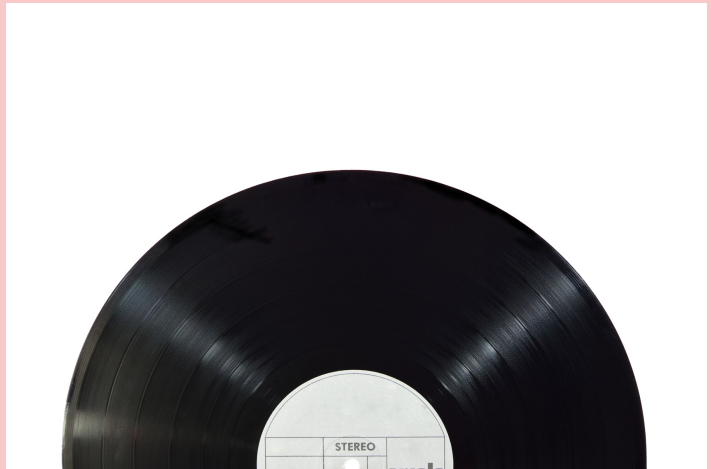
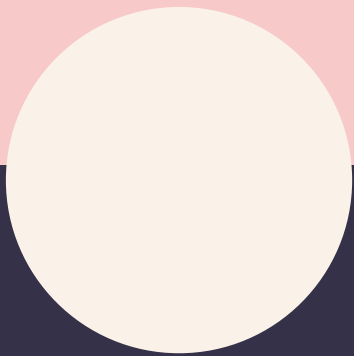


UNIVERSITAS UDAYANA  
INFORMATIKA



# MUSIC-G MANUAL BOOK

APLIKASI WEBSITE PREDIKSI GENRE MUSIK  
BERBASIS MACHINE LEARNING

Ayu Kadek Nadya Oktaviana  
Drs. I Wayan Santiyasa, M.Si.  
Dr. Ngurah Agus Sanjaya ER, S.Kom., M.Kom

# DAFTAR ISI

## 01

Pendahuluan

## 02

Penggunaan

## 03

Fitur Aplikasi

## 04

Source Code

# DESKRIPSI

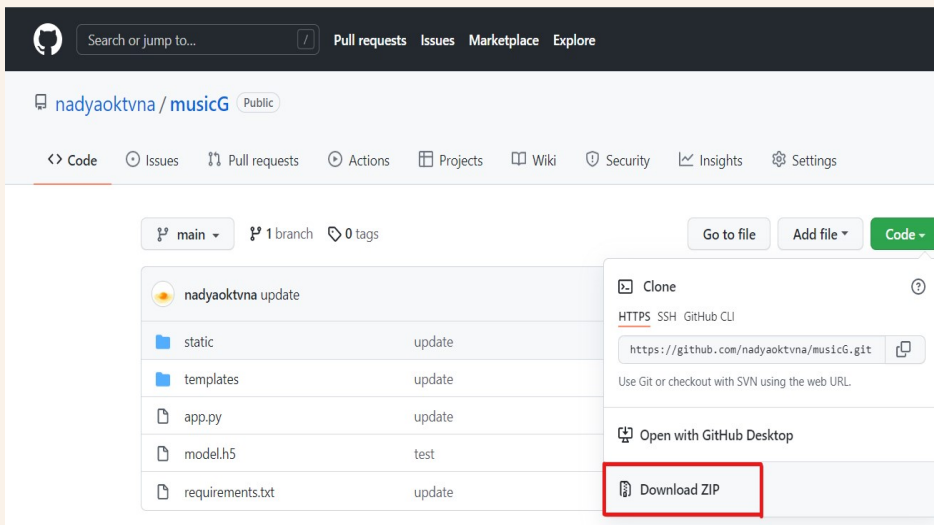
Music-G merupakan aplikasi berbasis website yang memanfaatkan konsep *machine learning* untuk memprediksi genre dari sebuah musik.

Genre yang dapat diprediksi yaitu Blues, Classical, Country, Disco, Hip-hop, Jazz, Metal, Pop, Reggae, dan Rock.

Musik yang diprediksi harus dalam bentuk format file audio .wav (WAVE-form). Pada website ini sudah disediakan link website luar untuk mengubah lagu ke bentuk .wav.

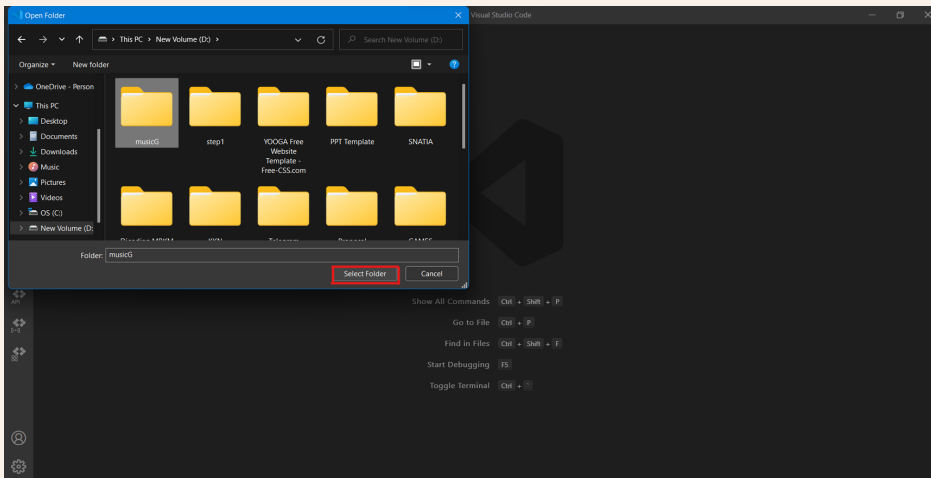


1. Unduh project melalui link git-hub:  
<https://github.com/nadyaoktvna/musicG>



Tekan tombol 'code', kemudian pilih 'Download ZIP'.

2. Unzip folder dan buka folder pada *text editor* seperti Visual Studio Code, PyCharm, dan lainnya.

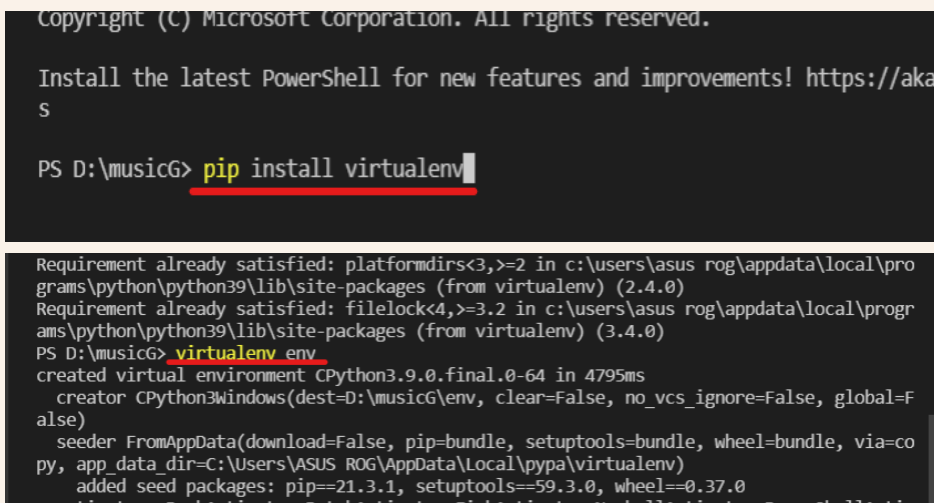


Untuk melakukan unzip folder, klik kanan pada file zip, lalu pilih opsi **Extract all**. Lalu tentukan destinasi file yang diekstrak.

*Text editor* yang digunakan pada manual book ini adalah VSCode (Visual Studio Code). Langkah membuka sebuah folder pada VSCode:

Buka aplikasi VSCode → **File** → **Open Folder...** → Pilih folder yang telah diunzip sebelumnya.

3. Setelah folder berhasil dibuka pada text editor. Buka terminal text editor dengan cara klik menu **Terminal** → **New Terminal**.



Sesudah membuka terminal, install virtual environment dengan cara mengetik 'pip install virtualenv'.

Selesai instalasi virtual environment, langkah selanjutnya adalah membuat virtual environment dengan cara mengetik 'virtualenv env'.

4. Tahap selanjutnya adalah instalasi requirements dalam terminal dengan perintah 'pip install -r requirements.txt'

```
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL

Requirement already satisfied: idna<4,>=2.5 in c:\users\asus rog\appdata\local\programs\python\python39\lib\site-packages (2021.10.8)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\asus rog\appdata\local\programs\python\python39\lib\site-packages (2021.10.8)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\asus rog\appdata\local\programs\python\python39\lib\site-packages (2.0.9)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\asus rog\appdata\local\programs\python\python39\lib\site-packages (1.26.7)
Requirement already satisfied: zipp>=0.5 in c:\users\asus rog\appdata\local\programs\python\python39\lib\site-packages (3.6.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in c:\users\asus rog\appdata\local\programs\python\python39\lib\site-packages (0.4.8)
Requirement already satisfied: oauthlib>=3.0.0 in c:\users\asus rog\appdata\local\programs\python\python39\lib\site-packages (3.1.1)
PS D:\musicG> pip install -r requirements.txt
```

Pada tahap ini memerlukan waktu yang sedikit lama apabila user belum menginstall beberapa hal di komputer masing-masing seperti install Tensorflow.

5. Selanjutnya setelah tahap instalasi requirements selesai, jalankan file app.py dengan perintah pada terminal 'python.exe app.py'

```
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL

Requirement already satisfied: idna<4,>=2.5 in c:\users\asus rog\appdata\local\programs\python\python39\lib\site-packages (2021.10.8)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\asus rog\appdata\local\programs\python\python39\lib\site-packages (2021.10.8)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\asus rog\appdata\local\programs\python\python39\lib\site-packages (2.0.9)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\asus rog\appdata\local\programs\python\python39\lib\site-packages (1.26.7)
Requirement already satisfied: zipp>=0.5 in c:\users\asus rog\appdata\local\programs\python\python39\lib\site-packages (3.6.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in c:\users\asus rog\appdata\local\programs\python\python39\lib\site-packages (0.4.8)
Requirement already satisfied: oauthlib>=3.0.0 in c:\users\asus rog\appdata\local\programs\python\python39\lib\site-packages (3.1.1)
PS D:\musicG> python.exe app.py
```

Tujuan tahap ini untuk mendapatkan link local deployment untuk menjalankan website.

6. Tunggu proses sebelumnya sampai muncul link. Klik link tersebut dan akan langsung diarahkan menuju website Music-G.

```
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL

und
2021-12-26 12:02:27.806245: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'C:\Windows\System32\user32.dll'; dlerror: user32.dll not found
2021-12-26 12:02:27.807524: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'C:\Windows\System32\user32.dll'; dlerror: user32.dll not found
2021-12-26 12:02:27.809000: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'C:\Windows\System32\user32.dll'; dlerror: user32.dll not found
2021-12-26 12:02:27.811707: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'C:\Windows\System32\user32.dll'; dlerror: user32.dll not found
2021-12-26 12:02:27.812224: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1850] Cannot dlopen some GPU libraries. Please make sure the binary has access to a CUDA device, or skip this device by setting TF_FORCE_GPU_ALLOW_GTOUCHES in performance-critical operations: AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
* Debugger is active!
* Debugger PIN: 121-467-866
* Running on all addresses.
WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://192.168.43.221:5000/ (Press CTRL+C to quit)
```

Salin link kemudian buka pada web browser masing-masing. Cara lainnya yaitu menekan tombol CTRL + Klik Link.

# HOME

Fitur Home merupakan tampilan awal website ketika diakses. Pada bagian atas, terdapat menu navigasi yaitu **Home**, **Predict**, dan **About**. Masing-masing menu dapat ditekan untuk mengarahkan ke halaman yang dimaksud.

Fitur ini juga menampilkan tujuan dari website ini yaitu "**Music Genre Prediction**" atau memprediksi genre dari sebuah musik/lagu dan terdapat tombol "**Predict**" untuk mengarahkan ke halaman Predict.





# PREDICT

Fitur Predict merupakan fitur utama dari website Music-G. Untuk melakukan prediksi genre, user dapat meng-unggah musik dengan menekan tombol "**Choose File**". Setelah memilih musik yang ingin di prediksi, user dapat mendengarkan musik yang dipilih. Kemudian untuk memprediksi genre, tekan tombol "**Predict**".

Jika audio yang diunggah bukan berformat audio .wav, maka user dapat mengubah audio tersebut dengan menekan tombol "**ini**" pada kalimat "Lagu harus dalam format `.wav`". Silahkan kunjungi website ini untuk mengubah musik anda menjadi `.wav`!" yang terdapat di halaman fitur predict.

— Predict —

## Genre Prediction

Lagu harus dalam format `.wav`. Silahkan kunjungi website ini untuk mengubah lagu anda menjadi `.wav`!

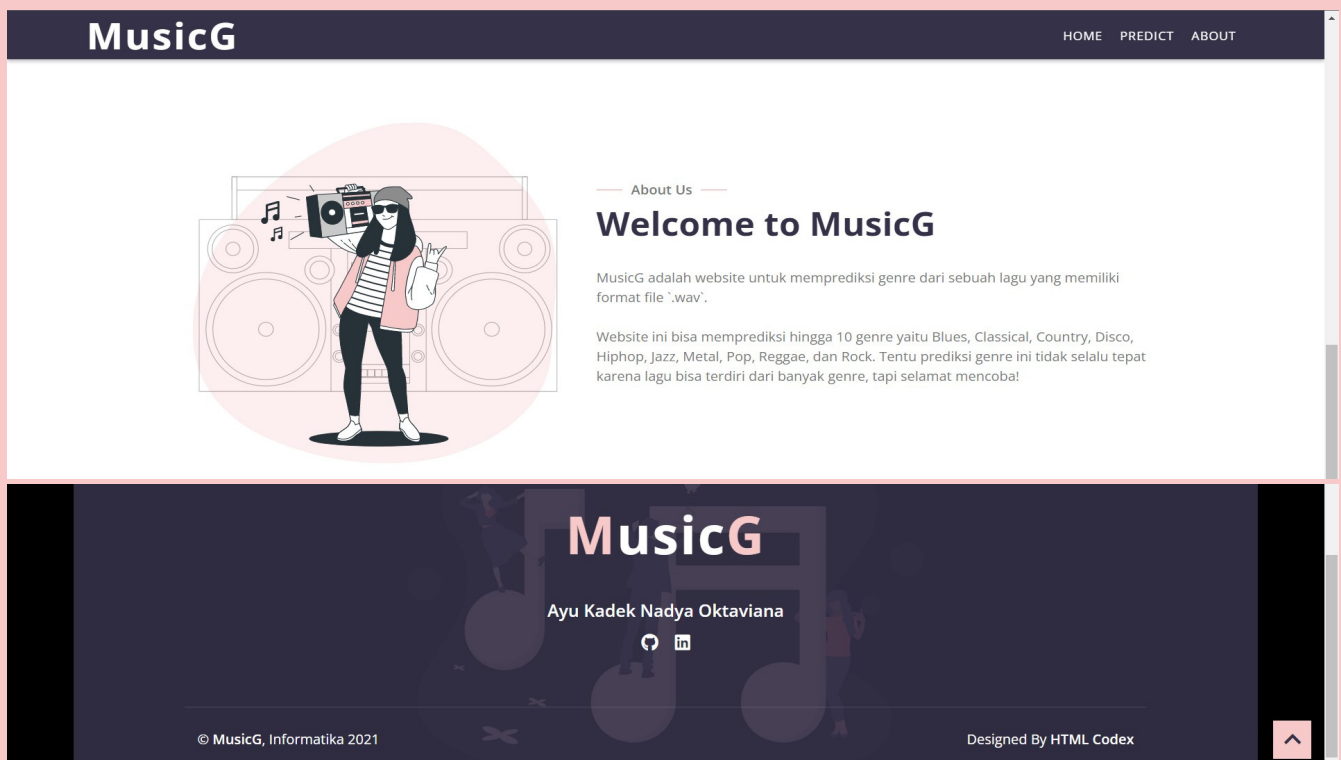
 No file chosen

▶ 0:00 / 0:00 ———— 🔊 ⋮

# ABOUT

Fitur About menampilkan informasi mengenai website dan dibawah halaman about terdapat *footer* yang memberi informasi mengenai *copyright*, nama pemilik, akun git-hub, dan tahun pembuatan aplikasi.

Fitur tambahan lainnya yaitu tombol otomatis menuju halaman atas. User hanya perlu menekan tombol tanda panah atas di pojok kanan bawah, kemudian halaman akan otomatis menuju halaman paling atas atau home tanpa perlu melakukan *scroll-up*.





# SOURCE CODE

## Deploy Model Machine Learning Using Flask

```
app = Flask(__name__)

model =
tf.keras.models.load_model('model.h5')
model.make_predict_function()

@app.route('/', methods=['GET'])
def main():
    return render_template('index.html')

@app.route('/', methods=['POST'])
def predict():
    if request.method == 'POST':
        file = request.files['file']
        if file:
            audio, sr = librosa.load(file)
            audio, _ = librosa.effects.trim(audio)
            audio = audio[:661500]

            mfccs_features = librosa.feature.mfcc(y =
audio, sr = 22050, n_mfcc = 40)
            mfccs_mean = np.mean(mfccs_features.T,
axis=0)
            mfccs_var = np.var(mfccs_features.T,
axis=0)
            mfccs_scaled_features =
mfccs_mean+mfccs_var
            mfccs_scaled_features =
mfccs_scaled_features.reshape(1,-1)

            predicted =
model.predict(mfccs_scaled_features)
            predicted_label = np.argmax(predicted,
axis=1)

            if predicted_label == [[0]]:
                genre_detected = 'Blues'
            elif predicted_label == [[1]]:
                genre_detected = 'Classical'
            elif predicted_label == [[2]]:
                genre_detected = 'Country'
            elif predicted_label == [[3]]:
                genre_detected = 'Disco'
            elif predicted_label == [[4]]:
                genre_detected = 'Hip-hop'
            elif predicted_label == [[5]]:
                genre_detected = 'Jazz'
            elif predicted_label == [[6]]:
                genre_detected = 'Metal'
            elif predicted_label == [[7]]:
                genre_detected = 'Pop'
            elif predicted_label == [[8]]:
                genre_detected = 'Reggae'
            else:
                genre_detected = 'Rock'

            return render_template('index.html',
prediction = genre_detected)

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=5000, debug=True)
```