

FINAL PROJECT: AUDIO SENTIMEN ANALYSIS
PENGANTAR PEMROSESAN DATA MULTIMEDIA
KELOMPOK 2



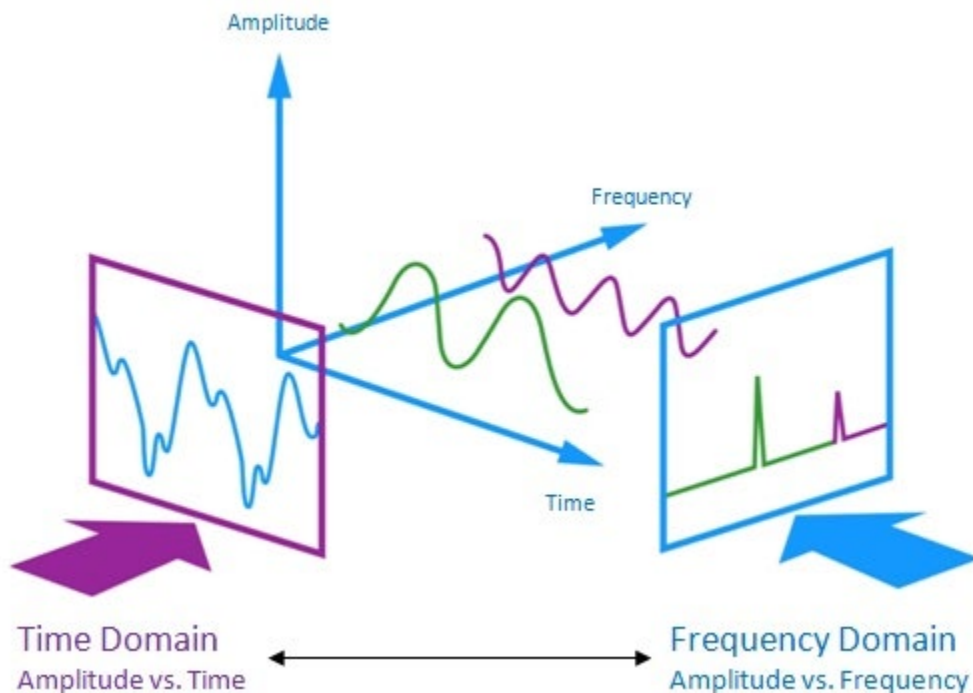
DISUSUN OLEH:
PANDE GEDE DANI WISMAGATHA (2108561022)
I GEDE NGURAH WAHYU ANANTA (2108561102)
I WAYAN DIMAS WIRAHADI SAPUTRA (2108561112)

PROGRAM STUDI INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS UDAYANA
JULI
2023

PENDAHULUAN

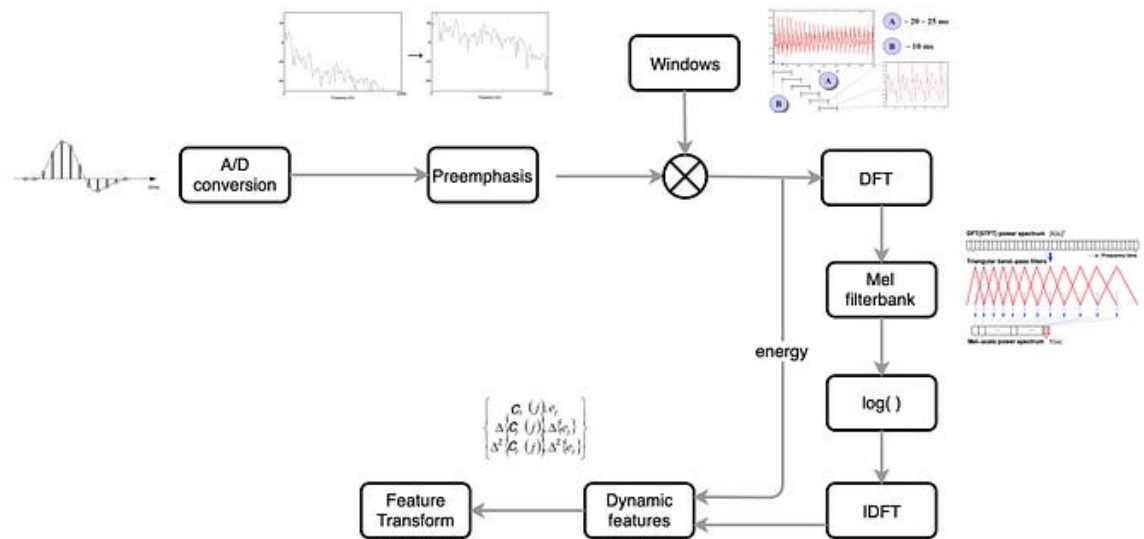
Mel Frequency Cepstral Coefficients (MFCC) adalah metode yang digunakan untuk melakukan *feature extraction* pada sinyal suara. *Feature extraction* adalah proses mengkonversikan sinyal suara menjadi beberapa parameter yang dapat digunakan untuk mengidentifikasi karakteristik suara. MFCC banyak digunakan dalam bidang speech technology, baik untuk speaker recognition maupun speech recognition. Metode ini mampu menangkap karakteristik suara yang sangat penting bagi pengenalan suara, atau dengan kata lain dapat menangkap informasi-informasi penting yang terkandung dalam signal suara . Untuk menghitung MFCC, adapun tahapan yang harus dilalui, yaitu *Fast Fourier Transform* (FFT).[1]

Fast Fourier Transform (FFT) adalah salah satu tahapan dalam menghitung MFCC. FFT digunakan untuk mengubah sinyal suara dari domain waktu ke domain frekuensi. Setelah itu, *Mel Frequency Wrapping* (MFW) digunakan untuk memetakan frekuensi ke skala Mel yang mendekati persepsi pendengaran manusia.[2]



Gambar 1. Perbandingan Time Domain dan Frequency Domain

Librosa adalah sebuah library Python yang digunakan untuk analisis musik dan audio. Library ini menyediakan berbagai fungsi untuk melakukan ekstraksi fitur, visualisasi, dan manipulasi data audio. Salah satu fitur yang dapat dihitung dengan Librosa adalah *Mel Frequency Cepstral Coefficients* (MFCC). Fungsi `librosa.feature.mfcc` dapat digunakan untuk menghitung koefisien MFCC dari sinyal audio. Selain itu, Librosa juga menyediakan fungsi `librosa.feature.delta` untuk menghitung delta dan delta-delta features dari koefisien MFCC yang dihasilkan oleh fungsi `librosa.feature.mfcc`. Delta dan delta-delta features memberikan informasi tambahan tentang perubahan sinyal suara dari waktu ke waktu.[3]



Gambar 2. MFCC Flowchart

Selain MFCC, adapun beberapa metode yang dapat digunakan dalam mengekstraksi fitur audio, seperti *Zero Crossing Rate* (ZCR) dan *Root Mean Squared Error* (RMSE). *Zero Crossing Rate* (ZCR) adalah jumlah kali sinyal audio melewati nilai nol dalam satu frame. ZCR dapat digunakan sebagai fitur untuk mengidentifikasi jenis suara, seperti suara vokal atau instrumen musik. *Root Mean Squared Error* (RMSE) adalah ukuran akurasi yang digunakan untuk membandingkan kesalahan prediksi dari berbagai model untuk satu dataset tertentu. RMSE dihitung dengan mengambil akar kuadrat dari rata-rata kuadrat kesalahan prediksi. Nilai RMSE selalu non-negatif, dan nilai 0 menunjukkan prediksi yang sempurna.[4]

$$ZCR = \frac{1}{2N} \sum_{n=1}^N |sign(x[n]) - sign(x[n-1])|$$

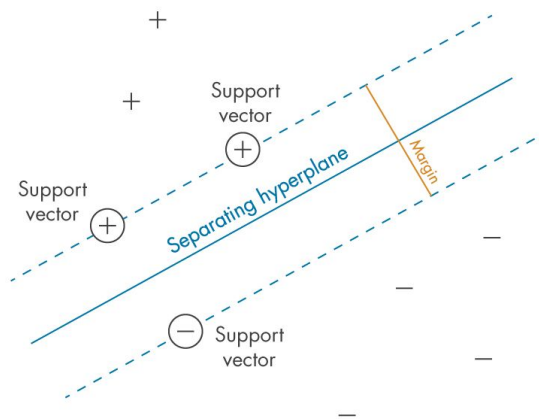
Gambar 3. Rumus Zero Crossing Rate (ZCR)

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(\hat{y}_i - y_i)^2}{n}}$$

Gambar 4. Rumus Root Mean Squared Error (RMSE)

Support Vector Machine (SVM) adalah salah satu algoritma machine learning dengan pendekatan supervised learning yang bekerja dengan mencari hyperplane atau fungsi pemisah terbaik untuk memisahkan kelas. Algoritma SVM memiliki konsep dan dasar matematis yang mapan sehingga menjadi algoritma yang populer. Algoritma ini dapat digunakan untuk klasifikasi (SVM classification) dan regresi (SVM regression).[5]

SVM bekerja diawali dengan menentukan batas kelas (*hyperplane*) yang memungkinkan dari dataset yang diberikan. Kemudian, algoritma ini memilih *hyperplane* berdasarkan margin maksimum yang dapat dicapai.[5]



Gambar 5. Ilustrasi Algoritma SVM

Algoritma SVM memiliki beberapa atribut, seperti:

1. *Hyperplane*

Hyperplane adalah batas keputusan yang membedakan dua kelas dalam SVM. Titik data yang jatuh di kedua sisi hyperplane dapat dikaitkan dengan kelas yang berbeda.[5]

2. *Support Vector*

Support vector ialah titik data atau vektor yang paling dekat dengan *hyperplane* dan yang mempengaruhi posisi *hyperplane*. Karena vektor-vektor ini mendukung *hyperplane*, maka disebut *support vector*. [5]

3. *Max Margin*

Margin adalah jarak antara support vector dari masing-masing kelas di sekitar hyperplane. Pada gambar dibawah, margin diilustrasikan dengan jarak antara 2 garis putus. Margin terbesar (max margin) dapat ditemukan dengan memaksimalkan nilai jarak antara hyperplane dan titik terdekatnya.[5]

4. *Kernel*

Kernel SVM biasanya digunakan ketika dataset berupa data yang tidak linier, atau data memuat kelas-kelas yang overlap atau tercampur.[6]

Adapun beberapa *kernel* yang ada pada SVM, yaitu:

- a. Gaussian Radial Basis Function (RBF) Kernel. Salah satu kernel yang paling kuat dan umum digunakan di SVM. Biasanya pilihan untuk data non-linear.
- b. Sigmoid Kernel. Lebih berguna dalam jaringan saraf daripada di mesin vektor dukungan, tetapi ada kasus penggunaan khusus sesekali.
- c. Polynomial Kernel.
- d. Lainnya, seperti Laplace RBF, Hyperbolic tangent, Sigmoid, Bessel function of the first kind, dan ANOVA RBF.

Adapun beberapa keunggulan dari algoritma SVM, yaitu:

1. Efektif pada dataset dengan margin pemisahan kelas yang jelas.[6]
2. Efektif dalam kasus di mana jumlah fitur lebih besar dari jumlah titik data.[6]

3. Fungsi kernel yang berbeda dapat ditentukan untuk fungsi keputusan.[6]
4. SVM dapat menemukan hubungan kompleks antara data tanpa kita perlu melakukan banyak transformasi sendiri.[6]

Dibalik keunggulan yang dimiliki oleh algoritma SVM ini, ada pula beberapa kelemahan yang dimilikinya, seperti:

1. SVM tidak berfungsi dengan baik pada dataset yang besar karena waktu pelatihan yang dibutuhkan lebih tinggi.[6]
2. SVM tidak secara langsung memberikan perkiraan probabilitas. Itu dihitung menggunakan validasi silang (cross validation) yang membutuhkan computation cost yang lebih tinggi.[6]
3. Kita perlu memahami fungsi kernel.[6]

CODE

Kode dipisah menjadi beberapa modul, yaitu:

1. Home.py
2. dataholder.py
3. process.py
4. UI.py
5. machine.py
6. teams.py
7. library.py
8. style.css
9. svm_model_kelompok_dimas_pande_wahyu.pkl
10. svm_scaler_kelompok_dimas_pande_wahyu.pkl

Dengan fungsi dan detail sebagai berikut;

1. Home.py
Program utama yang menjalankan streamlit sebagai UI.
2. dataholder.py
Program yang menampung dan memanipulasi pengambilan data Image untuk menampilkan foto anggota tim.
3. process.py
Program untuk memanggil model machine learning dan variabel skala kemudian menjalankan fungsi MFCC, ZCR dan RMSE ke data input agar data yang dimasukkan dapat menyesuaikan dengan data yang telah dipelajari oleh mesin.
4. UI.py
Program untuk mengubah tampilan pada web streamlit (menggunakan module streamlit).
5. machine.py
Sebuah program yang memberikan *page* (halaman) baru pada streamlit dengan isi rangkaian tahapan dalam pembuatan model machine learning.

6. teams.py
Sebuah program yang memberikan *page* (halaman) baru pada streamlit dengan foto para anggota tim.
7. library.py
Program yang menyimpan segala library yang dibutuhkan dalam menjalankan semua program.
8. style.css
Program yang digunakan untuk styling css media player audio.
9. svm_model_kelompok_dimas_pande_wahyu.pkl
Sebuah file pickle yang digunakan untuk menyimpan model SVM.
10. svm_scaler_kelompok_dimas_pande_wahyu.pkl
Sebuah file pickle yang digunakan untuk menyimpan variable skala pada model SVM.

Penjelasan lengkap ada pada kode. Kode bisa diakses menggunakan;

Github : [LvnnnX/Pemrosesan-Data-Multimedia: Final-Project has uploaded!](https://github.com/LvnnnX/Pemrosesan-Data-Multimedia: Final-Project has uploaded!),
[click this link below to access \(github.com\)](https://github.com/LvnnnX/Pemrosesan-Data-Multimedia) /
<https://github.com/LvnnnX/Pemrosesan-Data-Multimedia>

Streamlit : <https://bit.ly/Final-Project-PPDM-Kelompok-2>

Mohon untuk memerhatikan tutorial sebelum menjalankan kode tersebut.

PENUTUP

Mel Frequency Cepstral Coefficients (MFCC) adalah metode yang digunakan untuk melakukan feature extraction pada sinyal suara. Metode ini mampu menangkap karakteristik suara yang penting bagi pengenalan suara dan banyak digunakan dalam bidang speech technology. Librosa adalah library Python yang digunakan untuk analisis musik dan audio. Librosa menyediakan fungsi untuk menghitung koefisien MFCC dan fitur lainnya seperti delta dan delta-delta features. Selain MFCC, ada juga metode lain yang digunakan untuk ekstraksi fitur audio, seperti *Zero Crossing Rate* (ZCR) dan *Root Mean Squared Error* (RMSE). ZCR dapat digunakan untuk mengidentifikasi jenis suara, sedangkan RMSE adalah ukuran akurasi yang digunakan untuk membandingkan kesalahan prediksi dari berbagai model. *Support Vector Machine* (SVM) adalah algoritma machine learning yang digunakan untuk klasifikasi dan regresi. Algoritma SVM mencari hyperplane terbaik untuk memisahkan kelas dengan margin maksimum. Atribut-algoritma SVM meliputi *hyperplane*, *support vector*, *max margin*, dan *kernel*. *Kernel* SVM digunakan ketika dataset tidak linier atau terdapat overlap antar kelas. SVM memiliki keunggulan seperti efektif pada dataset dengan margin pemisahan yang jelas, efektif pada dataset dengan jumlah fitur yang lebih besar dari jumlah titik data, kemampuan menggunakan berbagai fungsi kernel, dan kemampuan menemukan hubungan kompleks antara data. Namun, SVM juga memiliki beberapa kelemahan, seperti tidak berfungsi dengan baik pada dataset yang besar, tidak memberikan perkiraan probabilitas secara langsung, dan membutuhkan pemahaman yang baik tentang fungsi kernel.

REFERENSI

- [1] “Speech Recognition — Feature Extraction MFCC & PLP | by Jonathan Hui | Medium.” <https://jonathan-hui.medium.com/speech-recognition-feature-extraction-mfcc-plp-5455f5a69dd9> (diakses 5 Juli 2023).
- [2] “Extract MFCC, log energy, delta, and delta-delta of audio signal - MATLAB mfcc.” <https://www.mathworks.com/help/audio/ref/mfcc.html> (diakses 4 Juli 2023).
- [3] McFee, Brian, “librosa/librosa: 0.10.0.post2,” Mar 2023, doi: 10.5281/zenodo.7746972.
- [4] “Root-mean-square deviation - Wikipedia.” https://en.wikipedia.org/wiki/Root-mean-square_deviation (diakses 5 Juli 2023).
- [5] “Penjelasan Lengkap Algoritma Support Vector Machine (SVM) - Trivusi.” <https://www.trivusi.web.id/2022/04/algoritma-svm.html> (diakses 5 Juli 2023).
- [6] “Support Vector Machine, Algoritma untuk Machine Learning – GEOSPASIALIS.” <https://geospasialis.com/support-vector-machine/> (diakses 5 Juli 2023).