**FINAL PROJECT: *AUDIO* SENTIMEN ANALYSIS**

**PENGANTAR PEMROSESAN DATA MULTIMEDIA**

**KELOMPOK 2**

Logo

Description automatically generated

**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**

# BAB I : PENDAHULUAN

## 1.1 Latar Belakang

Sentimen analisis pada *audio* adalah bidang yang bertujuan untuk mengidentifikasi, menganalisis, dan memahami ekspresi emosional yang terkandung dalam suara manusia. Dengan menggunakan teknik machine learning, sentimen analisis pada *audio* memungkinkan kita untuk mengukur dan mempelajari perasaan seseorang melalui suara mereka.

Tantangan utama dalam sentimen analisis pada *audio* adalah spektrum suara dari manusia. Suara manusia dapat mengandung berbagai aspek, seperti nada, intonasi, kecepatan bicara, dan variasi vokal. Memahami dan menginterpretasikan elemen-elemen ini dengan tepat merupakan tugas yang rumit. Namun, kemajuan dalam teknologi pengolahan sinyal dan machine learning telah memungkinkan pengembangan metode dan algoritma yang lebih canggih untuk menganalisis sentimen pada *audio*.

*Support Vector Machine* (SVM) merupakan sebuah algoritma yang sering dijumpai pada *project-project* machine learning merupakan sebuah algoritma yang dapat memprediksikan kategori maupun regresi. Algoritma ini mengambil keputusan berdasarkan *support vector* yang ditetapkan dengan rumus matematika.[1]

Dalam penelitina ini, algoritma SVM yang diimplementasikan dapat menjadi sebuah pendekatan yang efektif untuk mengklasifikasikan sentimen pada data *audio*. Dengan memberikan data yang cukup pada algoritma SVM, algoritma ini dapat mempelajari pola-pola dari sentimen yang diberikan, baik *sad* maupun *happy*. Dalam proses ini, parameter yang digunakan dapat disesuaikan untuk mendapatkan hasil dari *model* yang optimal dalam pengenalan sentimen.[1], [2]

## 1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, dapat diambil rumusan masalah sebagai berikut;

1. Bagaimana algoritma SVM dapat melakukan pemrosesan olahan data *audio.*
2. Bagaimana cara menentukan nilai parameter terbaik untuk membuat *model* SVM.
3. Bagaimana nilai akurasi, presisi, *recall* dan F1-*Scorenya*.

## 1.3 Tujuan

Berdasarkan rumusan masalah di atas, maka tujuan dari pembuatan program ini adalah:

1. Mengetahui cara algoritma SVM dapat melakukan pemrosesan olahan data *audio*.
2. Mengetahui cara menentukan nilai parameter yang terbaik untuk membuat *model* SVM.
3. Mengetahui nilai akurasi, presisi, *recall*, dan F1-*Scorenya*.

# BAB II : ISI

## 2.1 Manual Aplikasi

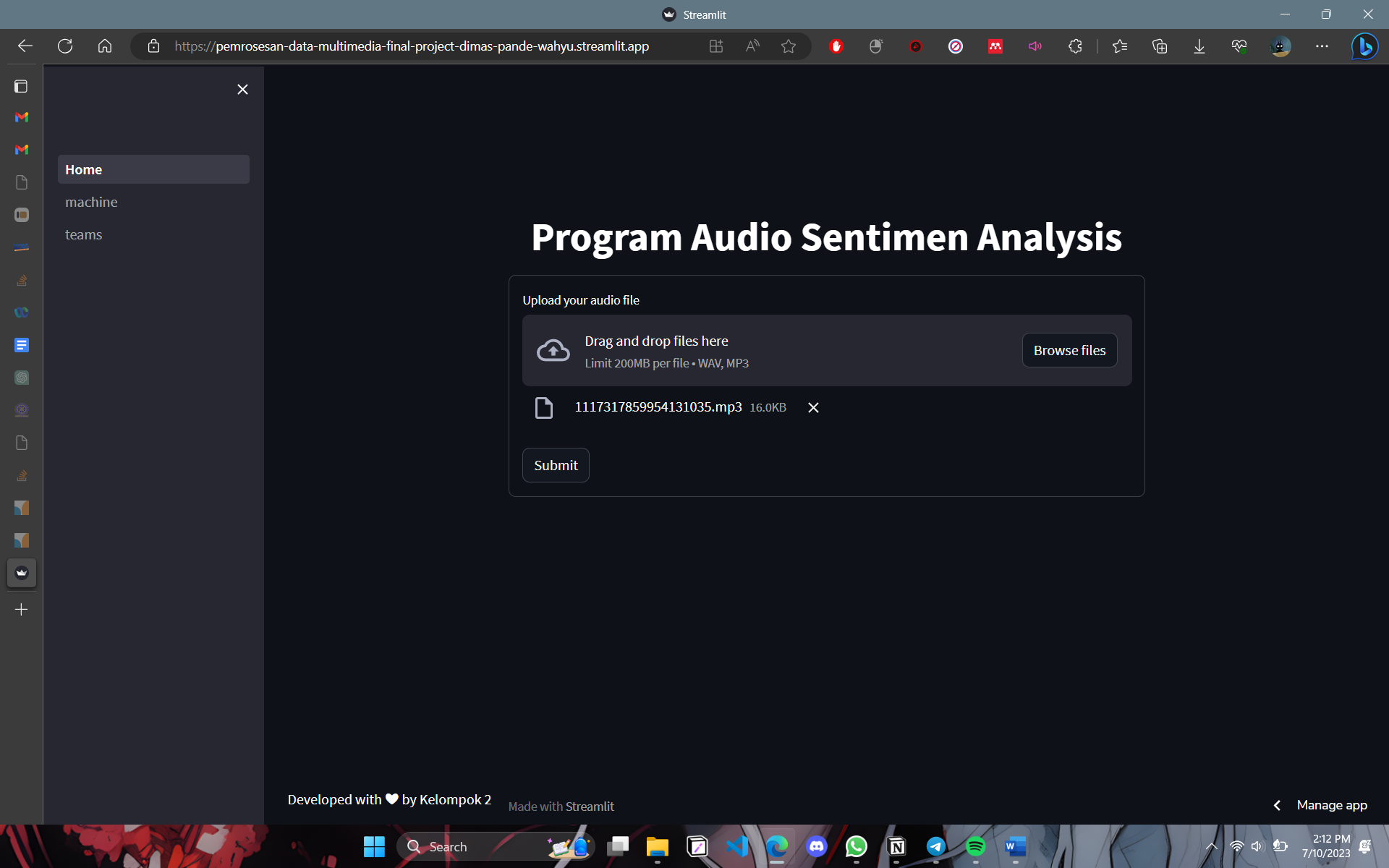
Program yang dibuat menggunakan *streamlit* dapat dijalankan dengan membuka lokasi folder dari program yang telah disimpan. Kemudian, membuka *command prompt* (CMD) atau terminal, lalu ketik “*streamlit run Home..py*” dan tekan enter. Home.py adalah nama dari *main file* / *file* utama yang akan dijalankan. Selain menggunakan *command prompt*, link <https://pemrosesan-data-multimedia-final-project-dimas-pande-wahyu.streamlit.app/> juga dapat digunakan untuk mengakses program. Contoh tampilan dari program yang telah dijalankan adalah sebagai berikut.

A screenshot of a computer

Description automatically generated

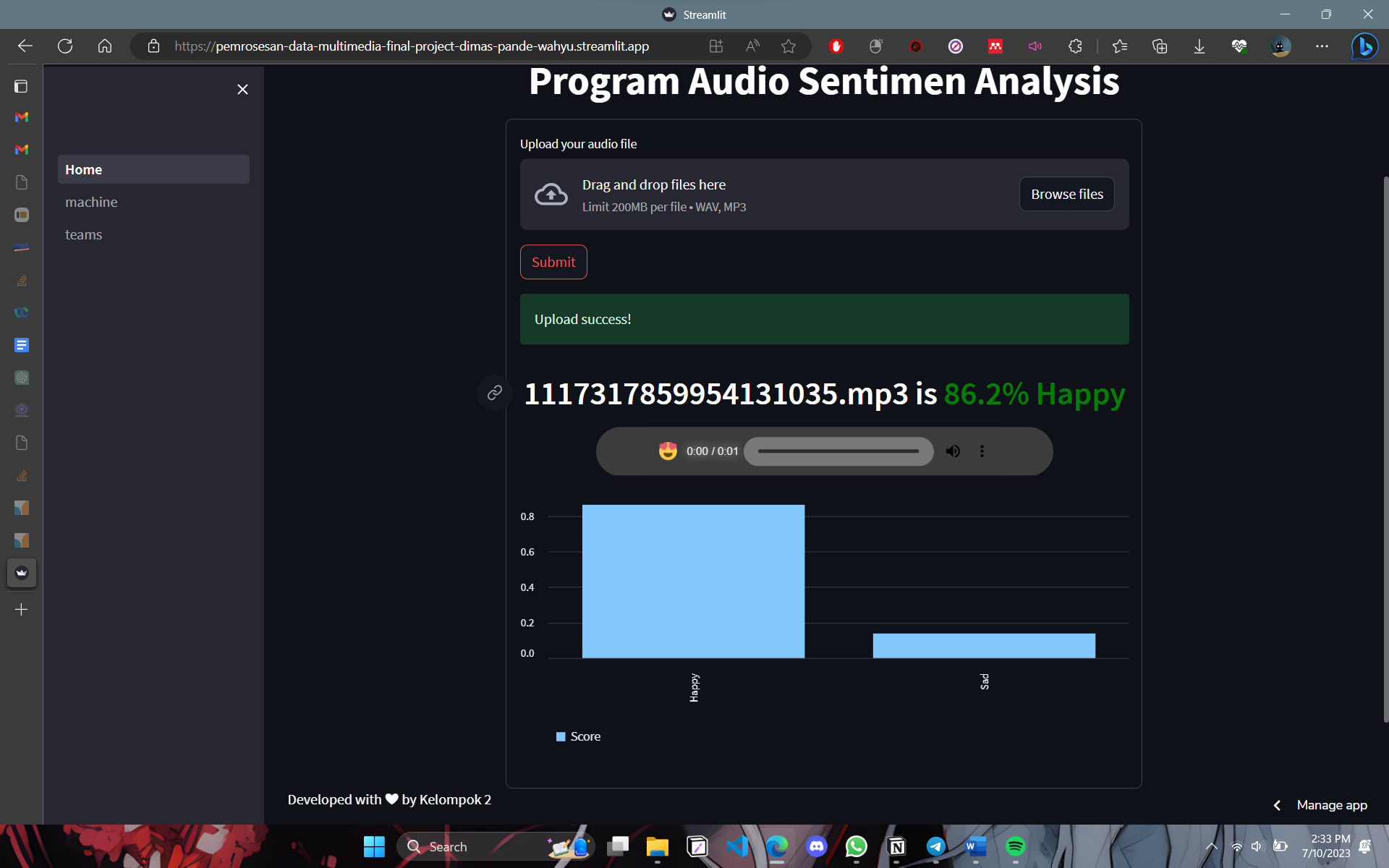
**Gambar 1.** Home Page pada Program

Setelah mendapatkan tampilan seperti pada gambar di atas. Pengguna dapat memasukkan *file audio* / suara yang ingin diinentifikasi sentimennya dengan menekan tombol “ *Browse files*” kemudian memilih *file-file* yang ingin diidentifikasi dengan maksimal ukuran *file* adalah 200MB dengan format WAV atau MP3. Setelah memasukkan *file* *audio*, *file* akan terlihat pada program seperti berikut.



**Gambar 2.** Home Page setelah memasukkan file audio

Setelah pengguna memasukkan *file* dan *file* tersebut nampak pada program, pengguna dapat menekan tombol ”*Submit*”. Setelah pengguna menekan tombol submit. Program akan mengambil *file* yang telah dimasukkan oleh pengguna kemudian melakukan *pre-processing* seperti mengekstrak fitur MFCC (*Mel-Frequency Cepstral Centroid*), ZCR (*Zero Crossing Rate*), dan RMSE (*Root Mean Squared Energy*). Kemudian, program akan mengubah nilai dari fitur yang telak diekstrak dengan menggunakan *MinMaxScaler* agar fitur memiliki skala yang sama. Terakhir, program akan memprediksi sentimen dari fitur yang telah diproses. Hasil prediksi dari fitur yang telah diproses adalah sebagai berikut.



**Gambar 3.** Home Page setelah program memprediksi

Program akan menampilkan nama dari *file* yang dimasukan oleh pengguna disertai dengan persentase dari sentimen yang diprediksi oleh program. Dibawah nama *file*, terdapat sebuah media player agar pengguna dapat mendengarkan audio yang baru saja dimasukan ke program. Dan pada bagian paling bawah, terdapat grafik yang menyatakan persentase dari sentimen yang telah di prediksi oleh program. Namun, jika pengguna tidak memasukan file apapun dan tetap menekan tombol “*Submit*”. Maka program akan mengeluarkan sebuah teks “*Please upload your audio file*” seperti gambar berikut.

A screenshot of a computer

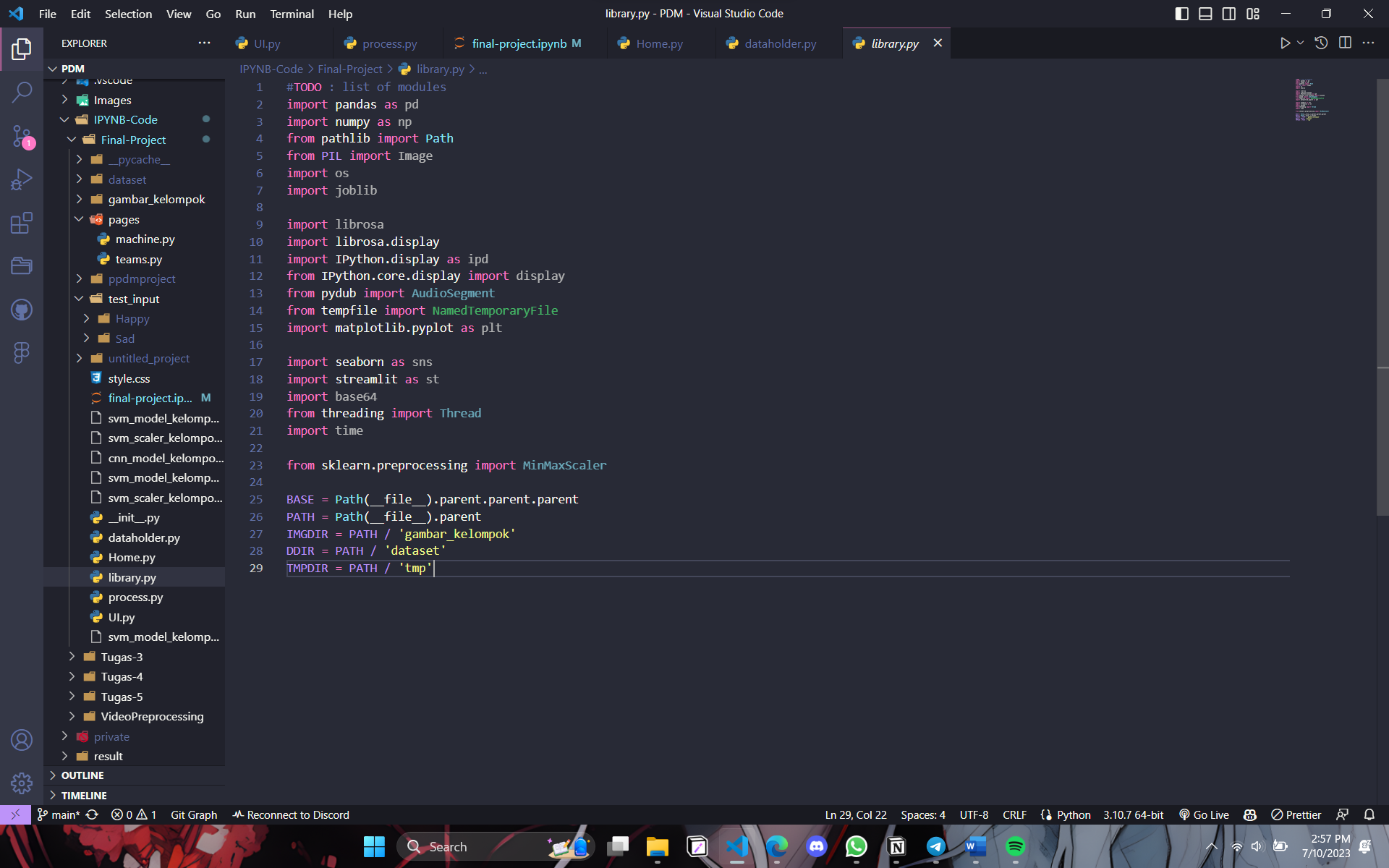
Description automatically generated

**Gambar 4.** Home Page jika submit tanpa file

## 2.2 *Source Code Program*

### 2.2.1 library.py

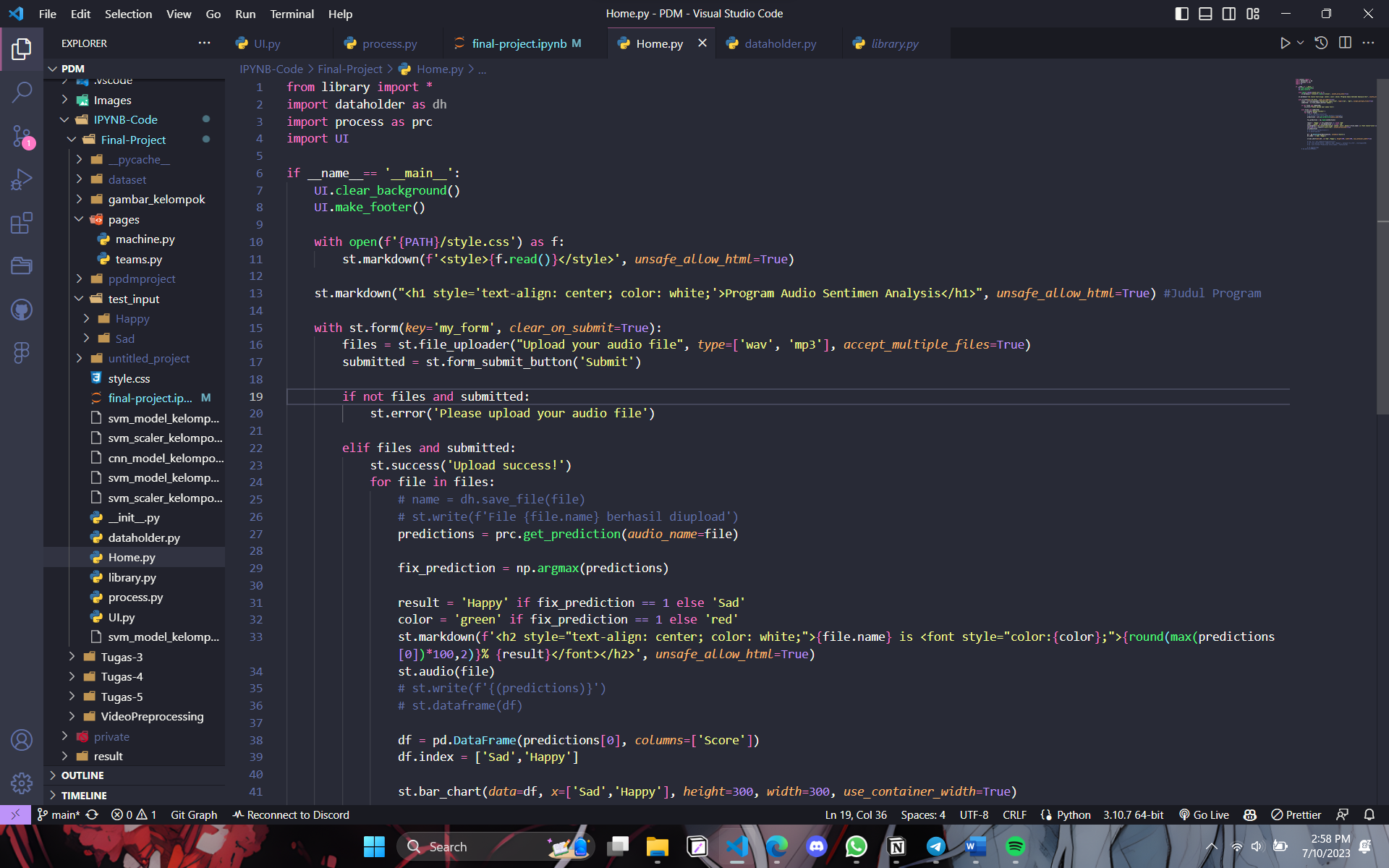
*File* library.py berisikan *library-library* / *modules* yang akan digunakan / *diimport* pada program untuk menjalankan program.



**Gambar 5.** library.py

### 2.2.2 Home.py

*File* Home.py menampilkan halaman utama (*Home Page*) pada web. File ini mengambil masukan dari user berupa *file audio* yang nantinya akan diprediksi serta menampilkan hasil prediksinya.



**Gambar 6.** Home.py

### 2.2.3 process.py

*File* process.py mengambil data *audio* yang sebelumnya ada pada Home.py. Kemudian, mengekstrak fitur MFCC, ZCR, dan RMSE dari data *audio* tersebut. Setelah fitur terekstrak, program ini kemudian mengubah skala dari fitur yang ada menjadi rentang 0 dan 1. Kemudian memanggil *model* SVM yang telah disimpan dan memprediksi fitur tersebut. Kemudian hasil prediksi di*return* dengan menggunakan *function*.

A screenshot of a computer program

Description automatically generated

**Gambar 7.** process.py

### 2.2.4 dataholder.py

*File* dataholder.py berisikan sebuah *function* yang digunakan untuk mengambil gambar pada directory gambar untuk ditampilkan pada web.

A screenshot of a computer program

Description automatically generated

**Gambar 8.** dataholder.py

### 2.2.5 UI.py

*File* UI.py berisikan fungsi yang digunakan untuk mengubah UI pada web.

A screenshot of a computer program

Description automatically generated

**Gambar 9.** UI.py

### 2.2.6 machine.py

*File* machine.py digunakan untuk menampilkan halaman “*machine*“ pada web. Halaman ini berisikan penjelasan singkat tentang bagaimana program bekerja, bagaimana program memproses data dan bagaimana model menampilkan prediksi.

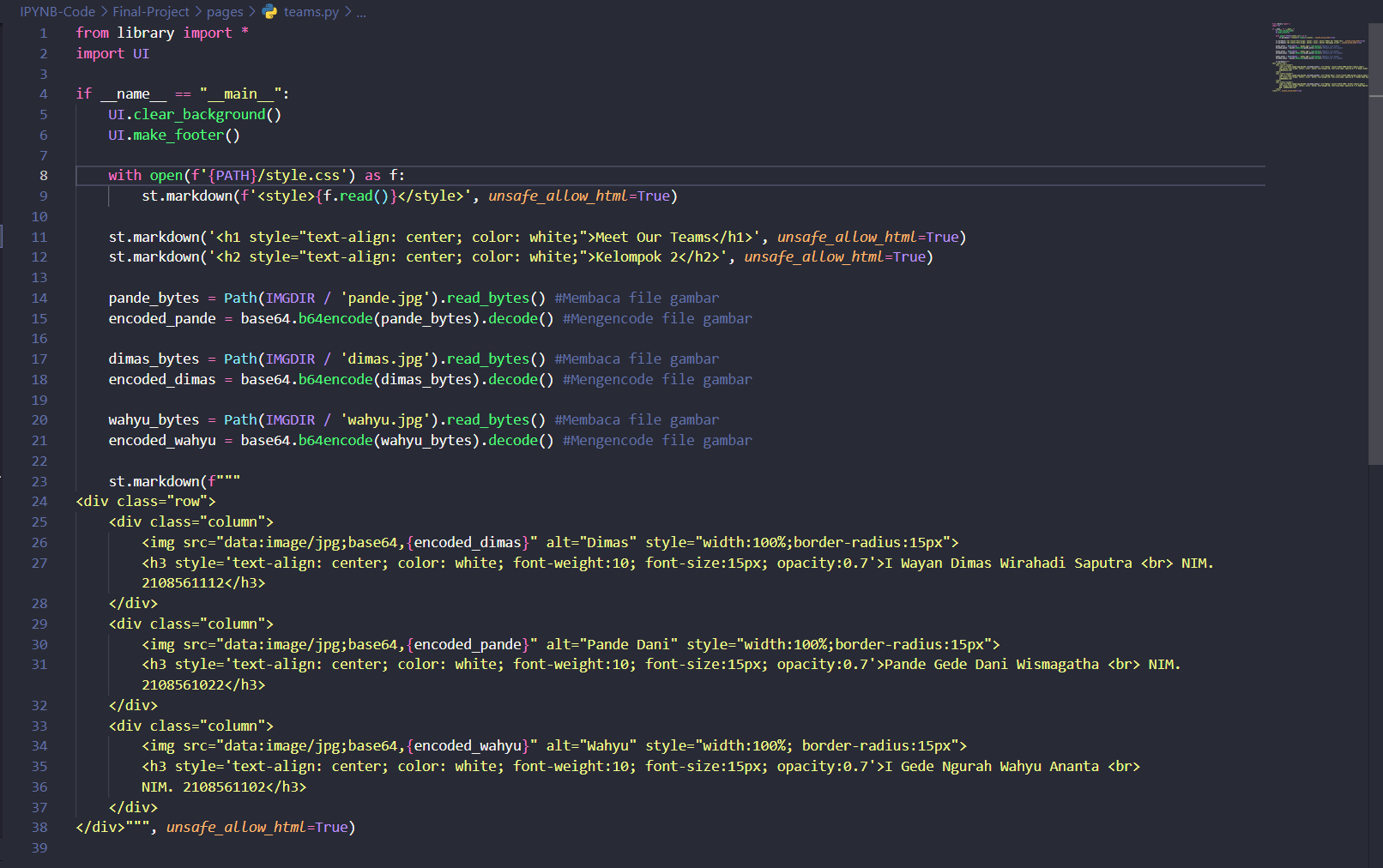
A screen shot of a computer screen

Description automatically generated

**Gambar 10.** machine.py

### 2.2.7 teams.py

*File* teams.py digunakan untuk menampilkan halaman “*teams*” pada web. Halaman ini berisikan foto dari tim pembuat program dengan memanggil function yang ada pada dataholder untuk menampilkan foto-foto tersebut.



**Gambar 11.** teams.py

**2.3 Evaluasi Sistem**

### 2.3.1 *Tuning Hyper-Parameter*

Pada pengujian ini, *model* terbaik dicari dengan menggunakan *Tuning Hyper-Parameter* dengan parameter-parameter yang terdapat pada algoritma *Support Vector Machine* (SVM) seperti C (parameter regularisasi), Gamma (tingkat kekurvaan), Kernel (fungsi pada masukan data), serta Decision Function Shape (fungsi pengambilan keputusan). Pada proses pembuatan model, penulis menggunakan metode *Nested Loop* untuk melakukan *Tuning Hyper-Parameter* seperti pada gambar berikut:

A screenshot of a computer screen

Description automatically generated

**Gambar 12.** Tuning Hyper-Parameter

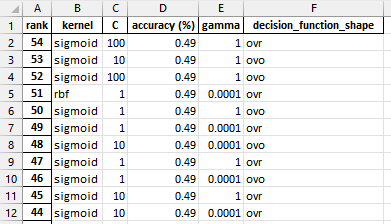
Seperti yang terlihat pada gambar, penulis menggunakan beberapa kombinasi pada parameter yang tersedia untuk mendapatkan hasil terbaik. Hasil akurasi terbaik dari kombinasi parameter yaitu:

A table with numbers and letters

Description automatically generated

**Gambar 13**. Hasil terbaik Tuning Hyper-Parameter

Dapat dilihat pada gambar di atas. Bahwa akurasi tertinggi adalah 86% dengan kombinasi parameter kernel rbf, nilai 1 pada C, nilai 1 pada gamma, dan nilai “*ovo*” pada *decision function shape*. Selain hasil terbaik, ada pula kombinasi yang memiliki akurasi paling rendah diantara kombinasi lainnya, yaitu:

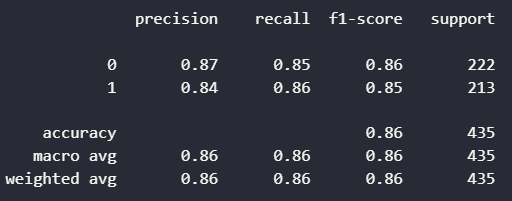


**Gambar 14.** Hasil terendah Tuning Hyper-Parameter

Berdasarkan urutan rank pada gambar di atas. Maka terlihat bahwa kernel sigmoid dengan nilai C 100, gamma 1 dan *decision function shape* “*ovr*” mendapatkan hasil terburuk pada akurasi, yaitu 49%.

### 2.3.2 *Classification Report Best Model*

Setelah mendapatkan model terbaik. Penulis melakukan pengambilan *Classification Report* pada model tersebut. Adapun hasil dari *Classification Report*nya, yaitu:



**Gambar 15.** Best Model Classification Report

Model terbaik mendapatkan *precision* dengan nilai 86%, *recall* dengan nilai 86%, f1-*score* dengan nilai 86% dan support 435. Ini menunjukkan bahwa model SVM ini mampu mengklasifikasikan sentimen dengan baik dan sedikit kesalahan.

.

# PENUTUP

Penelitian ini bertujuan untuk mengidentifikasi, menganalisis, dan memahami ekspresi emosional yang terkandung dalam suara manusia dengan menggunakan teknik machine learning. Salah satu tantangan utama dalam sentimen analisis pada audio adalah spektrum suara dari manusia. Namun, kemajuan dalam teknologi pengolahan sinyal dan machine learning telah memungkinkan pengembangan metode dan algoritma yang lebih canggih untuk menganalisis sentimen pada audio. Dalam penelitian ini, algoritma SVM diimplementasikan sebagai pendekatan yang efektif untuk mengklasifikasikan sentimen pada data audio. Penelitian ini mengeksplorasi cara menentukan nilai parameter terbaik untuk membuat model SVM serta mengukur nilai akurasi, presisi, recall dan F1-Scorenya*.* Didapatkan hasil bahwa algoritma SVM mampu untuk mengklasifikasikan sentimen suara manusia dengan sangat baik. Algoritma ini mendapat skor akurasi hingga 86% pada model terbaik dengan skor *precision, recall,* dan f1-*score* 86% dan *support* pada 435.

**PROGRAM DAPAT DIAKSES MENGGUNAKAN**

**Github :** [LvnnnX/PPDM-Final-Project (github.com)](https://github.com/LvnnnX/PPDM-Final-Project)

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

**Youtube :** <https://www.youtube.com/watch?v=srhhKhnfW2I>

# **REFERE**NSI

[1] M. Fahmi dan I. Suhartana, “Perbandingan Algoritma Decision Tree Dan Support Vector Machine Dalam Prediksi Kualitas Udara,” 2022. [Daring]. Tersedia pada: https://data.jakarta.go.id/.

[2] M. F. Naufal, “ANALISIS PERBANDINGAN ALGORITMA SVM, KNN, DAN CNN UNTUK KLASIFIKASI CITRA CUACA”, doi: 10.25126/jtiik.202184553.