

# Identifikasi Genre Musik dengan Menggunakan Metode Random Forest

Muhammad Abid As Sarofi, Irhamah dan Adatul Mukarromah  
Departemen Statistika, Institut Teknologi Sepuluh Nopember (ITS)  
*e-mail*: irhamah@statistika.its.ac.id

**Abstrak**—Genre musik merupakan pengelompokan musik sesuai dengan kemiripan antara satu musik dengan musik yang lainnya. Hal yang paling penting dalam pengidentifikasian musik adalah pengelompokan genre musik. Pengelompokan tersebut dilakukan secara manual pada umumnya dengan mendengarkan secara langsung lagu tersebut. Namun, hal tersebut dapat menimbulkan ketidakefisienan. Oleh karena itu, dilakukan penelitian yang bertujuan untuk mengidentifikasi sebuah lagu dengan menggunakan metode Random Forest dengan data yang digunakan adalah GTZAN dataset yang diperoleh dari laman MARSYAS. Metode supervised learning yang digunakan yaitu Random Forest karena metode tersebut lebih baik dalam hal mengklasifikasikan data karena bersifat robust terhadap outliers dan noise. Fitur ekstraksi yang digunakan dalam penelitian ini adalah MFCC karena mampu mengadaptasi pendengaran manusia. Model yang digunakan untuk identifikasi genre musik memiliki performa klasifikasi yang tinggi dengan penggunaan KCV untuk pembagian data training dan testing.

**Kata Kunci**—KCV, MFCC, Genre Musik, GTZAN dataset, Random Forest

## I. PENDAHULUAN

MUSIK adalah bagian dari karya seni yang menjadikan berbagai macam suara kedalam beberapa pola yang dapat dimengerti dan dipahami oleh manusia [1]. Musik dapat juga didefinisikan sebagai suatu karya seni dalam bentuk lagu atau komposisi musik yang mengungkapkan pikiran dan perasaan penciptanya melalui unsur-unsur musik seperti irama, melodi, harmoni, bentuk atau struktur lagu dan ekspresi sebagai satu kesatuan [2]. Musik dikelompokkan kedalam beberapa *genre* musik. *Genre* musik diartikan sebagai pengelompokan musik berdasarkan kemiripan jenis, tipe, atau kelompok sastra serta atas dasar bentuknya [3]. Berkembangnya *volume* musik yang kini beredar terorganisir dalam *database music digital* [4] dimana kategori yang paling umum digunakan untuk mengorganisir musik adalah *genre* musik tersebut.

Identifikasi *genre* pada musik dilakukan secara manual oleh seorang ahli [5], namun hal tersebut menimbulkan beberapa persoalan diantaranya adalah adanya ketidak efisienan dalam pengidentifikasian *genre* musik secara manual karena memerlukan waktu dan tenaga ahli yang memahami musik agar tidak terdapat kesalahan dalam melakukan identifikasi tersebut. Pengidentifikasian *genre* musik secara otomatis dapat membantu, mengurangi atau mengganti peran manusia dalam pemberian *genre* pada sebuah musik.

Data audio yang akan diidentifikasi genre musiknya perlu dilakukan fitur ekstraksi terlebih dahulu. Fitur ekstraksi yang digunakan dalam penelitian ini adalah *Mel Frequency*

*Cepstrum Coefficient* (MFCC), karena MFCC mampu mengadaptasi pendengaran manusia dalam bentuk digital. Keunggulan penggunaan MFCC sebagai fitur ekstraksi adalah kemampuan dalam mengenali karakteristik suara yang sangat penting bagi pengenalan suara dan menghasilkan data seminimal mungkin tanpa menghilangkan informasi penting [6]. Penggunaan MFCC sebagai fitur ekstraksi memberikan hasil yang lebih baik dibandingkan dengan fitur ekstraksi lainnya seperti *Linear Predictive Coefficient* (LPC) dan *Zero Crossing Rate* (ZCR) [7]. Kebaikan klasifikasi yang diperoleh dengan menggunakan MFCC sebagai fitur ekstraksi mampu mencapai 100%. Apabila dibandingkan dengan *Learning Vector Quantization* (LVQ), MFCC tetap memberikan performa klasifikasi yang lebih unggul [8] dengan nilai akurasi sebesar 93,75% pada data musik dengan *genre rock*, klasik, keroncong dan jazz.

Pengidentifikasian *genre* musik memerlukan sebuah metode yang mampu menangkap karakteristik dari musik tersebut. Metode yang digunakan untuk mengidentifikasi sebuah data adalah *supervised learning* salah satunya adalah *Random Forest*. *Random Forest* dipilih dalam penelitian ini karena memiliki keunggulan relatif *robust* terhadap *outliers* dan *noise*. Metode *Random Forest* menunjukkan performa terbaik [9] apabila dibandingkan dengan metode klasifikasi berbasis *decision tree* lainnya seperti, ADTree, LADTree, C4.5, CART, *Random Tree*, REPTree dan BFTree dengan nilai akurasi pada metode *Random Forest* sebesar 96,65% dalam hal mengklasifikasikan *file audio*. Oleh karena itu, penelitian ini bertujuan untuk mengidentifikasi genre musik dengan genre musik yang digunakan terbatas hanya menggunakan lima genre yaitu *disco*, *hiphop*, *jazz*, *pop* dan *reggae* pada sebuah sistem *Graphic User Interface*.

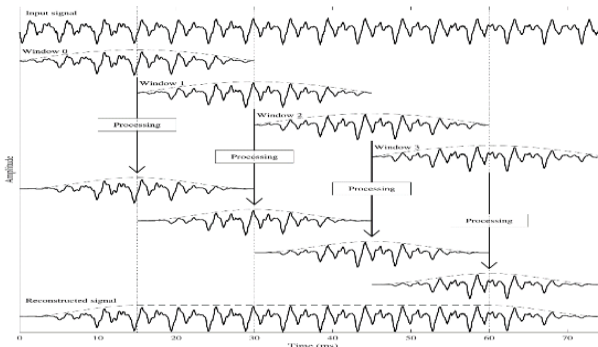
## II. TINJAUAN PUSTAKA

### A. Audio Mining

*Audio mining* adalah sebuah teknik dalam menganalisis konten sinyal *audio* secara otomatis yang umum digunakan pada bidang pengenalan suara otomatis salah satunya adalah pengenalan *genre* musik. Pengenalan konten sinyal *audio* memerlukan fitur ekstraksi baik berbasis waktu seperti *energy*, *Zero Crossing Rate* (ZCR) dan *entropy of energy* ataupun berbasis frekuensi seperti *spectral centroid and speed*, *spectral entropy*, *spectral flux*, *spectral rolloff*, MFCC dan *chroma vector* [10].

### B. Mel Frequency Cepstrum Coefficient(MFCC)

MFCC adalah salah satu teknik fitur ekstraksi yang dapat menyerupai pendengaran manusia dalam bentuk digital. Keunggulan dari MFCC adalah mampu menangkap karakteristik suara yang sangat penting bagi pengenalan sinyal suara, menghasilkan data seminimal mungkin tanpa



Gambar 1. Ilustrasi Framing pada MFCC

menghilangkan informasi yang penting dalam mengenali suara. Tahapan dalam MFCC adalah sebagai berikut [11]

C. Pre-Emphasis

Pre-emphasis adalah teknik dalam melakukan penekanan pada sinyal yang dapat memperkuat frekuensi yang tinggi. Tahapan tersebut menyeimbangkan spectrum frekuensi karena frekuensi yang tinggi memiliki magnitude yang lebih kecil dengan menggunakan persamaan (1).

$$y(g) = x(g) - \alpha x(g - 1) \tag{1}$$

dimana:

- $y(g)$  : nilai amplitudo hasil pre-emphasis
  - $x(g)$  : nilai amplitudo awal pada waktu ke-g
  - $\alpha$  : nilai koefisien filter
- dengan  $\alpha$  / koefisien filter yang digunakan bernilai 0,97.

D. Framing

Framing adalah tahap membagi sinyal kedalam beberapa frame yang lebih kecil. Framing dilakukan karena frekuensi dalam sinyal berubah dari waktu ke waktu. Ilustrasi dari tahapan framing seperti pada Gambar 1.

E. Window

Window adalah tahapan yang dilakukan setelah sinyal dibagi kedalam beberapa frame. Fungsi yang diterapkan pada tahapan ini adalah Hamming Window pada setiap frame. Persamaan Hamming Window seperti pada persamaan (2).

$$w[n] = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right) \tag{2}$$

dimana:

- $w[n]$  : nilai amplitudo hasil windowing pada window length ke-n dengan,  $0 \leq n \leq N - 1$ ,  $N$  adalah panjang window

F. Fourier-Transform dan Power Spectrum

Tahapan Fourier-Transform (FFT) mengaplikasikan  $N$ -point FFT pada setiap frame untuk menghitung spectrum frekuensi atau juga disebut sebagai Short-Time Fourier-Transform (STFT), dengan nilai  $N$  sebesar 512. Perhitungan FFT seperti pada persamaan (3).

$$FFT(x_k) = \sum_{n=0}^{N-1} x_n e^{-\frac{i2\pi kn}{N}} \tag{3}$$

dimana:

- $x_k$  : nilai FFT ke-k dimana,  $0 \leq k \leq \frac{N}{2} + 1$
- $x_n$  : nilai amplitudo hasil windowing ke-n

Selanjutnya dapat dihitung power spectrum (peridogram) pada setiap frame dengan menggunakan persamaan (4).

$$P = \frac{|FFT(x_i)|^2}{N} \tag{4}$$

dimana  $x_i$  adalah frame ke-i pada sinyal  $x$ .

G. Filter Banks

Filter banks adalah tahapan dalam MFCC yang mengaplikasikan triangular filters, dengan nilai filter sebanyak 40 pada skala mel ke power spectrum untuk mengekstraksi frequency bonds. Hal tersebut dilakukan dengan cara mengkonversi antara Hertz ( $f$ ) dan Mel ( $m$ ) menggunakan persamaan (5).

$$m = 2595 \log_{10}\left(1 + \frac{f_{sr}}{700}\right) \tag{5}$$

dimana  $f_{sr}$  adalah setengah dari nilai sample rate. Selanjutnya dapat juga dilakukan konversi dari nilai mel ke hertz dengan menggunakan persamaan (6).

$$f = 700(10^{m/2595} - 1) \tag{6}$$

dimana  $m$  adalah nilai mel scale. Selanjutnya mengaplikasikan filter bank ke dalam power spectrum dengan menggunakan persamaan (7) dan mengkonversi kedalam nilai desibel dengan persamaan (8).

$$H_m(k) = \begin{cases} 0, & k < f(m-1) \\ \frac{k-f(m-1)}{f(m)-f(m-1)}, & f(m-1) \leq k < f(m) \\ 1, & k = f(m) \\ \frac{f(m+1)-k}{f(m+1)-f(m)}, & f(m) < k \leq f(m+1) \\ 0, & k > f(m+1) \end{cases} \tag{7}$$

$$f_{banks}(k) = 20 \log_{10} H_m(k) \tag{8}$$

H. MFCC

Tahapan terakhir dalam MFCC adalah menerapkan discrete cosinus transform (DCT) pada koefisien filter bank menggunakan persamaan (9) dan (10).

$$y(k) = f_{sf} \left( 2 \sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi k(2n+1)}{2N}\right) \right) \tag{9}$$

$$f_{sf} = \begin{cases} \sqrt{\frac{1}{4N}}, & k = 0 \\ \sqrt{\frac{1}{2N}}, & k \text{ lainnya} \end{cases} \tag{10}$$

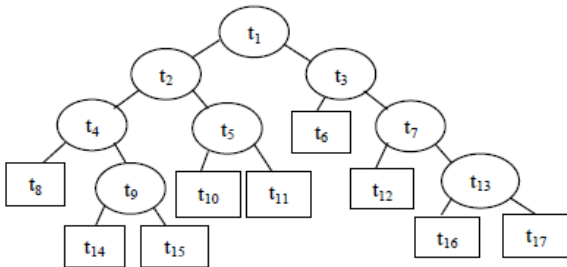
dimana:

- $y(k)$  : nilai MFCC ke-k
- $f_{sf}(k)$  : scaling factor

I. CART

CART (Classification and Regression Tree) adalah pendekatan model nonparametrik yang dapat menjelaskan variabel respon yang dipengaruhi oleh variabel prediktor yang bersifat kontinu maupun kategorik. Data dependent tergantung dari partisi serangkaian node yang bercabang ke kanan dan ke kiri dapat disebut simpul anak (child nodes) yang berasal dari simpul utama (parent node). Setelah partisi telah berhenti, child nodes disebut sebagai terminal nodes [12].

Ilustrasi pohon klasifikasi dapat dilihat pada Gambar 2. Simpul awal yang merupakan variabel terpenting dalam menduga kelas amatan disebut sebagai simpul utaman (parent node) dengan notasi  $t_1$ , simpul dalam (internal



Gambar 2. Ilustrasi Pohon Klasifikasi

nodes) dinotasikan sebagai simpul akhir (*terminal nodes*) dinotasikan dengan  $t_6, t_8, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_{16}$  dan  $t_{17}$  dimana setelahnya tidak ada lagi pemilihan. Setiap simpul berada pada kedalaman (*depth*) tertentu dimana  $t_1$  berada pada kealaman 1,  $t_2$  dan  $t_3$  berada pada kedalaman 2, dan begitu seterusnya hingga  $t_{14}, t_{15}, t_{16}$  dan  $t_{17}$  yang berada pada kedalaman 5.

Proses pembentukan pohon klasifikasi terdiri atas 3 tahapan, yaitu pemilihan, penentuan simpul terminal dan penandaan label kelas. Pada tahap pemilihan (*classifier*) data yang digunakan adalah sampel data *training/learning* ( $L$ ) yang dipilih berdasarkan aturan pemilihan dan kriteria *goodness of split*. Himpunan bagian yang dihasilkan dari proses pemilihan harus lebih homogen dibandingkan simpul induknya. Hal ini dapat dilakukan dengan mendefinisikan fungsi keheterogenan simpul (*impurity* atau  $imp(t)$ ). Fungsi heterogenitas yang umum digunakan adalah Indeks Gini.

Metode ini memiliki kelebihan yaitu proses perhitungan yang sederhana dan relatif cepat, serta mudah dan sesuai untuk diterapkan dalam berbagai kasus [13]. Fungsi Indeks Gini dituliskan dalam persamaan berikut.

$$imp(t) = \sum_{i,j=1} p(j|t)p(i|t), i \neq j \tag{11}$$

dengan  $p(j|t)$  adalah proporsi kelas  $j$  pada simpul  $t$  dan  $p(i|t)$  adalah proporsi kelas  $i$  pada simpul  $t$ . Selanjutnya menentukan kriteria *goodness of split* ( $\phi(s, t)$ ) untuk melakukan evaluasi pemilah dari pemilah  $s$  pada simpul  $t$ . *goodness of split* ( $\phi(s, t)$ ) didefinisikan sebagai penurunan heterogenitas sebagai berikut.

$$(\phi(s, t)) = imp(t) - p_L imp(t_L) - p_R imp(t_R) \tag{12}$$

dengan

- $imp(t)$  : fungsi heterogenitas pada simpul  $t$
- $p_L$  : proporsi pengamatan simpul kiri
- $p_R$  : proporsi pengamatan menuju simpul kanan
- $imp(t_L)$ : fungsi heterogenitas pada simpul anak kiri
- $imp(t_R)$ : fungsi heterogenitas pada simpul anak kanan

$$\Delta i(s^*, t_1) = \max_{s \in S} \Delta i(s, t) \tag{13}$$

Pemilah yang menghasilkan  $\phi(s, t)$  lebih tinggi merupakan pemilah terbaik karena mampu mereduksi heterogenitas. Pengembangan pohon dilakukan dengan pencarian pemilah yang mungkin pada simpul  $t_j$  yang kemudian akan dipilih menjadi  $t_2$  dan  $t_3$  oleh pemilah  $s^*$  dan begitu seterusnya.

Tahapan selanjutnya yaitu menentukan simpul terminal. Suatu simpul  $t$  akan menjadi simpul terminal atau tidak, akan dipilih kembali bila pada simpul  $t$  tidak terdapat penurunan keheterogenan secara berarti atau adanya batasan minimum  $n$  seperti halnya terdapat satu pengamatan pada tiap simpul anak. Jumlah minimum dalam suatu terminal akhir umumnya

adalah 5, dan apabila hal itu terpenuhi maka pengembangan pohon dihentikan. Selanjutnya, dilakukan penandaan label kelas pada terminal *nodes* berdasarkan aturan jumlah terbanyak. Label kelas simpul terminal  $t$  adalah  $j_0$  yang memberi nilai dugaan kesalahan pengklasifikasian simpul  $t$  terbesar. Proses pembentukan pohon klasifikasi berhenti saat terdapat hanya satu pengamatan dalam tiap-tiap simpul anak atau adanya batasan minimum  $n$ , semua pengamatan dalam tiap simpul anak identik dan adanya batasan jumlah level/kedalaman pohon maksimal.

$$p(j_0|t) = \max_j p(j|t) = \max_j \frac{N_j(t)}{N(t)} \tag{14}$$

dengan  $N_j(t)$  merupakan banyaknya amatan kelas  $j$  pada terminal *nodes*  $t$  dan  $N(t)$  merupakan jumlah total pengamatan dalam terminal *node*  $t$ . Label kelas untuk terminal *node*  $t$  adalah  $j_0$  yang memberikan nilai dugaan kesalahan pengklasifikasian pada simpul  $t$  paling kecil sebesar  $r(t) = 1 - \max_j p(j|t)$ . Setelah pohon klasifikasi terbentuk, perlu juga dilakukan pemangkasan pohon klasifikasi. Bagian pohon yang kurang penting dilakukan pemangkasan sehingga didapatkan pohon klasifikasi yang optimal. Pemangkasan didasarkan pada suatu penilaian ukuran sebuah pohon tanpa mengorbankan kebaikan ketepatan melalui pengurangan simpul pohon sehingga dicapai ukuran pohon yang layak. Ukuran pemangkasan yang digunakan untuk memperoleh ukuran pohon yang layak tersebut adalah *cost complexity minimum*.

$$R_\alpha(t) = R(t) + \alpha|T| \tag{15}$$

dimana

- $R(t)$  : *resubstitution estimate*
- $\alpha$  : kompleksitas parameter (*complexity parameter*)
- $|T|$  : ukuran banyaknya simpul terminal pohon  $T$

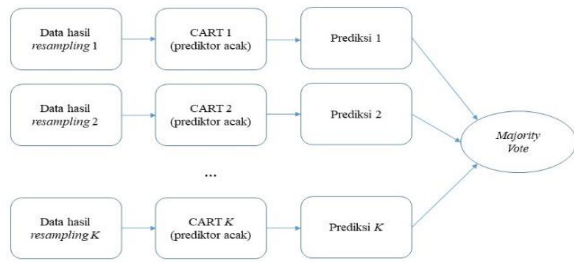
Setelah dilakukan pemangkasan agar dicapai ukuran pohon yang layak, selanjutnya dapat dilakukan penentuan pohon klasifikasi optimal. Ukuran pohon yang terlalu besar akan menyebabkan nilai *cost complexity* yang tinggi karena struktur data yang digambarkan cenderung kompleks sehingga perlu dipilih pohon optimal yang berukuran sederhana tetapi memberikan nilai penduga pengganti yang cukup kecil. Bila  $R(T)$  dipilih sebagai penduga terbaik, maka akan cenderung dipilih pohon yang besar, sebab pohon yang semakin besar akan membuat nilai  $R(T)$  semakin kecil.

**J. Random Forest (RF)**

*Random Forest* adalah gabungan dari pohon klasifikasi (CART) yang saling independen yang berasal dari distribusi yang sama melalui proses *voting* (jumlah terbanyak) untuk memperoleh prediksi klasifikasi. *Random forest* memiliki karakteristik yang dapat meminimumkan korelasi yang dapat menurunkan hasil kesalahan prediksi *random forest* [14].

Pada *random forest* pemilihan pemilah hanya melibatkan beberapa variabel prediktor yang terambil secara acak. Algoritma *random forest* dijelaskan sebagai berikut.

- a. Mengambil  $n$  data sampel dari *dataset* awal dengan menggunakan teknik *resampling bootstrap* dengan pengambilan.
- b. Menyusun pohon klasifikasi dari setiap *dataset* hasil *resampling bootstrap*, dengan penentuan pemilah terbaik didasarkan pada variabel prediktor yang diambil secara acak. Jumlah variabel yang diambil secara acak dapat ditentukan melalui perhitungan  $\log_2(Z + 1)$  dimana  $Z$  adalah banyaknya variabel prediktor atau  $\sqrt{Z}$



Gambar 3. Algoritma Random Forests

- c. Melakukan prediksi klasifikasi data sampel berdasarkan pohon klasifikasi yang terbentuk.
- d. Mengulangi prediksi klasifikasi data sampel berdasarkan pohon klasifikasi yang terbentuk. Pengulangan dilakukan sebanyak  $K$  kali.
- e. Melakukan prediksi klasifikasi data sampel akhir dengan mengkombinasikan hasil prediksi pohon klasifikasi yang diperoleh berdasarkan aturan *majority vote*.

Secara garis besar algoritma *random forests* tersebut dapat dijelaskan pada Gambar 3.

**K. Evaluasi Hasil Klasifikasi**

Tahapan evaluasi adalah tahapan untuk mengetahui tingkat akurasi dan kinerja dari hasil klasifikasi. Pengukuran ketepatan klasifikasi dilakukan untuk melihat performa klasifikasi yang telah dilakukan. Dalam mengukur ketepatan klasifikasi, perlu diketahui jumlah pada setiap kelas prediksi dan kelas aktual yang terdiri dari TP (*True Positive*), TN (*True Negative*), FP (*False Positive*), dan FN (*False Negative*). Nilai TP dan TN didefinisikan dengan  $n_{11}$ ,  $n_{22}$ ,  $n_{33}$ ,  $n_{44}$  dan  $n_{55}$  pada Tabel 1, sedangkan sisanya merupakan FP dan FN. Untuk kasus *multiclass*, *Confusion matrix* yang terbentuk seperti pada Tabel 1.

Pada kasus *multiclass* kinerja klasifikasi dapat diukur dengan menggunakan *accuracy* dan *Fscore*. *Accuracy* adalah banyak pengamatan yang terklasifikasi secara tepat sedangkan *Fscore* didapatkan dari nilai kombinasi antara *precision* dan *sensitivity*. *Precision* adalah banyaknya pengamatan yang tepat terprediksi positif dari keseluruhan dengan hasil prediksi positif, *sensitivity* adalah banyaknya pengamatan yang tepat diklasifikasikan sesuai kategorinya [15]. Perhitungan *accuracy*, *precision*, *sensitivity*, dan *Fscore* dapat dilakukan dengan menggunakan persamaan (16), (17), (18), dan (19).

$$accuracy = \frac{\sum_{k=1}^K TP_k + TN_k}{\sum_{k=1}^K TP_k + FN_k + FP_k + TN_k} \quad (16)$$

$$Fscore = \frac{2(precision \times sensitivity)}{precision + sensitivity} \quad (17)$$

$$sensitivity = \frac{\sum_{k=1}^K TP_k}{\sum_{k=1}^K TP_k + FN_k} \quad (18)$$

$$precision = \frac{\sum_{k=1}^K TP_k}{\sum_{k=1}^K TP_k + FP_k} \quad (19)$$

**L. K-fold Cross Validation (KCV)**

*K-fold cross validation* adalah salah satu metode yang digunakan untuk mempartisi data menjadi data *training* dan

Tabel 1. *Confusion Matrix* untuk *Multiclass*

Kelas Aktual	Kelas Prediksi					Total
	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$	
$C_1$	$n_{11}$	$n_{12}$	$n_{13}$	$n_{14}$	$n_{15}$	$n_{1.}$
$C_2$	$n_{21}$	$n_{22}$	$n_{23}$	$n_{24}$	$n_{25}$	$n_{2.}$
$C_3$	$n_{31}$	$n_{32}$	$n_{33}$	$n_{34}$	$n_{35}$	$n_{3.}$
$C_4$	$n_{41}$	$n_{42}$	$n_{43}$	$n_{44}$	$n_{45}$	$n_{4.}$
$C_5$	$n_{51}$	$n_{52}$	$n_{53}$	$n_{54}$	$n_{55}$	$n_{5.}$
Total	$n_{.1}$	$n_{.2}$	$n_{.3}$	$n_{.4}$	$n_{.5}$	$N_{..}$

Tabel 2. Variabel Penelitian

Variabel	Keterangan	Skala
$X_{i,n}$	Nilai MFCC	Interval
$Y$	Genre musik yang digunakan: $Y_1$ : Genre Disco $Y_2$ : Genre Hiphop $Y_3$ : Genre Jazz $Y_4$ : Genre Pop $Y_5$ : Genre Reggae	Nominal

data *testing*. Metode ini banyak digunakan peneliti karena dapat mengurangi bias yang terjadi dalam pengambilan sampel. *K-fold cross validation* secara berulang-ulang membagi data menjadi data *training* dan data *testing*, dimana setiap data mendapat kesempatan menjadi data *testing* [16].  $K$  merupakan besar angka partisi data yang digunakan untuk pembagian *training* dan *testing*.

**III. METODOLOGI PENELITIAN**

**A. Sumber Data**

Pada penelitian ini data yang digunakan adalah data sekunder, yaitu GTZAN dataset diambil dari *Music Analysis, Retrieval and Synthesis for Audio Signals* (MARSYAS) dengan data file berformat wav (*waveform audio file*). Data genre sebanyak 10 lagu. Spesifikasi file audio yang digunakan adalah memiliki format wav channel mono dengan frame rates 22050Hz. Contoh gambaran data yang digunakan tercantum pada Gambar 4. audio yang digunakan yaitu data audio dengan genre musik disco, hiphop, jazz, pop dan reggae dengan masing-masing.

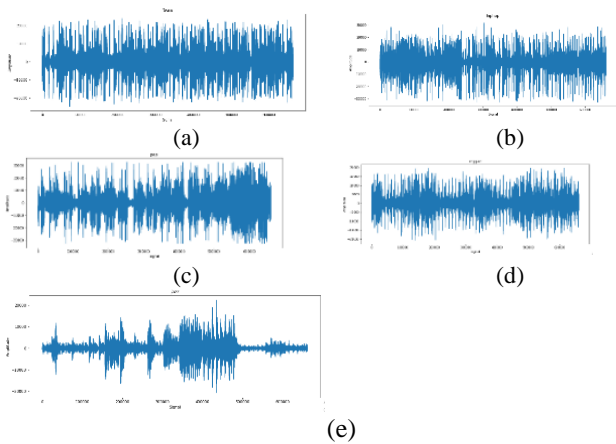
**B. Variabel Penelitian**

Variabel penelitian yang digunakan tercantum dalam Tabel 2.

**C. Langkah Analisis**

Langkah-langkah penelitian yang telah dilakukan berdasarkan dengan tujuan penelitian ini adalah sebagai berikut.

1. Mengumpulkan data audio musik pada laman MARSYAS.
2. Preprocessing data. Tahap preprocessing data meliputi:



Gambar 4. Contoh Gambaran Data Audio (a) Genre Disco (b) Genre Hip-hop (c) Genre Pop (d) Genre Reggae (e) Genre Jazz

- a. Mengimport data audio untuk mendapatkan nilai amplitudo.
- b. Melakukan tahapan *pre-emphasis* pada masing-masing file audio.
- c. Membagi sinyal ke dalam beberapa *frame* yang lebih kecil dengan rentang 25 ms untuk ukuran *frame* dan 10ms untuk ukurn *stride*.
- d. Menerapkan *window* pada setiap *frame*.
- e. Mengaplikasikan *N-point FFT* pada setiap *frame* untuk menghitung *spectrum* frekuensi dengan nilai  $N = 512$
- f. Mengaplikasikan *triangular filters*, dengan nilai *filter* sebanyak 40 pada skala *mel* ke *power spectrum* untuk mengekstraksi *frequency bonds*.
- g. Menerapkan *DCT* untuk menghitung nilai MFCC
3. Membagi data *training* dan data *testing* dengan menggunakan *K-folds cross validation*.
4. Melakukan klasifikasi dengan menggunakan metode *Random Forest* (RF)
5. Membuat *graphic user interface* (GUI) dengan menggunakan metode *random forest* terbaik pada pembagian *training* dan *testing* yang telah didapatkan.
6. Mengambil kesimpulan dan saran.

#### IV. ANALISIS DAN PEMBAHASAN

##### A. Karakteristik Data Hasil Preprocessing

Data hasil praproses yang didapatkan dari praproses dengan MFCC adalah 12 koefisien MFCC, dimana masing-masing koefisien MFCC menjelaskan hal yang berbeda. Koefisien pertama dalam MFCC merepresentasikan rata-rata *power* dalam *spectrum*. Koefisien kedua pada MFCC merepresentasikan *spectral centroid*, koefisien ketiga hingga ketujuh merepresentasikan *spectral centroid* yang lebih detail dan koefisien kedelapan hingga koefisien kedua belas merepresentasikan bentuk dari *spectrum*. Dari hasil praproses tersebut kemudian dibuat *spectrogram* MFCC. Bentuk *spectrogram* dari salah satu *audio* masing-masing kelas yang digunakan terdapat pada Tabel 3.

Tabel 4 diatas dapat dilihat bahwa *spectrogram* yang tergambaran antara *genre* musik pada kelas  $Y_1$  (*disco*),  $Y_2$  (*hiphop*),  $Y_4$  (*pop*), dan  $Y_5$  (*reggae*) tidak jauh berbeda yaitu terdapat warna gelap yang menggambarkan semakin keras

Tabel 3.  
*Spectrogram* Salah Satu Audio Setiap Kelas

$Y_k$	Waveform Audio	<i>Spectrogram</i>
$Y_1$		
$Y_2$		
$Y_3$		
$Y_4$		
$Y_5$		

suatu suara pada setiap beberapa *ms*, sedangkan pada  $Y_3$  (*jazz*) yang digunakan sebagai pembeda benar-benar menggambarkan *spectrogram* yang berbeda diantara yang lain yaitu warna gelap pada MFCC cenderung lebih memusat di beberapa durasi audio.

##### B. Klasifikasi Menggunakan RF

Metode klasifikasi yang digunakan selanjutnya adalah metode klasifikasi *Random Forest*. Sebelum dilakukan analisis dengan menggunakan metode *random forest*, perlu dilakukan analisis *CART* karena metode *CART* sendiri merupakan metode yang mendasari dari analisis *random forest*. Pada analisis *CART* digunakan data sebanyak 14.095 *frame* audio dengan pembagian data *training* dan *testing* menggunakan *KCV* dengan nilai  $K$  dari 2 hingga 10.

Analisis *CART* dimulai dengani pembentukan pohon klasifikasi yang maksimal. Berikut merupakan penjelasan untuk masing-masing tahapan analisis klasifikasi *CART* dengan menggunakan kombinasi data *training* dan *testing* menggunakan *KCV* dengan nilai  $K=2$  pada *fold* 1. Serangkaian tahapan analisis *CART* berikut juga dilakukan untuk pembagian data *training* dan *testing* yang lain yaitu dengan menggunakan nilai  $K$  dari 2 hingga 10 pada masing-masing *fold*. Tahapan awal dalam membentuk pohon klasifikasi adalah dengan menentukan variabel pemilah dan nilai variabel (*threshold*). Variabel pemilah dan *threshold* dipilih dari beberapa kemungkinan pemilah dari masing-masing variabel. Perhitungan banyaknya kemungkinan pemilah ditampilkan pada Tabel 4.

Dari berbagai kemungkinan pemilah dari tiap variabel, selanjutnya dihitung indeks gini yang merupakan ukuran keheterogenan simpul. Indeks gini lebih sering digunakan karena alasan kesederhanaan dalam proses perhitungan. Cara kerja indeks gini adalah melakukan pemilihan simpul dengan berfokus pada masing-masing simpul kanan atau kiri. Hasil perhitungan indeks gini kemudian digunakan untuk menentukan *goodness of split* dari masing-masing pemilah. Pemilah yang terpilih adalah variabel pemilah dan nilai variabel (*threshold*) yang memiliki nilai *goodness of split* tertinggi. Contoh perhitungan indeks gini pada variabel MFCC dengan menggunakan salah satu *threshold* seperti

Tabel 4.

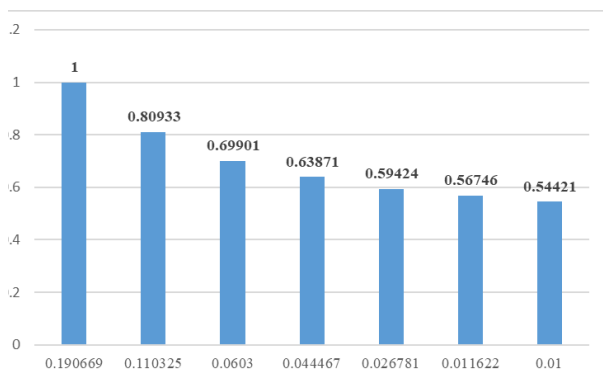
Perhitungan Kemungkinan Jumlah Pemilahan dari Setiap Variabel	
Variabel	Kemungkinan Pemilahan
X <sub>1</sub>	7475-1=7474
X <sub>2</sub>	7475-1=7474
X <sub>3</sub>	7475-1=7474
X <sub>4</sub>	7475-1=7474
X <sub>5</sub>	7475-1=7474
X <sub>6</sub>	7475-1=7474
X <sub>7</sub>	7475-1=7474
X <sub>8</sub>	7475-1=7474
X <sub>9</sub>	7475-1=7474
X <sub>10</sub>	7475-1=7474
X <sub>11</sub>	7475-1=7474
X <sub>12</sub>	7475-1=7474

Tabel 5.

Ilustrasi Pemilahan pada Sampel MFCC1						
MFCC1	Genre Musik					N
	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>3</sub>	Y <sub>4</sub>	Y <sub>5</sub>	
< 0,105	1457	1329	316	1317	786	5205
≥ 0,015	52	209	1160	150	699	2270

Tabel 6.

Nilai Importance Variable	
Variabel	Skor Variabel
MFCC1	100
MFCC9	56.7855
MFCC5	55.19072
MFCC2	53.56934
MFCC4	51.0028
MFCC7	44.81933
MFCC11	44.09287
MFCC8	42.75049
MFCC6	41.51806
MFCC3	40.72539
MFCC12	30.47991
MFCC10	30.16131



Gambar 5. Nilai Complexity Parameter beserta Error

pada Tabel 5. Selanjutnya melakukan perhitungan untuk nilai indeks gini pada masing-masing simpul kanan dan kiri sebagai berikut.

$$imp(t_L) = 2 \left( \frac{1457}{5205} \times \frac{1329}{5205} \times \dots \times \frac{1317}{5205} \times \frac{786}{5205} \right) = 0,000332$$

$$imp(t_R) = 2 \left( \frac{52}{2270} \times \dots \times \frac{699}{2270} \right) = 0,0000438608$$

Kemudian menentukan kriteria *goodness of split* untuk evaluasi pemilahan yang telah dilakukan oleh pemilahan *s* pada simpul *t*. karena hanya ada satu kemungkinan pemilahan, maka untuk simpul MFCC1 hanya ada satu kriteria *goodness of split*.

$$\phi(s, t) = imp(t) - \left( \frac{5205}{7475} \right) \times imp(t_L) - \left( \frac{2270}{7475} \right) \times imp(t_R)$$

$$\phi(s, t) = 0,000395317$$

Simpul *t* dikatakan sebagai simpul terminal jika tidak terdapat penurunan heterogenitas atau dengan kata lain hanya terdapat satu kelas pada simpul anak.

Besarnya kontribusi variabel sebagai pemilahan baik pemilahan utama maupun pengganti pada pohon klasifikasi maksimal yang terbentuk ditunjukkan melalui suatu angka skor yang ditampilkan pada Tabel 7.

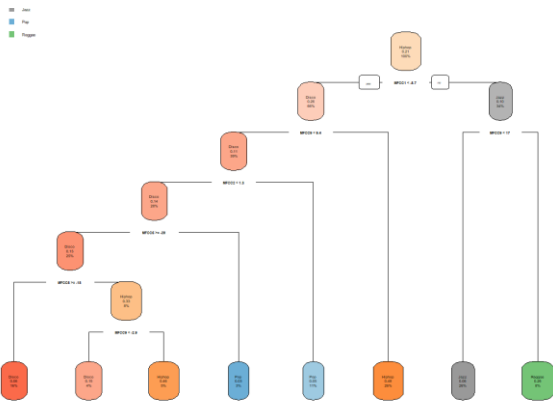
Tabel 6 menunjukkan bahwa semua variabel menjadi pembangun pohon klasifikasi. Akan tetapi, berdasarkan skor yang dihasilkan diketahui bahwa variabel MFCC1 merupakan variabel yang terpenting dan menjadi pemilahan utama dalam mengklasifikasikan musik kedalam beberapa

*genre* musik. Hasil dari pohon klasifikasi maksimal yang berukuran relatif besar tidak dapat ditampilkan dalam laporan ini, mengingat pohon klasifikasi maksimal tersebut sangat besar.

Langkah selanjutnya adalah melakukan pemangkasan pada pohon. Nilai yang digunakan adalah *complexity parameter*. Pemangkasan pohon tersebut bertujuan untuk menghindari adanya *overfitting* yang diakibatkan oleh jumlah pemilahan yang terlalu banyak. Gambar 5 menunjukkan nilai *complexity parameter* dan besar *error*, dimana dari Gambar 5 menunjukkan bahwa parameter *complexity parameter* terbaik adalah sebesar 0,01 karena memiliki *error* terkecil. Parameter tersebut didapatkan pada pembagian *nsplit* sebesar 7. Selanjutnya, dibentuk pohon keputusan dengan menggunakan nilai *complexity parameter* optimum yaitu 0,01 dengan *nsplit* sebesar 7 seperti pada Gambar 6. Gambar 6 merupakan pohon klasifikasi optimum yang menjelaskan bahwa variabel MFCC1 menjadi variabel pertama yang dapat memprediksi sebuah audio kedalam beberapa *genre* musik.

Setelah melakukan analisis dengan menggunakan CART, dapat dilanjutkan dengan melakukan analisis dengan menggunakan *Random Forest*. Dalam penelitian ini parameter kontrol yang ditentukan adalah jumlah pohon yang dibentuk akan dicobakan pada kombinasi 50, 100, 500 dan 1000 pohon melalui proses *grid search* dimana didapatkan satu nilai jumlah pohon terbaik yang akan dilanjutkan dalam proses *Random Forest* yaitu dengan jumlah pohon sebesar 1000. Sehingga, jumlah pohon yang digunakan dalam penelitian ini hanya 1000.

*K-Fold Cross Validation (KCV)* diaplikasikan untuk membagi data *training* dan data *testing* pada metode *Random Forest* dengan nilai *K* dari 2 hingga 10 dan didapatkan hasil kebaikan pada setiap nilai *K* seperti pada Tabel 7. Tabel 7 memperlihatkan bahwa dengan mengaplikasikan *KCV* pada pembagian data *training* dan *testing* dengan nilai *K* dari 2-10, menunjukkan bahwa dengan menggunakan *K* sebesar 9



Gambar 6. CART MFCC Fold 1

Tabel 7. Performa Metode Random Forest setiap K

K	Accuracy		Fscore	
	Training	Testing	Training	Testing
2	0,9961	0,8637	0,9961	0,8636
3	0,9965	0,8756	0,9965	0,8755
4	0,9967	0,8811	0,9967	0,8810
5	0,9969	0,8847	0,9969	0,8847
6	0,9968	0,8858	0,9968	0,8859
7	0,9968	0,8864	0,9968	0,8865
8	0,9968	0,8882	0,9968	0,8881
9	0,9969	0,8883	0,9970	0,8882
10	0,9968	0,8881	0,9968	0,8882
Mean	0,9967	0,8824	0,9967	0,8824

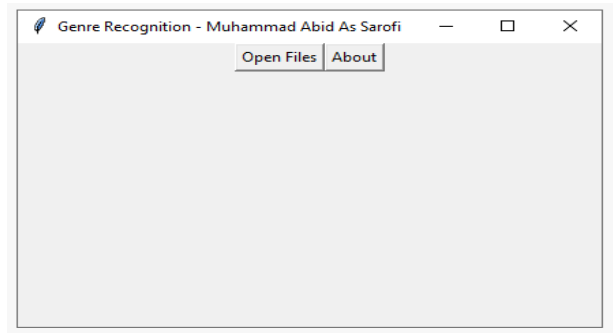
memberikan performa yang baik pada data *training* maupun data *testing* dengan nilai *accuracy*, *sensitivity*, *precision* dan *Fscore* berturut-turut sebesar 0,9969, 0,9969, 0,9970 dan 0,9970 sedangkan pada data *testing* sebesar 0,8883; 0,8882; 0,8883 dan 0,8882. Oleh karena itu, model *Random Forest* terbaik apabila menggunakan *KCV* dengan nilai *K* sebesar 9.

C. Graphic User Interface

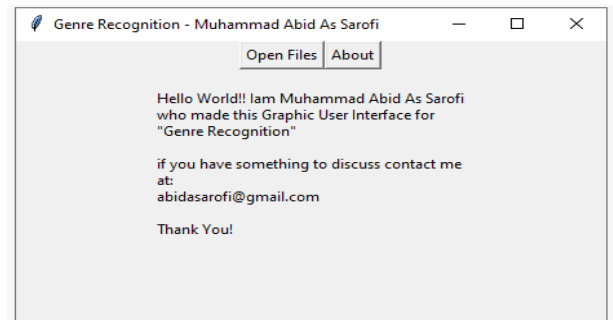
*Graphic user interface* (GUI) adalah sebuah sistem yang dapat membuat para pengguna atau user dapat berinteraksi dengan suatu perangkat komputer yang digunakan oleh pengguna tersebut. GUI pada penelitian ini digunakan untuk mendeteksi *genre* dari sebuah *file audio* musik. *File audio* musik yang digunakan dalam penelitian ini berformat *.wav* saja. Tampilan awal dari GUI pada penelitian ini seperti pada Gambar 7.

Gambar 7 merupakan tampilan awal dalam GUI pada penelitian ini. Terlihat pada bagian judul terdapat tulisan “Genre Recognition – Muhammad Abid As Sarofi”. Terdapat tombol bertuliskan “Open Files” dengan fungsi yaitu membuat *file explorer* untuk memilih lagu yang akan dideteksi *genrenya* dan “About” yang merupakan keterangan dari GUI. Ilustrasi penggunaan tombol “About” seperti pada Gambar 8.

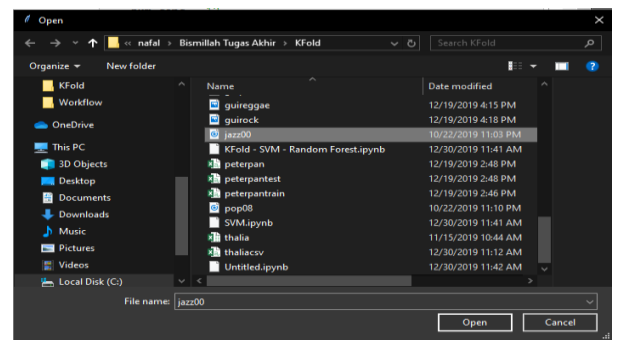
Gambar 8 merupakan *output* apabila tombol “About” ditekan. Keterangan pada Gambar 7 menunjukkan pembuat GUI dan tujuan GUI tersebut juga kontak dari pembuat GUI. Ilustrasi penggunaan tombol “Open Files” seperti pada Gambar 9. Gambar 9 menunjukkan tampilan pada saat “Open Files” ditekan yang akan membawa *user* menuju *file explorer* untuk mengimport lagu yang akan dideteksi *genrenya*.



Gambar 7. Tampilan Awal Graphic User Interface



Gambar 8. Tampilan “About” GUI



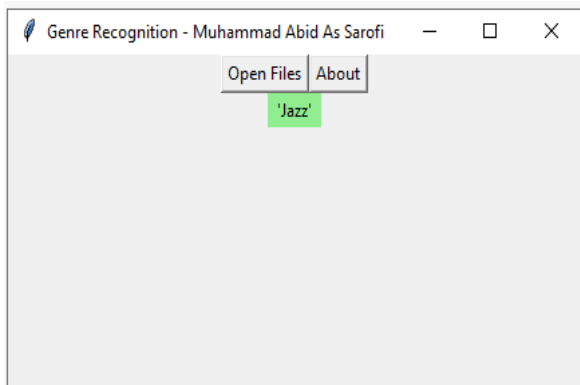
Gambar 9. Tampilan “Open Files” GUI

Contoh yang digunakan adalah dataset dari GTZAN dengan judul “jazz00” yang akan dideteksi *genre* musiknya. Setelah diklik tombol “Open” di *file explorer* seperti pada Gambar 9 maka akan muncul hasil seperti Gambar 10.

Gambar 10 merupakan tampilan hasil deteksi *genre* dari musik yang ingin diketahui *genrenya*. Terdapat tulisan ‘Jazz’ pada GUI tersebut yang menunjukkan bahwa *genre* musik dari *file audio* yang digunakan adalah *genre* musik Jazz.

V. KESIMPULAN DAN SARAN

Kesimpulan yang diperoleh dari penelitian ini adalah berdasarkan hasil deskripsi data hasil fitur ekstraksi data *audio* pada masing-masing *genre* yang digunakan didapatkan hasil bahwa *Mel Frequency Cepstrum Coefficient* (MFCC) pada *genre* musik yang memiliki keterkaitan terlihat sama, sedangkan *genre* musik yang berbeda memang memperlihatkan *spectrogram* yang berbeda. MFCC1 merupakan variabel terpenting dalam mengidentifikasi *genre* musik dan metode yang digunakan pada GUI untuk identifikasi *genre* musik memiliki nilai performansi klasifikasi pada data *training* maupun data *testing* dengan nilai *accuracy* dan *Fscore* berturut-turut sebesar 0,9969 dan 0,9970 sedangkan pada data *testing* sebesar 0,8883 dan 0,8882. Berdasarkan kesimpulan yang diperoleh, dapat



Gambar 10. Tampilan Hasil Deteksi Genre Musik GUI

dirumuskan saran sebagai pertimbangan penelitian selanjutnya adalah menggunakan fitur ekstraksi yang lain seperti berdasarkan *time domain* yaitu *energy*, *zero-crossing rate (ZCR)* dan *entropy of energy* ataupun *frequency domain* yaitu *spectral centroid and speed*, *spectral entropy*, *spectral flux*, *spectral rolloff* dan *chroma vector* agar mendapatkan hasil yang maksimal. Selain itu juga menggunakan metode klasifikasi yang lain agar didapatkan hasil yang lebih baik daripada menggunakan metode yang digunakan dalam penelitian ini, terutama dalam kasus *overfitting* pada metode *random forest* dan menambahkan jumlah *file audio* musik yang digunakan agar hasilnya lebih baik.

#### DAFTAR PUSTAKA

- [1] P. Banoe, *Kamus Musik*. Yogyakarta: Kanisius, 2003.
- [2] Jamalus, *Buku Pengajaran Musik Melalui Pengalaman Musik Panduan Pengajar*. Jakarta: Proyek Pengembangan Lembaga Pendidikan Tenaga Kependidikan, 1988.
- [3] KBBI Kemendikbud, "KBBI Daring," *Website KBBI Kemendikbud*. [Online]. Available: <https://kbbi.kemdikbud.go.id/entri/genre>. [Accessed: 16-Jun-2020].
- [4] L. Nanni, Y. M. G. Costa, A. Lumini, M. Y. Kim, and S. R. Baek, "Combining visual and acoustic features for music genre classification," *Expert Syst. Appl.*, vol. 45, pp. 108–117, 2016, doi: 10.1016/j.eswa.2015.09.018.
- [5] M. Genussov and I. Cohen, "Musical genre classification of audio signals using geometric methods," *Eur. Signal Process. Conf.*, vol. 10, no. 5, pp. 497–501, 2010.
- [6] D. Putra and A. Resmawan, "Verifikasi Biometrika Suara Menggunakan Metode Mfcc Dan Dtw," *Lontar Komput.*, vol. 2, no. 1, pp. 8–21, 2011, doi: 2088-1541.
- [7] M. M. Panchwagh and V. D. Katkar, "Music genre classification using data mining algorithm," *Conf. Adv. Signal Process. CASP 2016*, no. June, pp. 49–53, 2016, doi: 10.1109/CASP.2016.7746136.
- [8] M. R. Fansuri, "Klasifikasi Genre Musik Menggunakan Learning Vector Quantization (LVQ)," Institut Pertanian Bogor, 2011.
- [9] S. V. Thambi, K. T. Sreekumar, C. Santhosh Kumar, and P. C. Reghu Raj, "Random forest algorithm for improving the performance of speech/non-speech detection," *2014 1st Int. Conf. Comput. Syst. Commun. ICCSC 2014*, no. December, pp. 28–32, 2013, doi: 10.1109/COMPSC.2014.7032615.
- [10] T. Giannakopoulos and A. Pirkakis, *Introduction to Audio Analysis: a MATLAB® Approach*. Saint Louis: Elsevier Science, 2014.
- [11] H. Fayek, "Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between," 21-Apr-2016. [Online]. Available: <https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html>. [Accessed: 16-Jun-2020].
- [12] H. Zheng, L. Chen, X. Han, X. Zhao, and Y. Ma, "Classification and regression tree (CART) for analysis of soybean yield variability among fields in Northeast China: The importance of phosphorus application rates under drought conditions," *Agric. Ecosyst. Environ.*, vol. 132, no. 1–2, pp. 98–105, 2009, doi: 10.1016/j.agee.2009.03.004.
- [13] S. Brieman, Friedman, Olshen, *Classification and Regression Trees*. New York: Chapman & Hall, 1993.
- [14] L. Breiman, "Random forests," *Kluwer Acad. Publ.*, vol. 45, pp. 5–32, 2001, doi: 10.1201/9780367816377-11.
- [15] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009, doi: 10.1016/j.ipm.2009.03.002.
- [16] E. Gokgoz and A. Subasi, "Comparison of decision tree algorithms for EMG signal classification using DWT," *J. Biomed. Signal Process. Control*, vol. 18, pp. 138–144, 2015.