

Identifikasi Lagu Barat Populer Menggunakan *Speech Recognition*

Kevin Chunady [2501967556], Marvelio Chandra [2540110533], Adrian Natanael [2501960801], Singgih Tulus Makmud [2501967581], Justin Arola Rusman [2540126701]

1. Overview

Identifikasi lagu dengan penggunaan teknologi *speech recognition* menjadi semakin populer di kalangan industri teknologi. Algoritma *speech recognition* dapat dilatih untuk mengenali pola unik suara saat seseorang bernyanyi atau berbicara. Teknologi ini dapat digunakan untuk mengidentifikasi nama lagu dan membantu mempelajari lagu baru. Proses identifikasi ini dilakukan dengan menganalisis pola dan struktur dari ucapan dan membandingkannya dengan dataset lagu-lagu yang tersimpan. Setelah ditemukan kecocokan, maka judul lagu dan penyanyi dapat ditampilkan.

Dengan adanya teknologi seperti ini, kami berencana mengembangkan sebuah produk aplikasi pengidentifikasi lagu. Untuk memperkecil cakupan dari aplikasi kami, aplikasi ini akan difokuskan kepada lagu barat populer. Produk ini diharapkan untuk mengenali lagu barat populer dalam bentuk potongan-potongan atau dalam bentuk keseluruhan lagu secara akurat.

2. How the Product Works

Program kami akan mengambil input berupa *audio file* yang berisi potongan-potongan lirik dari lagu barat populer. Lalu, *audio file* akan kami preprocess terlebih dahulu menggunakan librosa untuk menyamakan sample rate dan menormalisasikan audio agar volume dan pitch menjadi setara agar pretrained model yang digunakan bisa menerjemahkan audio menjadi transkripsi yang tepat. Program kami akan memproses *audio file* yang dimasukkan user dan mengubahnya menjadi teks yang nantinya akan dicocokkan dengan lirik pada lagu barat yang

terdapat dalam dataset yang akan kami gunakan. Produk ini akan dibuat dengan bahasa pemrograman Python. Berikut adalah proses kerja program kami dalam bentuk *bullet point*:

- User akan memasukkan input berupa audio file yang berisi potongan lirik dari lagu - lagu barat populer.
- Input tersebut akan diproses di dalam program/aplikasi lalu diubah menjadi *speech-to-text* yang kemudian akan di cocokan dengan database lagu sampai ditemukan lirik lagu yang sesuai dengan input audio file yang dimasukan ke program/aplikasi.
- Program/aplikasi akan mengeluarkan output judul lagu berupa suara (*text-to-speech*).

3. Methodology

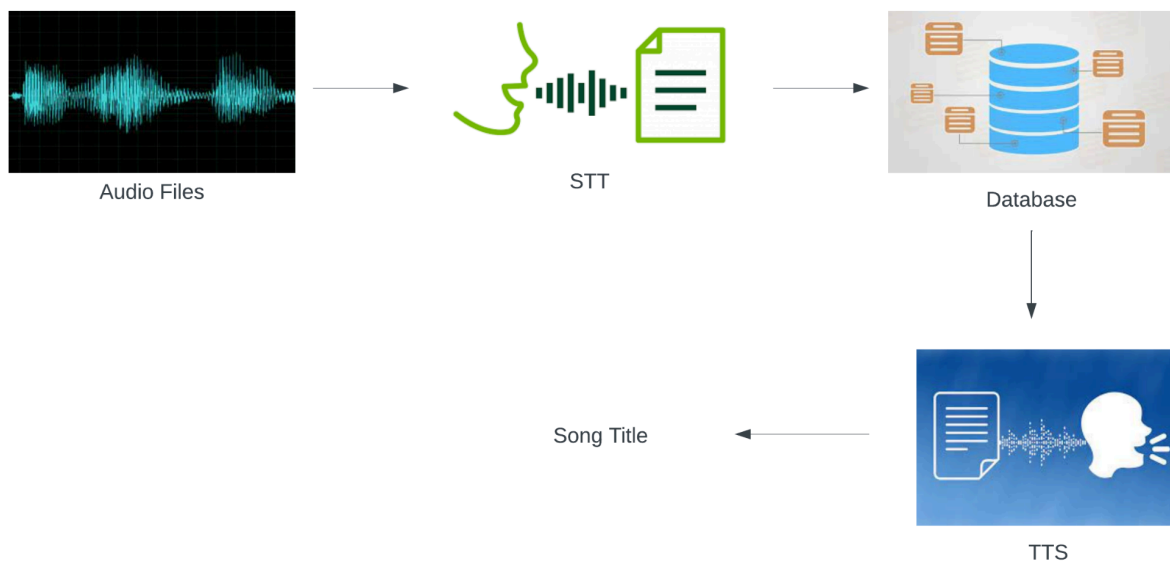


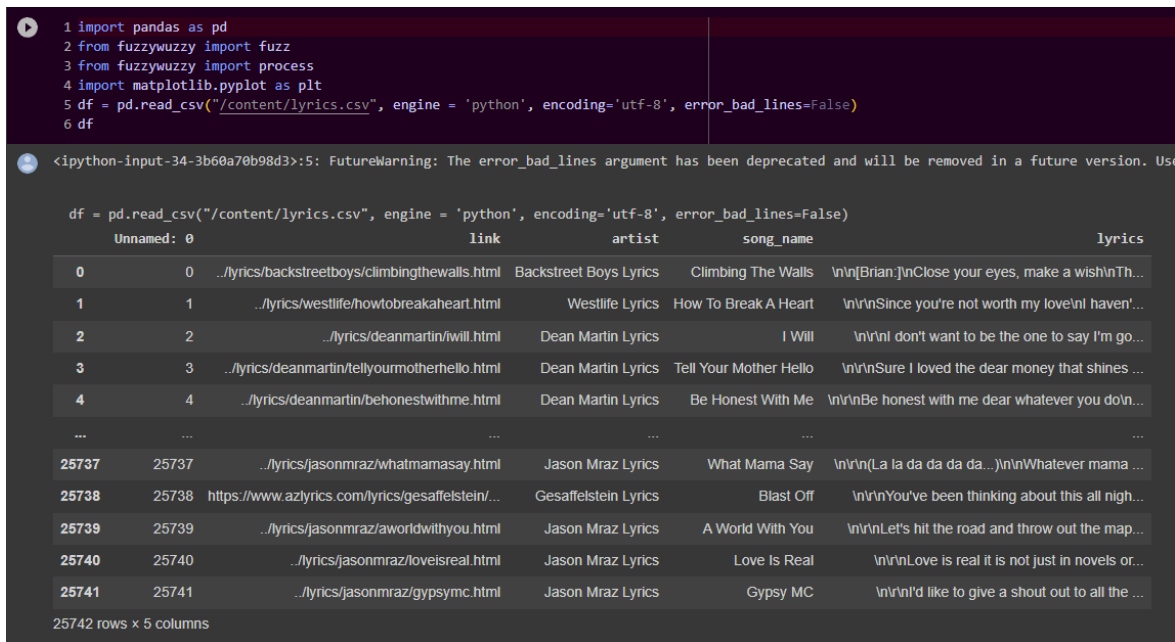
Figure 1. Product Workflow.

a. Dataset

Dataset yang digunakan adalah lirik lagu yang berjudul “*Songs Lyrics*” dari kaggle yang berisikan lagu - lagu barat populer sebanyak 25000 lagu dengan 150 penyanyi. Di dalam dataset terdapat 3 file csv yaitu album_details.csv, lyrics.csv, dan songs_detail.csv. File album_details.csv terdapat 6 kolom yaitu, id lagu, nama penyanyi, judul lagu, tipe lagu, dan tahun terbit. Pada file lyrics.csv, terdapat 5 kolom yang terdiri dari link lirik lagu, nama

penyanyi, judul lagu, dan lirik lagunya. Pada *file* songs_detail.csv, terdapat 5 kolom yaitu, id lagu, nama penyanyi, nama lagu, dan link lirik lagu. Kami hanya akan menggunakan *file* lyrics.csv untuk digunakan sebagai dataset implementasi kami. Berikut adalah link dari dataset yang kami gunakan:

<https://www.kaggle.com/datasets/terminate9298/songs-lyrics?select=lyrics.csv>



```

1 import pandas as pd
2 from fuzzywuzzy import fuzz
3 from fuzzywuzzy import process
4 import matplotlib.pyplot as plt
5 df = pd.read_csv("/content/lyrics.csv", engine = 'python', encoding='utf-8', error_bad_lines=False)
6 df

```

<ipython-input-34-3b60a70b98d3>:5: FutureWarning: The error_bad_lines argument has been deprecated and will be removed in a future version. Use

```

df = pd.read_csv("/content/lyrics.csv", engine = 'python', encoding='utf-8', error_bad_lines=False)

```

Unnamed: 0	link	artist	song_name	lyrics
0	../lyrics/backstreetboys/climbingthewalls.html	Backstreet Boys Lyrics	Climbing The Walls	\n\n[Brian.]nClose your eyes, make a wish\nTh...
1	../lyrics/westlife/howtobreakaheart.html	Westlife Lyrics	How To Break A Heart	\n\nSince you're not worth my love\nI haven't...
2	../lyrics/deanmartin/iwill.html	Dean Martin Lyrics	I Will	\n\nI don't want to be the one to say I'm go...
3	../lyrics/deanmartin/tellyourmotherhello.html	Dean Martin Lyrics	Tell Your Mother Hello	\n\nSure I loved the dear money that shines ...
4	../lyrics/deanmartin/behonestwithme.html	Dean Martin Lyrics	Be Honest With Me	\n\nBe honest with me dear whatever you do\n...
...
25737	../lyrics/jasonmraz/whatmamasay.html	Jason Mraz Lyrics	What Mama Say	\n\n(La la da da da da...)\n\nWhatever mama ...
25738	https://www.azlyrics.com/lyrics/gesaffelstein/...	Gesaffelstein Lyrics	Blast Off	\n\nYou've been thinking about this all nigh...
25739	../lyrics/jasonmraz/aworldwithyou.html	Jason Mraz Lyrics	A World With You	\n\nLet's hit the road and throw out the map...
25740	../lyrics/jasonmraz/loveisreal.html	Jason Mraz Lyrics	Love Is Real	\n\nLove is real it is not just in novels or...
25741	../lyrics/jasonmraz/gypsymc.html	Jason Mraz Lyrics	Gypsy MC	\n\nI'd like to give a shout out to all the ...

25742 rows x 5 columns

Figure 2. Dataset Table.

b. Library

- Fuzzy Wuzzy

Fuzzy Wuzzy merupakan suatu *library* python yang digunakan untuk mengecek kesamaan kata pada suatu *string*. Fuzzy *string matching* menggunakan *Levensthein Distance* untuk mengecek suatu kesamaan string. Lalu, fuzzy *string matching* akan memberikan indikator antara 0-100 yang menunjukkan similaritas antar string [1].

- GTTS

GTTS atau Google Text-to-Speech adalah sebuah *python library* yang menggunakan teknologi Google TOS untuk menginterpretasikan kata-kata secara lisan dengan menggunakan

intonasi, singkatan, dan titik koma yang tepat. Disini, gTTs digunakan untuk mengubah transkripsi teks berupa nama penyanyi dan judul lagu menjadi suatu mp3 yang akan menjadi hasil utama dari proyek ini [2].

- *Speech Recognition*

Library speech recognition merupakan *speech to text library* yang disediakan oleh Google *Speech Recognition*, *CMU Sphinx*, dan lain-lain. API yang disediakan dapat bekerja dengan *engine speech recognition* terbaik dan dapat mentranskripsi sebuah input (*speech*) menjadi tulisan (*text*) sesuai dengan bahasa yang ingin dituju [3].

- *Ffmpeg library*

Ffmpeg library merupakan *open source library* yang digunakan untuk menangani file video dan audio. Dalam hal ini, kami menggunakan *library ffmpeg* untuk mengkonversi audio. Kami menggunakan perpustakaan *ffmpeg* untuk merekam audio di terminal google collab untuk mendapatkan data nyata untuk diproses dalam pencocokan string di *fuzzywuzzy* [4].

4. Result

Pertama, dataset dari kaggle akan di-*import* menggunakan *library pandas* dan dimasukkan dalam sebuah *dataframe*:

0	0	../lyrics/backstreetboys/climbingthewalls.html	Backstreet Boys Lyrics	Climbing The Walls	\n\n[Brian.]Close your eyes, make a wish\nTh...
1	1	../lyrics/westlife/howtobreakaheart.html	Westlife Lyrics	How To Break A Heart	\n\nSince you're not worth my love\nI haven't...
2	2	../lyrics/deanmartin/iwill.html	Dean Martin Lyrics	I Will	\n\nI don't want to be the one to say I'm go...
3	3	../lyrics/deanmartin/tellyourmotherhello.html	Dean Martin Lyrics	Tell Your Mother Hello	\n\nSure I loved the dear money that shines ...
4	4	../lyrics/deanmartin/behonestwithme.html	Dean Martin Lyrics	Be Honest With Me	\n\nBe honest with me dear whatever you do\n...
...
25737	25737	../lyrics/jasonmraz/whatmamasay.html	Jason Mraz Lyrics	What Mama Say	\n\n(La la da da da da...)\n\nWhatever mama ...
25738	25738	https://www.azlyrics.com/lyrics/gesaffelstein/...	Gesaffelstein Lyrics	Blast Off	\n\nYou've been thinking about this all nigh...
25739	25739	../lyrics/jasonmraz/aworldwithyou.html	Jason Mraz Lyrics	A World With You	\n\nLet's hit the road and throw out the map...
25740	25740	../lyrics/jasonmraz/loveisreal.html	Jason Mraz Lyrics	Love Is Real	\n\nLove is real it is not just in novels or...
25741	25741	../lyrics/jasonmraz/gypsymc.html	Jason Mraz Lyrics	Gypsy MC	\n\nI'd like to give a shout out to all the ...
25742 rows x 5 columns					

Figure 3. Dataset Visualization.

Kemudian, dengan menggunakan pustaka ffmpeg di python, kami membuat perekam terminal, sehingga kami dapat merekam audio di *notebook* collab.

```
1 !pip install ffmpeg-python

2 from IPython.display import HTML, Audio
3 from google.colab.output import eval_js
4 from base64 import b64decode
5 import numpy as np
6 from scipy.io.wavfile import read as wav_read
7 import io
8 import ffmpeg
9
10 AUDIO_HTML = """
11 <script>
12 var my_div = document.createElement("DIV");
13 var my_p = document.createElement("P");
14 var my_btn = document.createElement("BUTTON");
15 var t = document.createTextNode("Press to start recording");
16 my_btn.appendChild(t);
17 //my_p.appendChild(my_btn);
18 my_div.appendChild(my_btn);
19 document.body.appendChild(my_div);
20
21 var base64data = 0;
22 var reader;
23 var recorder, gumStream;
24 var recordButton = my_btn;
25
26 var handleSuccess = function(stream) {
27   gumStream = stream;
28   var options = {
29     //bitsPerSecond: 8000, //chrome seems to ignore, always 48k
30     mimeType : 'audio/webm;codecs=opus'
31     //mimeType : 'audio/webm;codecs=pcm'
32   };
33   //recorder = new MediaRecorder(stream, options);
34   recorder = new MediaRecorder(stream);
35   recorder.ondataavailable = function(e) {
36     var url = URL.createObjectURL(e.data);
37     var preview = document.createElement('audio');
38     preview.controls = true;
39     preview.src = url;
40     document.body.appendChild(preview);
41
42     reader = new FileReader();
43     reader.readAsDataURL(e.data);
44     reader.onloadend = function() {
45       base64data = reader.result;
46       //console.log("Inside FileReader:" + base64data);
47     }
48   };
49   recorder.start();
50 };
51
52 recordButton.innerText = "Recording... press to stop";
53
54 navigator.mediaDevices.getUserMedia({audio: true}).then(handleSuccess);
55
56
```

Figure 4. Record Function 1.

```

56
57 function toggleRecording() {
58   if (recorder && recorder.state == "recording") {
59     recorder.stop();
60     gumStream.getAudioTracks()[0].stop();
61     recordButton.innerText = "Saving the recording... pls wait!"
62   }
63 }
64
65 // https://stackoverflow.com/a/951057
66 function sleep(ms) {
67   return new Promise(resolve => setTimeout(resolve, ms));
68 }
69
70 var data = new Promise(resolve=>{
71 //recordButton.addEventListener("click", toggleRecording);
72 recordButton.onclick = ()=>{
73 toggleRecording()
74
75 sleep(2000).then(() => {
76   // wait 2000ms for the data to be available...
77   // ideally this should use something like await...
78   //console.log("Inside data:" + base64data)
79   resolve(base64data.toString())
80
81 });
82
83 }
84 });
85
86 </script>
87 ""
88
89 def get_audio():
90   display(HTML(AUDIO_HTML))
91   data = eval_js("data")
92   binary = b64decode(data.split(',')[1])
93
94   process = (ffmpeg
95     .input('pipe:0')
96     .output('pipe:1', format='wav')
97     .run_async(pipe_stdin=True, pipe_stdout=True, pipe_stderr=True, quiet=True, overwrite_output=
98   )
99   output, err = process.communicate(input=binary)
100
101   riff_chunk_size = len(output) - 8
102   # Break up the chunk size into four bytes, held in b.
103   q = riff_chunk_size
104   b = []
105   for i in range(4):
106     q, r = divmod(q, 256)
107     b.append(r)
108
109   # Replace bytes 4:8 in proc.stdout with the actual size of the RIFF chunk.
110   riff = output[:4] + bytes(b) + output[8:]
111
112   sr, audio = wav_read(io.BytesIO(riff))
113
114   return audio, sr

```

Figure 5. Record Function 2.

Pada kode berikut, kita akan merekam suara manusia dan meletakkannya di file .wav bernama recording.wav yang akan disimpan di runtime notebook. Rekaman ini berisi suara manusia yang mengatakan "it's been a long day, without you my friend."

```
[ ] 1 import scipy

[ ] 1 audio, sr = get_audio()
    2 scipy.io.wavfile.write('recording.wav', sr, audio)
```

Recording... press to stop

Figure 6. Terminal Recorder.

Kemudian, dengan menggunakan *library speech recognition* yang merupakan *library* yang disediakan oleh Google API of Speech ke teks, kami mengubah ucapan yaitu "recording.wav" dan mentranskripsikannya menjadi teks dalam variabel Sample_lyrics. Kami pertama-tama membuat fungsi pengenalan dan menugaskannya ke dalam variabel "r" dan menugaskan file audio ke dalam variabel "suara". Kemudian teks yang ditranskrip akan dimasukkan ke dalam variabel Sample_lyrics. Kemudian, kita membuat kolom baru bernama match_score yang akan disisipkan dengan fuzzy_partial_ratio dari sample_lyrics dan lirik dalam dataset, lalu kita sortir dan tampilkan judul yang paling cocok dengan fitur text to speech menggunakan GTTS.

```
1 from gtts import gTTS
2 from IPython.display import Audio

1 import speech_recognition as sr
2 r = sr.Recognizer()
3 suara = sr.AudioFile('/content/recording.wav')
4 with suara as source:
5     audio = r.record(source)
6 Sample_lyrics = r.recognize_google(audio, language='en-IN')
7 lirik = df[['artist', 'lyrics']].values
8 df['match_score'] = df['lyrics'].apply(lambda x: fuzz.partial_ratio(x, Sample_lyrics))
9 sorted_df = df.sort_values('match_score', ascending=False)
10 sorted_df[0:3]
11 tts = gTTS(f'The most similar song to your search is {(sorted_df.song_name[0:1].values[0])} by {(sorted_df.artist[0:1].values[0])} with a match score of {(sorted_df.match_score[0:1].values[0])}')
12 tts.save('1.wav')
13 sound_file = '1.wav'
14 Audio(sound_file, autoplay=True)
```

0:00 / 0:09

```
[ ] 1 sorted_df[0:3]
```

Unnamed: 0	link	artist	song_name	lyrics	match_score
6499	..lyrics/charlieputh/see-you-again-piano-demos...	Charlie Puth Lyrics	See You Again (Piano Demo Version)	'It's been a long day without you, my fr...	98
6467	https://www.lyrics.com/lyrics/wizkhalifa/see...	Wiz Khalifa Lyrics	See You Again	'Wiz(Charlie Puth)'It's been a long day with...	98
22927	..lyrics/macmiller/right.html	Mac Miller Lyrics	Right	'It'sYeah, it's been a while without your fac...	67

```
[ ] 1 Sample_lyrics
```

'It's been a long day without you my friend'

Figure 7. Product Result

Dari `sorted_df`, kita bisa melihat hasil `match_score` yang diurutkan menaik dari terendah ke tertinggi, skor ini diambil dengan menggunakan pencocokan string fuzzy parsial fuzzy yang menggunakan levenshtein distance untuk mencocokkan beberapa urutan teks dengan `sample_lyrics`. Kami menggunakan fuzzy wuzzy parsial karena jika kami mencocokkan urutan lirik dengan lirik penuh maka skor pertandingan tidak akan sebaik dan liriknya akan sangat cocok. Terakhir akan dioutput judul lagu yang paling cocok dengan rekaman audio dilihat dengan nilai match-score yang paling tinggi.

References

- [1] Rao, P. J., Rao, K. N., & Gokuruboyina, S. (2022). An Experimental Study with Fuzzy-Wuzzy (Partial Ratio) for Identifying the Similarity between English and French Languages for Plagiarism Detection. *International Journal of Advanced Computer Science and Applications*, 13(10). <https://doi.org/10.14569/IJACSA.2022.0131047>
- [2] Subhash, S., Srivatsa, P. N., Siddesh, S., Ullas, A., & Santhosh, B. (2020). Artificial intelligence-based voice assistant. *Proceedings of the World Conference on Smart Trends in Systems, Security and Sustainability, WS4 2020*. <https://doi.org/10.1109/WorldS450073.2020.9210344>
- [3] Hussein, N. A., & Shujaa, M. I. (2019). Smart ambulance: Speed Clearance in the Internet of Things paradigm using Voice Chat. *Advances in Science, Technology and Engineering Systems*, 4(6). <https://doi.org/10.25046/aj040635>
- [4] Malik, M. (2019). Framework For Lossless Data Compression Using Python. *International Journal of Engineering and Computer Science*, 8(03). <https://doi.org/10.18535/ijecs/v8i03.4296>