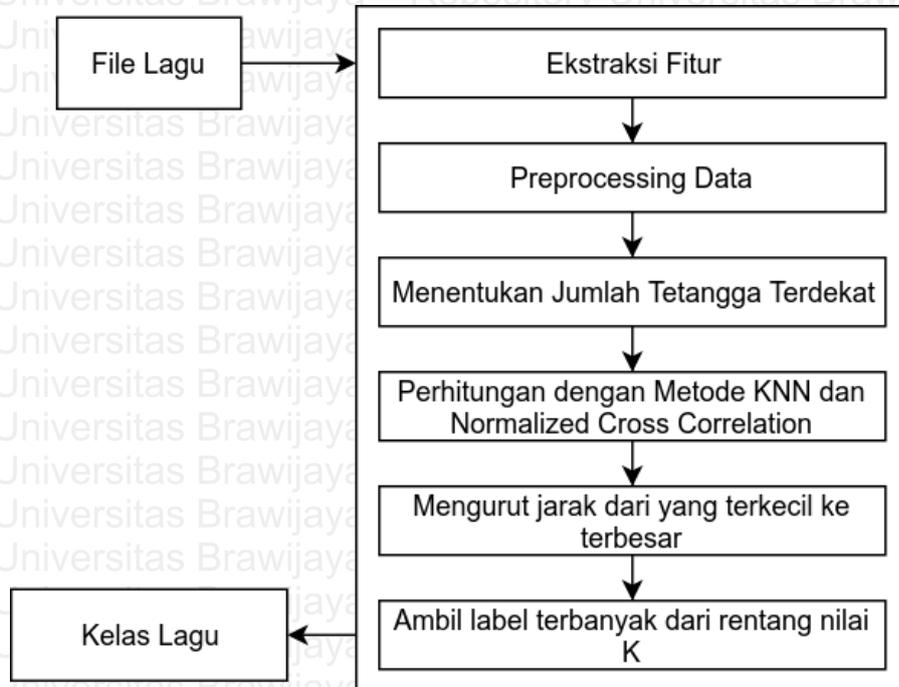
Data dalam penelitian ini menggunakan *single label*, yang menyatakan setiap lagu memiliki satu genre. Genre yang akan digunakan sebagai label berjumlah 5, yaitu Rock, Electronic, Pop, Jazz, dan Hip Hop. Informasi genre didapat dari *website* discogs.com dan genre yang diambil adalah genre dengan nilai voting terbesar dalam sebuah lagu.

Saat pengubahan audio lagu menjadi data numerik, digunakan dua parameter untuk membatasi pengambilan data yang diambil. Parameter



tersebut adalah durasi dan *offset*. Durasi adalah jumlah waktu yang diambil pada sebuah data dan *offset* adalah waktu awal ketika pengambilan data dimulai.

3.4 Perancangan Algoritme



Gambar 3.1 Tahap perancangan algoritme

Tahap perancangan algoritme adalah tahap penggambaran jalannya *input*, proses, dan *output*. Pada tahap ini akan dijelaskan *input*, proses, dan *output* dari penggunaan metode K-NN. Berikut adalah langkah-langkah pada tahap perancangan algoritme yang mengacu pada Gambar 3.1:

1. Dilakukan ekstraksi fitur pada *input*. Ekstraksi fitur ini dilakukan untuk mendapatkan empat fitur akustik dari sebuah lagu, ke empat fitur ini adalah *zero crossing rate*, *spectral centroid*, *spectral rolloff*, dan *energy*.
2. Dilakukan normalisasi data pada fitur-fitur akustik yang telah diekstraksi.
3. Ditentukan nilai K atau tetangga terdekat dari metode K-NN.
4. Perhitungan dengan Metode K-NN dengan persamaan jarak yang diganti menjadi persamaan *normalized cross correlation*.
5. Mengurutkan hasil yang didapatkan dari metode K-NN berdasarkan nilai terkecil ke nilai terbesar.
6. Mengambil data yang telah diurutkan sebanyak K dalam K-NN, kemudian diambil mayoritas kelas dari tetangga terdekat (K) sebagai kelas data uji.



3.5 Teknik Analisis

Pada tahap ini dilakukan pengujian keberhasilan dan akurasi dari algoritme yang telah diimplementasi pada sistem. Pengujian sistem dilakukan dengan cara melakukan evaluasi *confusion matrix multiclass problem* untuk mencari nilai *precision*, *recall*, dan *f-measure*. Hasil evaluasi yang telah didapat nantinya akan dibuat grafik untuk dapat dianalisis.

3.6 Kesimpulan dan Saran

Tahap terakhir adalah Kesimpulan dan saran. Kesimpulan didapat dari hasil pengumpulan data, pengolahan data, implementasi dan pengujian dari sistem pada penelitian ini. Kesimpulan yang diambil harus sesuai dengan rumusan masalah yang telah ditulis sebelumnya. Setelahnya, penulis akan menuliskan saran untuk penelitian selanjutnya agar penelitian selanjutnya memiliki hasil yang lebih baik.

3.7 Peralatan Pendukung

Pada lingkungan pengembangan sistem ditunjukkan lingkungan sistem yang digunakan ketika sistem Klasifikasi Genre Lagu dengan Fitur Akustik Menggunakan Metode *K-Nearest Neighbor* dibuat dan diimplementasikan. Deskripsi dari lingkungan pengembangan yang dilakukan pada pengembangan sistem penelitian ini adalah sebagai berikut:

1. Perangkat keras:
 1. *Processor* intel® Core™ i5-3210M CPU @ 2.50GHz 2.50GHz
 2. *Memory* 4 GB RAM
2. Perangkat lunak:
 1. Sistem Operasi Antergos
 2. Python 3.6.0
 3. PyCharm Python IDE Community Version
 4. Jupyter Notebook
 5. Librosa
 6. Numpy
 7. Pickle



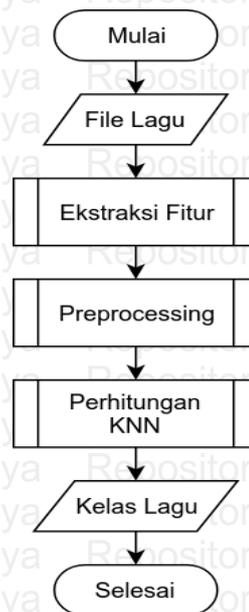
BAB 4 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan dibahas mengenai perancangan yang digunakan untuk membangun sistem dan implementasinya pada sistem Klasifikasi Genre Lagu Dengan Fitur Akustik menggunakan metode *K-Nearest Neighbor*, perancangan dibuat berdasarkan teori-teori yang telah dibahas pada bab sebelumnya. Terdapat dua jenis perancangan yang dilakukan pada penelitian ini yaitu, perancangan algoritme dan perancangan pengujian. Kemudian terdapat tahap implementasi yang menggunakan bahasa pemrograman Python dengan dua kelas sebagai proses ekstraksi fitur dan proses klasifikasi genre.

4.1 Perancangan Algoritme

4.1.1 Proses Utama Sistem

Dalam perancangan algoritme akan dijelaskan setiap proses yang dilakukan oleh sistem klasifikasi lagu dengan fitur akustik menggunakan metode K-NN. Subbab 4.1.1 akan menjelaskan jalannya keseluruhan sistem. Gambar 4.1 menunjukkan diagram alir dari keseluruhan sistem yang menunjukkan proses dari penginputan data sampai dibentuknya kelas lagu oleh sistem.



Gambar 4.1 Diagram alir proses algoritme

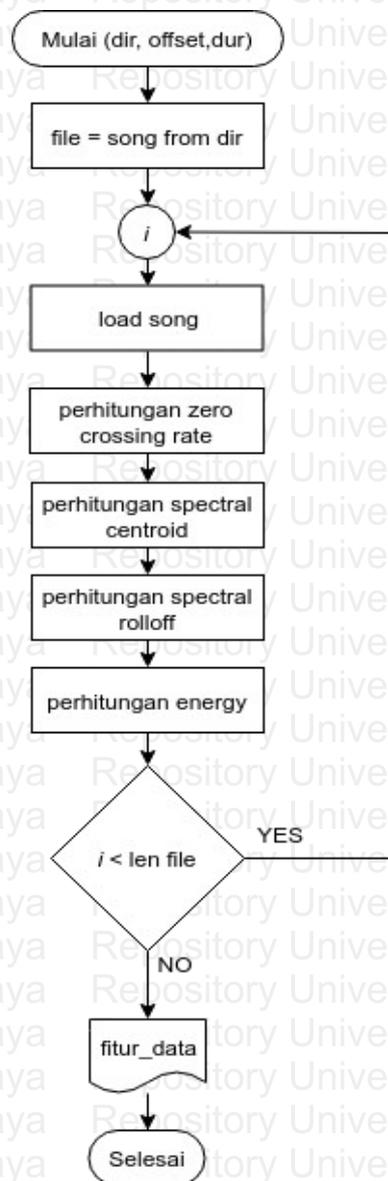
Berikut ini adalah penjelasan dari setiap tahapan yang ada pada diagram alir proses algoritme:

1. Sistem menerima *input* berupa *file* lagu dengan format .mp3 atau .m4a. File lagu ini kemudian dipanggil dengan *library* Librosa menggunakan fungsi *load* untuk mengubah file audio menjadi kumpulan data numerik yang merepresentasikan *waveform* yang ada pada file audio, perubahan data menjadi nilai numerik dilakukan untuk memudahkan sistem mengolah data yang akan diklasifikasi.



2. Proses ekstraksi fitur dilakukan untuk mengambil data data pada *file* lagu. Pengekstraksian fitur dilakukan menggunakan *Library Librosa* dan fitur yang digunakan adalah *zero crossing rate*, *spectral centroid*, *spectral rolloff*, dan *energy*.
3. Proses *preprocessing* adalah proses normalisasi data yang telah diambil pada proses ekstraksi fitur. Pada proses ini digunakan persamaan *z-score* untuk menormalisasi data.
4. Proses perhitungan K-NN dilakukan untuk mengklasifikasikan data berdasarkan jarak kedekatannya. Penghitungan Jarak diganti dengan penghitungan kemiripan sinyal menggunakan persamaan NCC.
5. Hasil keluaran dari sistem berupa kelas lagu yang didapat berdasarkan mayoritas kelas yang ada pada tetangga terdekat dari perhitungan K-NN.

4.1.2 Ekstraksi Fitur



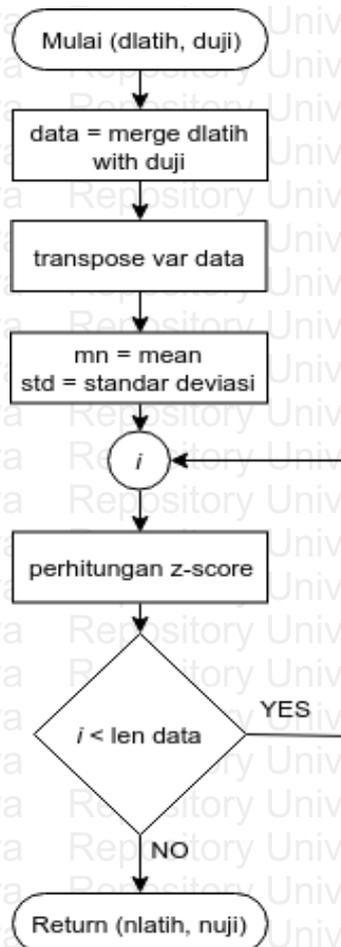
Gambar 4.2 Diagram alir proses ekstraksi fitur



Gambar 4.2 menjelaskan tahapan proses yang ada pada proses ekstraksi fitur. Berikut ini adalah penjelasan dari setiap tahapan yang ada pada diagram alir proses ekstraksi fitur:

1. *Input* pada proses ekstraksi fitur adalah *dir* yang menentukan apakah data yang diambil adalah data latih atau data uji, durasi yang menentukan waktu yang diambil dalam sebuah lagu, dan *offset* yang menentukan waktu dimulainya data diambil.
2. Kemudian dilakukan pengambilan nama-nama lagu di dalam direktori berdasarkan variabel *dir*.
3. Dilakukan perulangan berdasarkan jumlah lagu. Di setiap perulangan dilakukan perubahan audio lagu menjadi data numerik, kemudian dilakukan proses ekstraksi fitur *zero crossing rate*, *spectral centroid*, *spectral rolloff*, dan *energy*.
4. Hasil keluaran proses adalah *fitur_data* yang menyimpan semua nilai fitur ekstraksi pada proses. Hasil ini disimpan dalam sebuah file biner berdasarkan nilai dari parameter yang digunakan.

4.1.3 Preprocessing



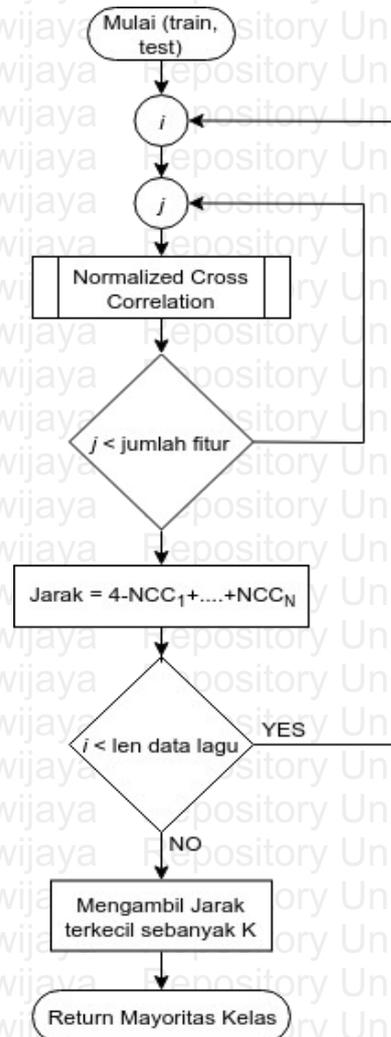
Gambar 4.3 Diagram alir proses *preprocessing*



Gambar 4.3 menjelaskan tahapan proses yang ada pada metode *preprocessing*. Berikut ini adalah penjelasan dari setiap tahapan yang ada pada diagram alir proses perhitungan K-NN:

1. Pada diagram alir ini *input* berupa data latih dan data uji lagu.
2. Selanjutnya kedua data *input* akan digabung dengan fungsi merge.
3. Data yang telah digabung kemudian di-transpose untuk mengelompokkan nilai berdasarkan fiturnya.
4. Kemudian dicari nilai rata-rata dan standar deviasinya dan disimpan pada variabel mean dan std.
5. Lalu dilakukan perulangan yang menormalisasi seluruh data menggunakan normalisasi *z-score*.
6. Kemudian hasil dari proses adalah data yang telah dipisah kembali menjadi nuji dan nlatih.

4.1.4 Perhitungan K-NN



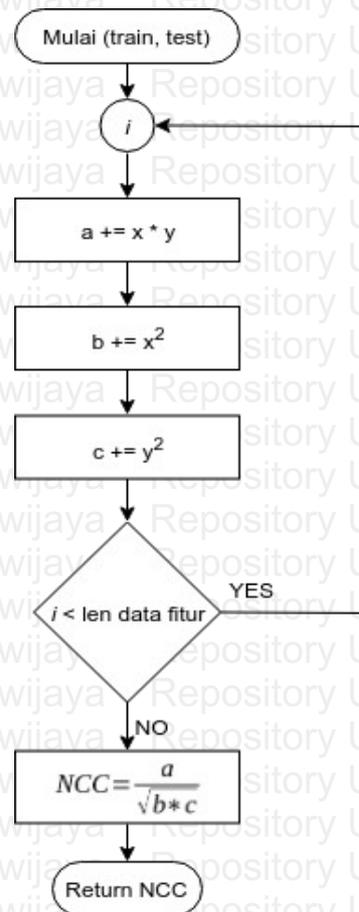
Gambar 4.4 Diagram alir metode *k-nearest neighbor*



Gambar 4.4 menjelaskan tahapan proses yang ada pada metode K-NN. Berikut ini adalah penjelasan dari setiap tahapan yang ada pada diagram alir proses perhitungan K-NN:

1. Pada diagram alir ini *input* yang diterima berupa data lagu, hasil yang didapat dari proses ekstraksi fitur.
2. Selanjutnya dilakukan proses nested loop yang melakukan perulangan sebanyak data latih yang digunakan untuk perulangan pertama dan perulangan lagi sebanyak fitur yang digunakan untuk perulangan kedua.
3. Proses *normalized cross correlation* dilakukan untuk setiap fitur yang ada pada lagu.
4. Setelah perulangan pada setiap fitur lagu dilakukan proses pengurangan nilai 4 dengan nilai penjumlahan seluruh nilai NCC pada sebuah lagu.
5. Setelah perulangan seluruh data latih dilakukan pengambilan jarak terkecil sebanyak K yang telah ditentukan.
6. Hasil dari proses ini adalah kelas yang didapat dari kelas terbanyak ketika pengambilan jarak sejumlah K dari K-NN.

4.1.5 Perhitungan NCC



Gambar 4.5 Diagram alir persamaan *normalized cross correlation*



Gambar 4.5 menjelaskan tahapan proses yang dilakukan untuk menghitung persamaan NCC. Berikut ini adalah penjelasan dari setiap tahapan yang ada pada diagram alir proses perhitungan *normalized cross correlation*:

1. *Input* yang diterima pada proses ini berupa *list* atau *array* dari sebuah fitur akustik lagu, terdapat dua input yang digunakan yaitu data fitur latih dan data fitur uji.
2. Dilakukan perulangan sebanyak panjang *array* yang dimulai dari *index* 0.
3. Dilakukan proses penambahan variabel *a* dengan nilai yang didapatkan dari perkalian *x* dengan *y*.
4. Dilakukan proses penambahan variabel *b* dengan nilai yang didapatkan dari nilai *x* kuadrat.
5. Dilakukan proses penambahan variabel *c* dengan nilai yang didapatkan dari nilai *y* kuadrat.
6. Setelah perulangan selesai akan dilakukan penyimpanan nilai *a* dibagi akar dari *b* dikali *c* pada variabel NCC.

4.2 Perhitungan Manual

Dalam perhitungan manual dilakukan perhitungan manual metode K-NN untuk menjelaskan perhitungan metode K-NN pada sistem klasifikasi genre lagu dengan fitur akustik. Data yang digunakan pada perhitungan manual lebih sedikit dari data yang digunakan pada sistem karena perhitungan manual digunakan sebagai contoh ketika metode K-NN diimplementasi ke dalam sistem. Tahap-tahap perhitungan manual pada metode K-NN adalah sebagai berikut:

1. Data Lagu

Pada data latih yang digunakan untuk perhitungan manual diambil data sebanyak sepuluh data. Terdapat empat fitur yaitu, *zero crossing rate*, *spectral centroid*, *spectral rolloff*, dan *energy* yang datanya diambil dari lima data pertama yang mewakili data sebenarnya. Setiap fitur memiliki tabel masing-masing yang dijelaskan pada Tabel 4.1 - Tabel 4.4

Tabel 4.1 Data lagu fitur *zero crossing rate*

Song	ZCR				
Hero	0,021	0,037	0,050	0,050	0,050
Girl Like You	0,015	0,022	0,025	0,023	0,022



Song	ZCR				
	Help Me Out	0,018	0,028	0,038	0,035
Wait	0,013	0,018	0,024	0,034	0,051
Best 4 You	0,042	0,063	0,081	0,079	0,067
Smells Like Teen Spirit	0,139	0,188	0,236	0,192	0,184
Drain You	0,111	0,179	0,229	0,203	0,216
Come As You Are	0,090	0,163	0,230	0,270	0,247
Something In The Way	0,023	0,033	0,042	0,042	0,043
You Know You're Right	0,103	0,155	0,203	0,204	0,213

Tabel 4.2 Data lagu fitur *spectral centroid*

Song	Spectral Centroid				
	Hero	1441,056	1262,156	1155,438	1181,168
Girl Like You	1251,674	1311,670	1297,721	1090,631	1076,892
Help Me Out	1675,649	1747,812	1756,071	1679,403	1662,743
Wait	2517,244	2373,082	2092,948	1963,064	2113,694
Best 4 You	1577,391	1623,575	1794,005	1828,258	1695,868
Smells Like Teen Spirit	3000,978	2827,812	2482,592	2196,810	1931,556
Drain You	3327,882	3349,865	3356,282	2768,872	2338,150



Song	Spectral Centroid				
Come As You Are	2185,195	2388,878	2704,531	2684,599	2623,862
Something In The Way	1293,408	1349,336	1379,688	1408,640	1454,667
You Know You're Right	3013,414	3130,038	3129,597	2761,643	2829,292

Tabel 4.3 Data lagu fitur *spectral rolloff*

Song	Spectral Rolloff				
Hero	2810,083	2131,787	1981,055	2142,554	2185,620
Girl Like You	2605,518	2799,316	2863,916	1647,290	2293,286
Help Me Out	3434,546	3563,745	3940,576	3919,043	3875,977
Wait	66752,930	6459,961	5803,198	5404,834	5717,065
Best 4 You	2217,920	2637,817	3348,413	3488,379	3090,015
Smells Like Teen Spirit	5835,498	5609,399	5092,603	4414,307	4005,176
Drain You	6459,961	6395,361	6481,494	5953,931	5200,269
Come As You Are	4931,104	4984,937	5189,502	5060,303	4974,170
Something In The Way	2357,886	2497,852	2594,751	2605,518	2605,518
You Know You're Right	6384,595	6427,661	6384,595	5889,331	6029,297

**Tabel 4.4 Data lagu fitur energy**

Song	Energy				
	Hero	20,158	33,529	42,695	40,562
Girl Like You	84,338	84,846	86,849	82,480	80,164
Help Me Out	17,973	16,067	14,493	13,165	13,776
Wait	182,856	173,941	161,176	117,875	100,427
Best 4 You	33,679	31,861	28,127	32,517	41,037
Smells Like Teen Spirit	1,266	2,247	2,692	3,454	6,082
Drain You	7,551	7,221	6,135	5,849	6,182
Come As You Are	50,314	29,406	20,715	19,447	19,373
Something In The Way	15,199	16,089	18,291	16,893	16,879
You Know You're Right	2,141	2,223	2,461	2,182	2,033

Tabel 4.5 Data lagu kelas genre

Song	Genre
Hero	Pop
Girl Like You	Pop
Help Me Out	Pop
Wait	Pop
Best 4 You	Pop
Smells Like Teen Spirit	Rock
Drain You	Rock
Come As You Are	Rock



Song	Genre
Something In The Way	Rock
You Know You're Right	Rock

Pada data latih yang digunakan untuk perhitungan manual ditentukan dua genre sebagai kelas, yaitu genre Pop dan genre Rock. Kedua genre ini dipilih dikarenakan memiliki tingkat kemiripan yang kecil, sehingga memudahkan perhitungan manual dikarenakan menggunakan sedikit data.

2. Data Uji

Data uji yang digunakan adalah sebuah lagu dengan judul "Sugar". Terdapat empat fitur yang digunakan yaitu, *zero crossing rate*, *spectral centroid*, *spectral rolloff*, dan *energy*. Pada perhitungan manual ini setiap fitur hanya diambil lima data pertama untuk memudahkan perhitungan.

Tabel 4.6 Data uji fitur *zero crossing rate*

Song	ZCR				
Sugar	0,050	0,069	0,083	0,067	0,071

Tabel 4.7 Data uji fitur *spectral centroid*

Song	Spectral Centroid				
Sugar	2528,107	2542,152	2573,283	2575,352	2508,931

Tabel 4.8 Data uji fitur *spectral rolloff*

Song	Spectral Rolloff				
Sugar	5534,033	5620,166	5598,633	5587,866	5404,834

Tabel 4.9 Data uji fitur *energy*

Song	Energy				
Sugar	19,482	35,963	64,814	74,449	59,654



3. Perhitungan *normalized cross correlation*

Tabel 4.10 Data hasil perhitungan NCC

Song	ZCR	Spectral Centroid	Spectral Rollof	Energy	SUM
Hero	0,988	0,996	0,991	0,981	0,031
Girl Like You	0,999	0,997	0,984	0,927	0,092
Help Me Out	0,992	1,000	0,998	0,883	0,119
Wait	0,916	0,995	0,594	0,849	0,562
Best 4 You	0,995	0,999	0,988	0,921	0,093
Smells Like Teen Spirit	0,999	0,988	0,991	0,928	0,426
Drain You	0,996	0,991	0,998	0,888	0,123
Come As You Are	0,974	0,997	1,000	0,723	0,280
Something In The Way	0,993	0,999	0,999	0,946	0,055
You Know You're Right	0,990	0,999	0,999	0,933	0,069

Pada penelitian klasifikasi genre lagu dengan fitur akustik menggunakan metode K-NN digunakan persamaan *normalized cross correlation* sebagai pengganti dari persamaan *euclidean distance*. Persamaan NCC memiliki hasil yang baik untuk mengukur kemiripan diantara dua buah lagu. Normalisasi z-score akan dilakukan untuk setiap fitur sebelum digunakan persamaan *normalized cross correlation*. Terakhir untuk menyesuaikan dengan metode K-NN maka nilai 4 akan dikurangi dengan data yang telah dijumlah.

Contoh perhitungan dari persamaan NCC pada penelitian klasifikasi genre lagu dengan fitur akustik menggunakan metode K-NN adalah:



$$SC = \frac{1441,1 * 2528,1 + 1262,2 * 2542,2 + 1155,4 * 2573,3 + 1181,2 * 2575,4 + 1169,6 * 2508,9}{\sqrt{(1441,1^2 + 1262,2^2 + 1155,4^2 + 1181,2^2 + 1169,6^2) * (2528,1^2 + 2542,2^2 + 2573,3^2 + 2575,4^2 + 2508,9^2)}}$$

$$SC = \frac{15801246,56}{\sqrt{7767721,1 * 32402814,3}}$$

$$SC = \frac{15801246,56}{15864930,64}$$

$$SC = 0,9959858583$$

$$SR = \frac{2810,1 * 5534 + 2131,8 * 5620,2 + 1981,1 * 5598,6 + 2142,6 * 5587,9 + 2185,6 * 5404,8}{\sqrt{(2810,1^2 + 2131,8^2 + 1981,1^2 + 2142,6^2 + 2185,6^2) * (5534,0^2 + 5620,2^2 + 5598,6^2 + 5587,9^2 + 5404,8^2)}}$$

$$SR = \frac{62408987,64}{\sqrt{25733572,58 * 153992615,4}}$$

$$SR = \frac{62408987,64}{62950616,72}$$

$$SR = 0,9913959686$$

$$E = \frac{20,159 * 19,482 + 33,529 * 35,963 + 42,695 * 64,814 + 40,562 * 74,449 + 34,833 * 59,654}{\sqrt{(20,159^2 + 33,529^2 + 42,695^2 + 40,562^2 + 34,833^2) * (19,482^2 + 35,963^2 + 64,814^2 + 74,449^2 + 59,654^2)}}$$

$$E = \frac{9465,06}{\sqrt{6212,98 * 14974,74}}$$

$$E = \frac{9465,06}{9645,6}$$

$$E = 0,9812818095$$

$$ZCR = \frac{0,021 * 0,05 + 0,037 * 0,069 + 0,05 * 0,083 + 0,05 * 0,067 + 0,05 * 0,071}{\sqrt{(0,021^2 + 0,037^2 + 0,05^2 + 0,05^2 + 0,05^2) * (0,05^2 + 0,069^2 + 0,083^2 + 0,067^2 + 0,071^2)}}$$

$$ZCR = \frac{0,015}{\sqrt{0,009 * 0,237}}$$

$$ZCR = \frac{0,015}{0,014}$$

$$ZCR = 0,9875762579$$

$$Res = 4 - (0,9875762579 + 0,9959858583 + 0,9913959686 + 0,9812818095)$$

$$Res = 4 - 3,956$$

$$Res = 0,044$$



4. Mengambil K pertama

Tabel 4.11 Data tetangga terdekat

Song	Jarak	Genre
Hero	0,044	Pop
Something In The Way	0,063	Rock
You Know You're Right	0,079	Rock
Girl Like You	0,093	Pop
Smell Like Teen Spirit	0,094	Rock

Pada perhitungan manual, K yang digunakan adalah sebesar 5. K ini diambil dari setengah total data latih. Penentuan K tidak memiliki persamaan pasti karena penentuan kelas dapat dipengaruhi oleh data dan fitur yang digunakan.

5. Penentuan Kelas Mayoritas

Mayoritas kelas yang ada pada tetangga terdekat adalah kelas dari data tersebut. Dari hasil perhitungan manual diatas maka genre dari lagu "Sugar" adalah Rock dikarenakan kelas dengan jumlah terbanyak adalah Rock yang berjumlah 3.

4.3 Perancangan Pengujian

Dalam perancangan pengujian dilakukan pengujian untuk mengetahui seberapa baik sistem bekerja. Pada penelitian klasifikasi lagu dengan fitur akustik menggunakan metode K-NN terdapat tiga pengujian yang akan dilakukan yaitu, pengujian durasi, pengujian jumlah tetangga (K), dan pengujian *offset*. Pada pengujian akan dilakukan evaluasi *confusion matrix multiclass problem* untuk mencari nilai dari *macro average precision*, *macro average recall*, dan *macro average f-measure*. Dipilihnya *macro average scores* sebagai pengujian karena pada penelitian ini hasil pengujian yang dicari adalah *precision*, *recall*, dan *f-measure*. Ketika pengujian yang digunakan adalah akurasi maka *micro average scores* lebih tepat digunakan, karena *precision*, *recall*, dan *f-measure* pada *micro average scores* akan memiliki nilai yang sama (Asch, 2013).

4.3.1 Pengujian Durasi

Pada pengujian durasi akan ditentukan durasi lagu yang diambil ketika sistem melakukan ekstraksi fitur dengan nilai 10, 20 dan 30 detik. Tabel dari pengujian durasi ditunjukkan pada Tabel 4.12 di bawah ini.

**Tabel 4.12 Pengujian durasi**

Durasi	Precision	Recall	F-Measure
10			
20			
30			

4.3.2 Pengujian *Offset*

Pengujian *Offset* dilakukan untuk menguji pada waktu mana sebaiknya data lagu diambil. Pada pengujian ini digunakan *offset* 30, 60, dan 120. Tabel yang menunjukkan pengujian *offset* akan ditunjukkan pada Tabel 4.13.

Tabel 4.13 Pengujian *offset*

Offset	Precision	Recall	F-Measure
30			
60			
90			
120			
150			

4.3.3 Pengujian Jumlah Tetangga

Pengujian jumlah tetangga akan menguji jumlah pengambilan data sebanyak K pada metode K-NN yang akan menentukan hasil akhir dari sebuah sistem. Pada penelitian ini digunakan K sebanyak 5, 10, dan 15. Tabel yang menunjukkan pengujian jumlah tetangga ditunjukkan pada Tabel 4.14 di bawah ini.

Tabel 4.14 Pengujian jumlah tetangga

K	Precision	Recall	F-Measure
5			
10			
15			



4.4 Implementasi

Pada tahap implementasi algoritme dilakukan pengimplementasian algoritme yang mengacu dari perancangan algoritme pada subbab 4.1. Implementasi ini dikerjakan dengan bahasa pemrograman Python pada IDE Pycharm *Community Edition*. Terdapat dua *file* yang dibuat pada implementasi algoritme yang disesuaikan dengan kegunaannya di setiap proses, yaitu *Feature Extraction.py* dan *Main.py*.

4.2.1 Implementasi Ekstraksi Fitur

Ekstraksi Fitur digunakan untuk mengekstraksi data berdasarkan fitur yang berasal dari file lagu. Terdapat empat fitur yang diekstraksi menggunakan *library* Librosa pada proses ini yaitu, *zero crossing rate*, *spectral centroid*, *spectral rolloff*, dan *energy*. Digunakan pula *library* Pickle di akhir proses untuk menyimpan data dengan format biner.

	Feature Extraction.py
1	<code>import librosa, numpy, pickle</code>
2	<code>from os import listdir</code>
3	<code>from os.path import isfile, join</code>
4	
5	<code>tp='latih'</code>
6	<code>dur=10</code>
7	<code>offs=30</code>
8	<code>file=[f for f in listdir('lagu/'+tp) if isfile(join('lagu/'+tp, f))]</code>
9	
10	<code>def extraction(song, offset, duration):</code>
11	<code> hop_length = 256</code>
12	<code> frame_length = 512</code>
13	<code> x, sr = librosa.load(song, offset=offset, duration=duration)</code>
14	<code> zeros = librosa.feature.zero_crossing_rate(x)[0]</code>
15	<code> spec_centroid = librosa.feature.spectral_centroid(x, sr=sr)[0]</code>
16	<code> spec_rolloff=librosa.feature.spectral_rolloff(x + 0.01, sr=sr)[0]</code>
17	<code> energy = numpy.array([sum(x[i:i + frame_length] ** 2) i in</code>
18	<code> range(0, len(x), hop_length)])</code>
19	<code> return [zeros, spec_centroid, spec_rolloff, energy, song[22:-4]]</code>
20	<code>result=[extraction('lagu/'+tp+'/' + x, offs, dur) for x in file]</code>
21	<code>Pickle.dump(result, open('databin/'+tp+str(offs)+str(dur+'.p'), 'wb'))</code>

Baris 1-3 pada program dilakukan import *library* Librosa untuk pengolahan suara, *library* Numpy untuk pengolahan data, *library* Pickle untuk menyimpan dan memanggil file dengan format biner, dan *library* os untuk mengambil informasi pada sebuah direktori. Pada baris 5-8 terdapat 3 variabel input yaitu, *tp* yang digunakan untuk menentukan data digunakan sebagai data latih atau data uji, *dur* yang digunakan untuk menentukan durasi lagu yang diambil, dan *offs* yang digunakan untuk menentukan waktu awal (detik) pengambilan data. Kemudian terdapat variabel *file* yang digunakan untuk mengambil nama-nama lagu pada sebuah direktori. Baris 10-18 merupakan fungsi yang digunakan untuk ekstraksi fitur bernama *extraction* dengan 3 parameter yaitu *song*, *offset*, dan *duration*. Baris 11 dan 12 dilakukan inisialisasi variabel *hop_length* dan variabel



frame_length yang akan digunakan sebagai parameter dalam ekstraksi fitur *energy*, baris 13 digunakan untuk mengubah file lagu menjadi data yang disimpan pada variabel *x* yang menyimpan *time series* dalam bentuk *numpy.ndarray* dan variabel *sr* yang menyimpan nilai *sample rate* dari lagu. Ekstraksi fitur dilakukan pada baris 14-17 untuk mengambil nilai *zero crossing rate*, *spectral centroid*, *spectral rolloff*, dan *energy*. Pada baris 18 fungsi *extraction* akan mengembalikan list berisi nilai *zero crossing rate*, *spectral centroid*, *spectral rolloff*, *energy*, dan nama lagu. Lalu pada baris 19 akan dipanggil fungsi *extraction* sebanyak file lagu yang ada di direktori dan disimpan pada variabel *result*. Setelahnya akan dilakukan fungsi *dump* pada variabel *result* untuk disimpan sebagai file biner menggunakan fungsi *pickle.dump*.

4.2.2 Implementasi Perhitungan K-NN

Pada tahap implementasi perhitungan K-NN terdapat tiga proses utama yaitu, proses *preprocessing*, proses *cross correlation*, dan proses K-NN.

1. Preprocessing

	Main.py
1	<code>def zscore(train,test):</code>
2	<code>norm=np.concatenate((train,test))</code>
3	<code>temp=[np.concatenate((x),axis=None) for x in np.transpose</code>
4	<code>(norm)]</code>
5	<code>mean=[np.mean(x) for x in temp[:4]]</code>
6	<code>std=[np.std(x) for x in temp[:4]]</code>
7	<code>form=lambda a,i:(a-mean[i])/std[i]</code>
8	<code>train=[[form(b[a],a) for a in range(4)]+[b[4]] for b in train]</code>
9	<code>test=[[form(b[a],a) for a in range(4)]+[b[4]] for b in test]</code>
	<code>return train,test</code>

Pada proses *preprocessing* akan dilakukan normalisasi menggunakan persamaan *z-score*. Baris pertama dalam program merupakan fungsi *z-score* yang dengan parameter *train* dan *test*. Kemudian di baris ke-2 parameter *train* dan *test* akan digabung dan disimpan pada variabel *norm*. Pada baris ke-3 list akan digabung kembali sesuai fiturnya dan disimpan pada variabel *temp*. Variabel *mean* pada baris 4 menyimpan nilai rata-rata dari variabel *temp* dan variabel *std* pada baris 5 menyimpan nilai standar deviasi dari variabel *temp*. Variabel *form* pada baris 6 menyimpan fungsi persamaan *z-score* menggunakan fungsi *lambda*. Baris 7 dan 8 dilakukan normalisasi pada variabel *train* dan *test* yang kemudian kedua variabel tersebut akan dikembalikan pada baris 9.

2. Cross Correlation

	Main.py
1	<code>def cros_corr(train,test):</code>
2	<code>return sum([a*b for a,b in zip(train,test)])/np.sqrt(sum([a**2</code>
3	<code>for a in train]) * sum([a**2 for a in test]))</code>
4	<code>def init(uji):</code>
5	<code>a=[[cros_corr(x[i],uji[i]) for i in range(4)] for x in data]</code>
6	<code>a=[[4-sum(x),y[4]] for x,y in zip(a,data)]</code>



Terdapat dua fungsi pada proses *Cross Correlation*, fungsi *cross-corr* pada baris 1 dan 2 dengan parameter *train* dan *test* yang mengembalikan nilai dari hasil persamaan *normalized cross correlation* dan fungsi *init* dengan parameter uji pada baris 4-6 yang melakukan fungsi *cross_corr* pada seluruh fitur lagu dan diulang sebanyak data yang ada pada variabel *data*, kemudian hasil yang didapat akan disimpan pada variabel *a*. Pada baris ke 6 dilakukan perulangan pada variabel *a* dan variabel *data*, pada proses ini disimpan dua data yaitu nilai 4 akan dikurangi nilai dari keempat fitur dan nama lagu dari setiap data.

3. Proses K-NN

```

Main.py
1 a=[a[i]+[kelas[x[1]]] for i,x in enumerate(a) if x[1] in kelas]
2 a.sort(key=lambda x:x[0])
3
4 res = {'Hip Hop':[0,0], 'Pop':[0,0], 'Electronic':[0,0], 'Rock'[0,0],
5 'Jazz':[0,0]}
6 for x in a[:K]:
7     if x[2] in res:
8         res[x[2]][0]+=1
9         res[x[2]][1]+=x[0]
10
11 try:
12     return [mode([x[2] for x in a[:K]])]
13 except:
14     return [min([v[1],k] for k,v in res.items() if v[0] == max([v[0]
15 for v in res.values()]))[1]]

```

Proses Proses K-NN diawali dengan baris 1 yang menambahkan genre pada setiap data di variabel *a*, lalu variabel *a* akan diurutkan berdasarkan nilai hasil perhitungan K-NN yang berada pada *index 0*. Pada baris 4 terdapat *dictionary* *res* yang menyimpan jumlah lagu dan total nilai K-NN pada setiap genre, kedua nilai ini didapat dari perulangan pada baris 5-8 dengan cara menambahkan jumlah lagu dan total nilai pada *key* yang sesuai dengan genre yang ada pada suatu *index*. Perulangan ini akan dilakukan sebanyak *K* atau tetangga terdekat yang telah ditentukan. Untuk mengembalikan nilai dilakukan fungsi *try* yang kemudian mengembalikan genre dengan jumlah terbesar pada variabel *a* dengan panjang *K*. Ketika fungsi ini mengalami *error* dikarenakan terdapat lebih dari satu genre dengan nilai terbesar, maka akan dijalankan fungsi *except* yang mengembalikan genre dengan nilai K-NN terkecil dari seluruh genre dengan jumlah terbesar.

Gambar 4.6 menunjukkan hasil program dari sistem yang telah dibangun. Nilai yang digunakan adalah *durasi* bernilai 10, *offset* bernilai 120, dan tetangga terdekat (*K*) bernilai 10.

```

[arza@Arza-Hev Skripsi]$ /usr/bin/python /home/arza/Skripsi/Main.py
duration:10      offset:120      K:10

Song             : Lonely
Actual           : Jazz
Predicted        : Jazz

```

Gambar 4.6 Hasil Program



BAB 5 HASIL DAN PEMBAHASAN

Pada bab ini akan dilakukan pengujian berdasarkan perancangan pengujian yang telah dijelaskan pada bab 4.1.3. Terdapat tiga hal yang diuji pada penelitian ini yaitu pengujian *offset*, pengujian durasi, dan pengujian tetangga terdekat. Evaluasi yang digunakan adalah *confusion matrix multiclass problem* untuk menentukan *macro average precision*, *macro average recall*, dan *macro average f-measure*. Pada setiap pengujian diberikan standar nilai durasi, *offset* dan tetangga terdekat (K) yang digunakan yaitu durasi bernilai 10, *offset* bernilai 120, dan K bernilai 10.

5.1 Evaluasi *Confusion Matrix*

Setelah sistem berhasil diimplementasi maka akan dilakukan evaluasi *confusion matrix multiclass problem*. Evaluasi dilakukan untuk mencari nilai *precision*, *recall*, dan *f-measure* dari setiap pengujian yang telah ditentukan pada Subbab 4.1.3. Data yang digunakan pada subbab ini adalah data yang memiliki hasil keluaran terbaik yaitu ketika nilai durasi, *offset* dan tetangga terdekat (K) yang digunakan bernilai 10, 120, dan 10. Evaluasi *confusion matrix multiclass problem* akan ditunjukkan pada Tabel 5.1.

Tabel 5.1 *Confusion matrix*

Actual	Predicted				
	Rock	Hip Hop	Jazz	Pop	Electronic
Rock	6	0	0	0	0
Hip Hop	2	3	0	1	0
Jazz	1	0	5	0	0
Pop	0	2	1	2	1
Electronic	1	0	0	2	3

Pada Tabel 5.1 genre vertikal adalah genre aktual dan genre horizontal adalah genre prediksi. Dalam tabel ini ditunjukkan genre Rock adalah genre terbaik yang berhasil diprediksi sistem dengan semua lagu diprediksi benar, sedangkan genre Pop adalah genre yang paling sulit untuk diprediksi oleh sistem dengan dua lagu berhasil diprediksi.

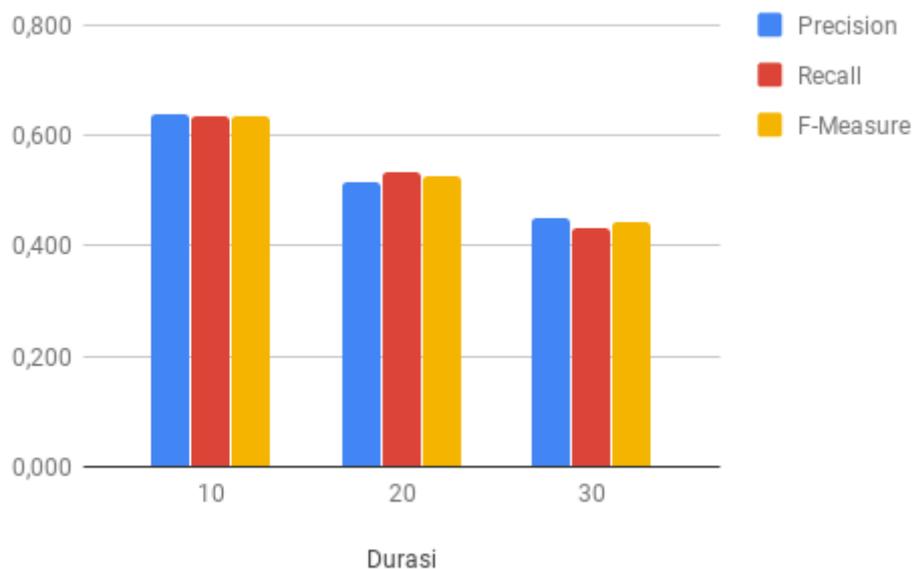


5.2 Pengujian Durasi

Tabel 5.2 Hasil pengujian durasi

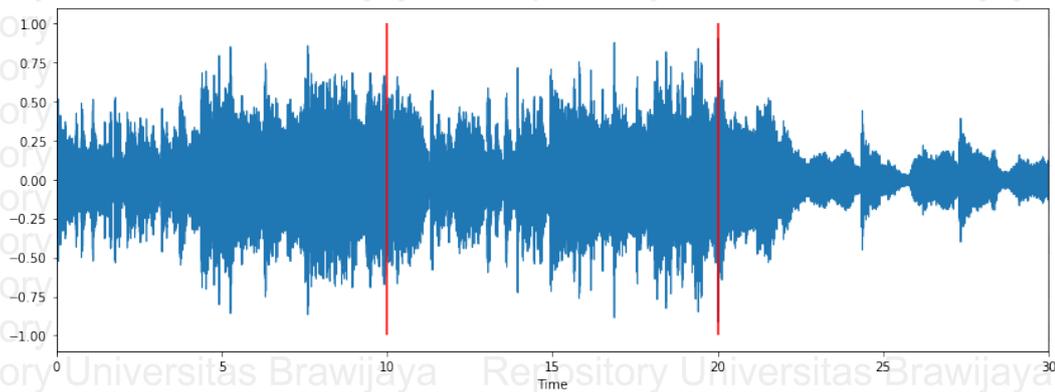
Durasi	Precision	Recall	F-Measure
10	0,637	0,633	0,635
20	0,517	0,533	0,525
30	0,450	0,433	0,442

Pada Tabel 5.2 ditunjukkan hasil pengujian durasi dengan nilai *offset* 120 dan nilai tetangga terdekat (K) 10. Hasil tertinggi diperoleh ketika data yang diambil berdurasi 10 detik dengan *macro average precision* bernilai 0,637, *macro average recall* bernilai 0,633, dan *macro average f-measure* bernilai 0,635. Hasil dari pengujian durasi disertakan pula dalam Gambar 5.1 untuk *macro average* yang menunjukkan penurunan nilai di setiap penambahan nilai pada durasi.



Gambar 5.1 Evaluasi *macro average* durasi

Gambar 5.1 menunjukkan evaluasi terbaik didapat ketika durasi bernilai 10 detik. Grafik menunjukkan semakin besar durasi yang digunakan maka nilai evaluasi yang didapat akan semakin kecil. Penyebab penurunan akurasi ini akan dijelaskan dengan contoh pada Gambar 5.2.



Gambar 5.2 *Waveform* sebuah lagu berdurasi 30 detik

pada Gambar 5.2 terlihat gambar *waveform* yang menunjukkan durasi sebuah lagu dengan genre Rock terbagi menjadi tiga bagian. Bagian pertama adalah lagu yang dimulai dari 10 detik, bagian kedua lagu yang dimulai pada detik ke-20, dan bagian ketiga adalah lagu yang dimulai pada detik ke-30. Ketika data diambil dari bagian satu maka data terlihat masih memiliki pola, lalu ketika bagian satu dan dua digunakan maka sinyal pada lagu semakin bervariasi, sehingga ketika data diambil dengan durasi 30 detik maka sinyal pada lagu akan lebih bervariasi. Berdasarkan analisis tersebut, semakin besar durasi yang digunakan maka sistem akan semakin sulit menemukan lagu yang bergenre sama dengan data uji.

5.3 Pengujian *Offset*

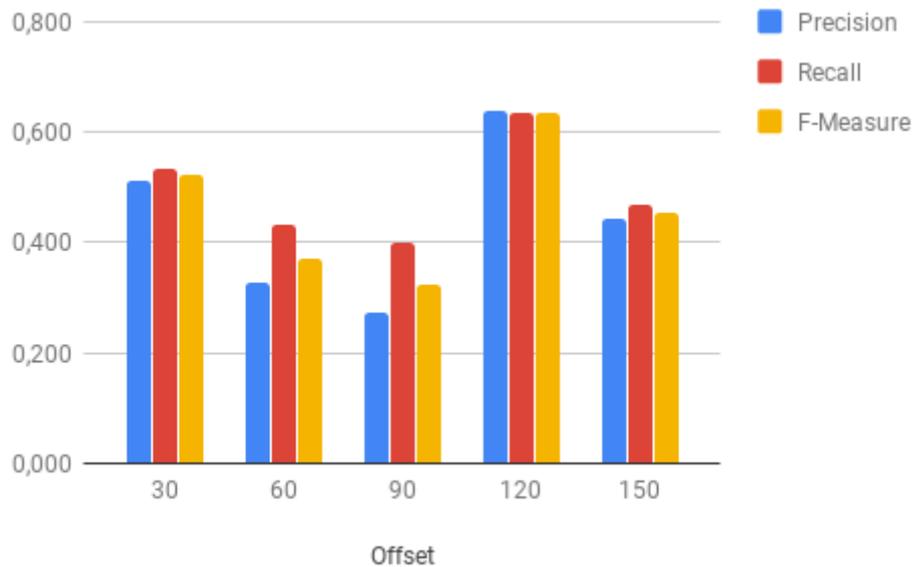
Tabel 5.3 Hasil pengujian *offset*

Offset	Precision	Recall	F-Measure
30	0,512	0,533	0,523
60	0,326	0,433	0,372
90	0,272	0,400	0,324
120	0,637	0,633	0,635
150	0,467	0,442	0,454

Pada Tabel 5.3 ditunjukkan hasil pengujian *offset* dengan nilai durasi 10 dan nilai K 10. Hasil tertinggi diperoleh ketika data diambil pada detik ke 120 dengan *macro average precision* bernilai 0,637, *macro average recall* bernilai 0,633, dan *macro average f-measure* bernilai 0,635. Hasil dari pengujian durasi disertakan pula dalam Gambar 5.3 untuk *macro average* yang memperlihatkan kurva



dengan nilai terendah pada *offset* 90 dalam grafik tersebut, ini membuktikan bahwa hasil yang didapatkan sistem dapat membaik pada *offset* tertentu.



Gambar 5.3 Evaluasi *macro average offset*

Dapat dilihat pada gambar 5.3 evaluasi terbaik berada pada *offset* bernilai 120. Ini dikarenakan pada lagu terdapat *refrain* yang merupakan bagian yang diulang dalam sebuah lagu. Lagu lebih mudah diprediksi karena pada *refrain* genre sebuah lagu lebih dapat terlihat. Umumnya *refrain* berada di tengah lagu dan pada penelitian ini rata-rata lagu memiliki durasi sekitar 4 menit, ini membuktikan kenapa *offset* bernilai 120 memiliki nilai tertinggi.

5.4 Pengujian *Tetangga Terdekat*

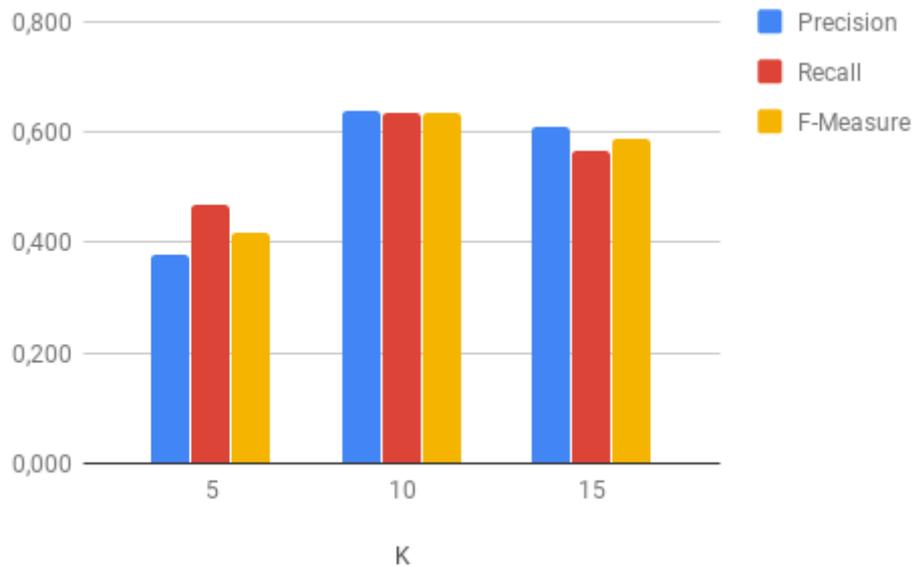
Tabel 5.4 Hasil pengujian jumlah tetangga

K	Precision	Recall	F-Measure
5	0,376	0,467	0,416
10	0,637	0,633	0,635
15	0,608	0,567	0,587

Pada Tabel 5.4 ditunjukkan hasil pengujian tetangga terdekat (K) dengan nilai *offset* 120, durasi 10 dan nilai K berupa 5,10, dan 15. Hasil tertinggi diperoleh ketika sistem mengambil 10 tetangga terdekat dengan *macro average precision* bernilai 0,637, *macro average recall* bernilai 0,633, dan *macro average f-measure* bernilai 0,635. Hasil dari pengujian durasi disertakan pula Gambar 5.4



untuk *macro average* yang menunjukkan nilai tertinggi berada pada tetangga terdekat (K) 10.



Gambar 5.4 Evaluasi *macro average* tetangga terdekat

Penentuan nilai *optimum* tetangga terdekat (K) pada K-NN selalu berbeda pada setiap penelitian, hal ini bergantung pada dataset yang digunakan dalam penelitian tersebut. Ketika tetangga terdekat (K) memiliki nilai terlalu tinggi maka sistem memiliki ruang lingkup yang terlalu luas sehingga kelas yang salah atau noise masuk ke dalam ruang lingkup dan biasa disebut *overfitting*, sedangkan jika nilai tetangga terdekat (K) terlalu rendah maka sistem memiliki ruang lingkup yang terlalu sempit sehingga kelas yang benar tidak dapat masuk ke dalam ruang lingkup dan biasa disebut *underfitting*. Untuk mengetahuinya maka harus dilakukan pengujian terlebih dahulu. Dari Gambar 5.4 yang menunjukkan hasil evaluasi pada pengujian tetangga terdekat (K) didapatkan akurasi tertinggi ketika tetangga terdekat (K) bernilai 10. Berdasarkan analisis tersebut, nilai optimum tetangga terdekat (K) pada penelitian Klasifikasi Genre Lagu dengan Fitur Akustik Menggunakan Metode *K-Nearest Neighbor* adalah 10.

Hasil klasifikasi terbaik yang didapatkan pada penelitian Klasifikasi Lagu dengan Fitur Akustik Menggunakan Metode K-NN adalah ketika *offset*, durasi, dan tetangga terdekat (K) berturut-turut bernilai 120,10, dan 10. Sistem berhasil memprediksi 19 lagu dengan genre Rock terprediksi sebanyak 6 lagu, genre Hip hop sebanyak 3 lagu, genre Jazz sebanyak 5 lagu, genre Pop sebanyak 2 lagu, dan genre Electronic sebanyak 3 lagu. Ditunjukkan 10 lagu dari hasil klasifikasi terbaik pada Tabel 5.5 dengan 7 lagu yang berhasil diklasifikasi.



Tabel 5.5 Hasil klasifikasi

No	Artis	Judul Lagu	Kelas Aktual	Kelas Prediksi
1	Led Zeppelin	Black Dog	Rock	Rock
2	Aerosmith	Beautiful	Rock	Rock
3	George Benson	Affirmation	Jazz	Jazz
4	Daft Punk	Within	Electronic	Electronic
5	Cardi B	Best Life	Hip Hop	Hip Hop
6	Tom Waits	Lonely	Jazz	Jazz
7	Depeche Mode	Cover Me	Electronic	Pop
8	Nujabes	Kumomi	Hip Hop	Hip Hop
9	Justin Timberlake	Higher Higher	Pop	Hip Hop
10	Frank Sinatra	Where Are You	Pop	Jazz



BAB 6 PENUTUP

6.1 Kesimpulan

Berdasarkan penerapan dan pengujian yang telah dilakukan pada penelitian Klasifikasi Genre Lagu dengan Fitur Akustik Menggunakan Metode *K-Nearest Neighbor* maka dapat diambil kesimpulan:

1. Penelitian ini berhasil merancang sistem klasifikasi genre lagu menggunakan metode *K-Nearest Neighbor* yang penghitungan jaraknya diganti dengan penghitungan *Normalized Cross Correlation*. Penelitian ini menggunakan data hasil ekstraksi fitur yaitu *zero crossing rate*, *spectral centroid*, *spectral rolloff*, dan *energy* yang didapat dari *file* lagu yang telah dilakukan proses *preprocessing* menggunakan normalisasi *z-score*. Proses yang dijalankan pada sistem adalah ekstraksi fitur, normalisasi data, menjalankan metode K-NN dengan persamaan NCC, dan terakhir mengambil label mayoritas dari tetangga terdekat yang telah ditentukan.
2. Hasil pengujian yang didapat dari penelitian menggunakan evaluasi *confusion matrix multiclass problem* ditunjukkan dengan *macro average scores*. Hasil pengujian terbaik didapat ketika pengujian durasi bernilai 10, *offset* bernilai 120, dan tetangga terdekat (K) bernilai 10 dengan nilai *macro average precision* bernilai 0,637, *macro average recall* bernilai 0,633, dan *macro average f-measure* bernilai 0,635. Hasil ini terbilang kurang baik jika dibandingkan dengan penelitian Tamatjita dan Harjoko (2014) yang memiliki nilai akurasi tertinggi 96,67% dan penelitian Costa et al. (2012) yang memiliki nilai akurasi tertinggi 80,65%.

6.2 Saran

Saran yang dapat diberikan berdasarkan penelitian Klasifikasi Genre Lagu dengan Fitur Akustik Menggunakan Metode *K-Nearest Neighbor* untuk pengembangan klasifikasi genre lagu adalah:

1. Penambahan fitur yang digunakan untuk dapat mengklasifikasikan genre lagu dengan lebih baik sehingga dapat meningkatkan akurasi.
2. Penggunaan genre yang memiliki tipe suara unik seperti genre Rock, tidak seperti genre Pop yang memiliki banyak tipe suara.
3. Penggunaan *refrain* lagu sebagai *dataset* yang digunakan dalam penelitian.
4. Pengembangan sistem menjadi klasifikasi *multilabel* untuk memudahkan sistem mengenali lagu yang memiliki genre lebih dari satu.
5. Penggunaan metode lain seperti SVM, ELM, atau *backpropagation* untuk mengetahui apakah metode klasifikasi lain dapat memberi hasil akurasi lebih baik daripada metode K-NN.



DAFTAR REFERENSI

- Al-Qutt, M.M., Hamad, A.M., Salem, M.A., dan Aziz, M.H.A., 2011, Performance Comparison for Feature Selection in Musical Information Retrieval, *Ain Shams University*
- Apriansyah, A., Ilhamsyah dan Rismawan, T., 2016. Prototype Kunci Otomatis Pada Pintu Berdasarkan Suara Pengguna Menggunakan Metode K-NN (K-Nearest Neighbor). *Jurnal Coding, Sistem Komputer Untan*, 04(1), pp.45–56.
- Asch, V. V., 2013, Macro- and Micro-Averaged Evaluation Measures [[BASIC DRAFT]], [online] Tersedia di: <<https://pdfs.semanticscholar.org/1d10/6a2730801b6210a67f7622e4d192bb309303.pdf>> [di akses 23 Juni 2019]
- Costa, Y.M.G., Oliveira, L.S., Koerich, A.L., Gouyon, F. dan Martins, J.G., 2012. Music genre classification using LBP textural features. *Signal Processing*, [online] 92(11), pp.2723–2737. Tersedia di: <<http://dx.doi.org/10.1016/j.sigpro.2012.04.023>>.[di akses 25 Juni 2019]
- Fetra, N. dan Irsyad, M., 2015. Aplikasi Pencarian Chord dalam Membantu Penciptaan Lagu Menggunakan Algoritma Fast Fourier Transform (FFT) dan Metode Klasifikasi K-Nearest Neighbor (K-NN). *Jurnal CoreIT*, 1(2 ISSN: 2460-738X), pp.30–36.
- Guo, Z., Wang, Q., Liu, G., Guo, J. dan Lu, Y., 2012. A music retrieval system using melody and lyric. *Proceedings of the 2012 IEEE International Conference on Multimedia and Expo Workshops, ICMEW 2012*, pp.343–348.
- Kaminskas, M. dan Ricci, F., 2012. Contextual music information retrieval and recommendation: State of the art and challenges. *Computer Science Review*, [online] 6(2–3), pp.89–119. Tersedia di: <<http://dx.doi.org/10.1016/j.cosrev.2012.04.002>>.[di akses 10 Januari 2019]
- Kulkarni, N. dan Bairagi, V., 2018. Use of Complexity Features for Diagnosis of Alzheimer Disease. *EEG-Based Diagnosis of Alzheimer Disease*, pp.47–59.
- Mohabeer, H., Soyjaudah, K.M.S. dan Pavaday, N., 2011. Enhancing The Performance Of Neural Network Classifiers Using Selected Biometric Features. *Engineering*, (c), pp.140–144.



Murugappan, M., 2011. Human emotion classification using wavelet transform and K-NN. *Proceedings of the 2011 International Conference on Pattern Analysis and Intelligent Robotics, ICPAIR 2011*, 1(November), pp.148–153.

Schedl, M., Yang, Y.-H. dan Herrera-Boyer, P., 2016. Introduction to Intelligent Music Systems and Applications. *ACM Transactions on Intelligent Systems and Technology*, 8(2), pp.1–8.

Seppänen, J., 1999. Audio Signal Processing Basics. pp.1–17. *Tampere University of Technology Signal Processing Laboratory* [online] Tersedia di:<<https://www.cs.tut.fi/sgn/arg/intro/basics.html>> [di akses 30 Juni 2019]

Stevetjoa, 2014, Notes on Music Information Retrieval. [online] Tersedia di:<<https://musicinformationretrieval.com>> [di akses 15 Januari 2019]

Tamatjita, N. dan Harjoko, A., 2014. Klasifikasi Lagu Berdasarkan Genre pada Format WAV. *Ijccs*, 8(2), pp.145–154.



LAMPIRAN A HASIL KELUARAN PROGRAM

No	Artis	Judul Lagu	Kelas Aktual	Kelas Prediksi
1	Led Zeppelin	Black Dog	Rock	Rock
2	Aerosmith	Beautiful	Rock	Rock
3	George Benson	Affirmation	Jazz	Jazz
4	Daft Punk	Within	Electronic	Electronic
5	Cardi B	Best Life	Hip Hop	Hip Hop
6	Tom Waits	Lonely	Jazz	Jazz
7	Depeche Mode	Cover Me	Electronic	Pop
8	Nujabes	Kumomi	Hip Hop	Hip Hop
9	Justin Timberlake	Higher Higher	Pop	Hip Hop
10	Frank Sinatra	Where Are You	Pop	Jazz
11	Ray Charles	There's No You	Jazz	Jazz
12	21 Savage	No Peace	Hip Hop	Pop
13	Taylor Swift	Delicate	Pop	Electronic
14	Maroon 5	Bet My Heart	Pop	Hip Hop
15	Frank Sinatra	Witchcraft	Jazz	Rock
16	Green Day	See The Light	Rock	Rock
17	Daft Punk	Beyond	Electronic	Electronic
18	Eminem	Framed	Hip Hop	Rock
19	Alan Walker	Alone	Electronic	Rock
20	Taylor Swift	End Game	Pop	Pop
21	Green Day	21 Guns	Rock	Rock
22	21 Savage	Lord Forgive Me	Hip Hop	Hip Hop



No	Artis	Judul Lagu	Kelas Aktual	Kelas Prediksi
23	George Benson	Lady	Jazz	Jazz
24	Depeche Mode	So Much Love	Electronic	Electronic
25	Ray Charles	Come Rain Or Come Shine	Jazz	Jazz
26	Alan Walker	Faded	Electronic	Pop
27	Cardi B	Bodak Yellow	Hip Hop	Rock
28	Monkey Majik	Fly	Pop	Pop
29	Led Zeppelin	Kashmir	Rock	Rock
30	Nirvana	Drain You	Rock	Rock



LAMPIRAN B TABEL CONFUSION MATRIX

B.1 Confusion Matrix Durasi 10, Offset 120, K 10

Predicted Actual	Rock	Hip Hop	Jazz	Pop	Electronic
Rock	6	0	0	0	0
Hip Hop	2	3	0	1	0
Jazz	1	0	5	0	0
Pop	0	2	1	2	1
Electronic	1	0	0	2	3

B.2 Confusion Matrix Durasi 20, Offset 120, K 10

Predicted Actual	Rock	Hip Hop	Jazz	Pop	Electronic
Rock	6	0	0	0	0
Hip Hop	2	1	0	3	0
Jazz	1	0	5	0	0
Pop	1	0	1	1	3
Electronic	2	1	0	0	3

B.3 Confusion Matrix Durasi 30, Offset 120, K 10

Predicted Actual	Rock	Hip Hop	Jazz	Pop	Electronic
Rock	5	0	1	0	0
Hip Hop	2	1	1	0	2
Jazz	1	0	5	0	0



Predicted Actual	Rock	Hip Hop	Jazz	Pop	Electronic
Pop	1	1	1	2	1
Electronic	3	1	2	0	0

B.4 Confusion Matrix Durasi 10, Offset 30, K 10

Predicted Actual	Rock	Hip Hop	Jazz	Pop	Electronic
Rock	6	0	0	0	0
Hip Hop	3	3	0	0	0
Jazz	0	0	6	0	0
Pop	1	3	2	0	0
Electronic	3	0	2	0	1

B.5 Confusion Matrix Durasi 10, Offset 60, K 10

Predicted Actual	Rock	Hip Hop	Jazz	Pop	Electronic
Rock	6	0	0	0	0
Hip Hop	2	1	0	2	1
Jazz	1	0	5	0	0
Pop	0	3	2	0	1
Electronic	2	0	3	0	1

B.6 Confusion Matrix Durasi 10, Offset 90, K 10

Predicted Actual	Rock	Hip Hop	Jazz	Pop	Electronic
------------------	------	---------	------	-----	------------



Rock	6	0	0	0	0
Hip Hop	3	1	0	2	0
Jazz	1	0	5	0	0
Pop	2	1	1	0	2
Electronic	3	1	2	0	0

B.7 Confusion Matrix Durasi 10, Offset 150, K 10

Predicted Actual	Rock	Hip Hop	Jazz	Pop	Electronic
Rock	6	0	0	0	0
Hip Hop	2	1	0	2	1
Jazz	0	0	3	0	3
Pop	1	2	1	2	0
Electronic	2	0	2	0	2

B.8 Confusion Matrix Durasi 10, Offset 120, K 5

Predicted Actual	Rock	Hip Hop	Jazz	Pop	Electronic
Rock	6	0	0	0	0
Hip Hop	2	3	0	1	0
Jazz	2	0	4	0	0
Pop	0	2	1	1	2
Electronic	1	2	2	1	0



B.9 Confusion Matrix Durasi 10, Offset 120, K 15

Predicted Actual	Rock	Hip Hop	Jazz	Pop	Electronic
Rock	5	0	1	0	0
Hip Hop	2	1	0	0	3
Jazz	1	0	5	0	0
Pop	0	3	1	2	0
Electronic	1	1	0	0	4