

LAPORAN AKHIR
PROJECT-BASED LEARNING
MATA KULIAH ALGORITMA DAN PEMROGRAMAN LANJUT
TAHAP 1: RANCANGAN SCRIPT PEMROGRAMAN FUNGSIONAL
DAN/ATAU OOP PADA STUDI KASUS
NATURAL LANGUAGE PROCESSING
KELAS D



**“Analisis Video Youtube Keluhan Pelanggan dengan Menggunakan
Teknologi Automatic Speech Recognition (ASR) dan Analisis
Sentimen”**

DISUSUN OLEH KELOMPOK “I” :

- | | |
|-----------------------------|---------------------------|
| 1. RANGGA LAKSANA ARYANANDA | (21083010036) - KETUA |
| 2. ELLEXIA LEONIE GUNAWAN | (21083010027) - ANGGOTA |
| 3. VALENTINO BELARDO | (21083010109) - ANGGOTA |
| 4. FIRTA MARCELIA | (21083010115) - ANGGOTA |
| 5. CHRISTINA HALIM | (21083010124) - ANGGOTA |

DOSEN PENGAMPU:

TRESNA MAULANA FAHRUDIN, S.S.T., MT (20219930501200)
SUGIARTO, S.KOM., M.KOM (198702142021211001)

PROGRAM STUDI SAINS DATA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”
JAWA TIMUR
2022

KATA PENGANTAR

Puji syukur kehadirat Tuhan yang Maha Esa yang telah memberikan rahmat dan hidayah- Nya sehingga kami dapat menyelesaikan *project based learning* yang berjudul “Analisis Video Youtube Keluhan Pelanggan dengan Menggunakan Teknologi Automatic Speech Recognition (ASR) Untuk Transkrip Otomatis Video” ini dengan tepat pada waktunya.

Adapun tujuan dari *project* ini adalah untuk memenuhi tugas dari Bapak Tresna Maulana F., S.S.T., MT pada mata kuliah Algoritma dan Pemrograman Lanjut di UPN “Veteran” Jawa Timur. Selain itu, makalah ini juga bertujuan untuk menambah wawasan tentang penerapan Natural Language Processing dengan konsep OOP Python bagi para pembaca dan juga bagi penulis.

Kami mengucapkan terima kasih kepada Bapak Tresna Maulana F., S.S.T., MT selaku dosen mata kuliah Algoritma dan Pemrograman Lanjut yang telah memberikan *project* ini sehingga dapat menambah pengetahuan dan wawasan mengenai Natural Language Processing dengan konsep OOP Python. Kami juga mengucapkan terima kasih kepada semua pihak yang telah berpartisipasi dalam pembuatan project ini sehingga kami dapat menyelesaikan project makalah ini.

Kami menyadari, makalah yang kami tulis ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran yang membangun akan kami nantikan demi kesempurnaan project makalah ini.

Surabaya, 23 Maret 2022

(Penyusun)

DAFTAR ISI

KATA PENGANTAR	I
1. BAB 1: PENDAHULUAN	1
1.1 LATAR BELAKANG	1
1.2 PERMASALAHAN	2
1.3 TUJUAN	2
2. BAB II: TINJAUAN PUSTAKA	4
2.1 TEORI PENUNJANG.....	4
2.1.1 Pelanggan dan Keluhan Pelanggan	4
2.1.2 Automatic Speech Recognition	4
2.1.3 Mel Frequency Cepstral Coefficients (MFCC)	5
2.1.4 Transkripsi Otomatis	5
2.1.5 Speech to Text	5
2.1.6 Hidden Markov Model (HMM)	6
2.1.7 Sentiment Analysis	7
2.1.8 VADER Sentiment Analysis	8
2.1.9 Video to WAV	8
2.2 PENELITIAN TERKAIT	9
2.2.1 Speech Recognition Untuk Mengendalikan Pagar Menggunakan Arduino Nano	9
2.2.2 Speech to Text Untuk Bahasa Indonesia	9
2.2.3 Aplikasi Speech Recognition Sebagai Pengenalan Ucapan Tunawicara Menggunakan Google Cloud Speech API Berbasis Android	9
2.2.4 Aplikasi Speech Recognition dan Text to Speech pada Messenger Berbasis Android	10
3. BAB III: METODOLOGI PENELITIAN	12
A. DESAIN SISTEM SECARA UMUM	12
B. DESAIN SISTEM SPEECH RECOGNITION (CARA KERJA SPEECH RECOGNITION)	13
C. RENCANA ANALISIS DAN VISUALISASI	14
4. BAB IV: HASIL DAN PEMBAHASAN	15
4.1 CODE PROGRAM SPEECH RECOGNITION UNTUK MENDETEKSI SUARA SECARA MANUAL.....	15
4.2 CODE PROGRAM UNTUK MENGONVERSI FILE AUDIO WAV KE TEXT	18
4.3 HASIL DETEKSI SUARA MANUAL DENGAN SPEECH RECOGNITION PYTHON	26
4.4 HASIL KONVERSI AUDIO WAV KE TEXT DENGAN SPEECH RECOGNITION PYTHON.....	27
4.5 HASIL ANALISIS DATA DENGAN SENTIMEN ANALISIS	28
4.6 HASIL WORD CLOUD	29
4.7 HASIL UI TRANSCRIBE VIDEO	29

5. BAB V: KESIMPULAN	31
6. DAFTAR PUSTAKA	32
7. LAMPIRAN	33

FINAL

1. BAB 1: PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi yang sangat pesat membawa pengaruh yang besar bagi kehidupan saat ini. Banyak sekali teknologi-teknologi yang sudah berkembang pesat dan banyak diterapkan dalam kehidupan sehari-hari. Teknologi-teknologi ini digunakan oleh manusia untuk mempermudah segala aktivitas yang ada. Salah satu teknologi yang saat ini sudah banyak digunakan adalah teknologi Speech Recognition atau bisa juga disebut dengan Automatic Speech Recognition (ASR). Speech Recognition ini adalah suatu sistem untuk mengenali pola suara manusia yang dimana nanti suara ini dapat ditangkap oleh perangkat komputer dan kemudian perangkat komputer ini dapat mengenali dan memahami kata-kata tersebut. Secara sederhana sistem ini dapat mengubah inputan suara menjadi teks oleh karena itu Speech Recognition terkadang disebut juga speech-to-text (STT). Dengan adanya teknologi ini manusia dapat berinteraksi langsung dengan mesin sehingga dapat berguna untuk membantu kebutuhan manusia sehari-hari.

Dalam era dunia perekonomian, setiap pelaku industri baik jasa maupun barang tentunya tidak akan lepas dengan yang namanya pelayanan pelanggan. Pelayanan pelanggan ini sangat penting sekali karena pelanggan adalah aset yang sangat berharga bagi pelaku industri. Oleh karena itu, kepuasan pelanggan menjadi prioritas utama dalam sebuah bisnis. Untuk mencapai tujuan tersebut maka diperlukannya tindakan lebih lanjut yaitu dengan analisis data-data keluhan pelanggan. Analisis ini dapat dilakukan dengan menggunakan berbagai metode-metode yang ada sehingga nantinya hasil yang didapatkan dapat berupa insight yaitu berupa informasi atau keputusan yang nantinya dapat digunakan oleh stakeholder atau pihak yang berkepentingan.

Keluhan pelanggan menjadi permasalahan serius dalam dunia e-commerce Indonesia.. Hampir 90% keluhan pelanggan di Indonesia ternyata tidak disebabkan permasalahan produk atau barangnya namun karena pengirimannya. Meskipun pertumbuhan e-commerce Indonesia adalah yang paling kuat yaitu sebesar 10,3% selama lima tahun terakhir, pengiriman tetap menjadi tantangan terbesar dengan 36% konsumen menyatakan ketidakpuasan dalam pengalaman pengiriman e-commerce mereka. Hal ini terungkap dalam survei baru oleh Parcel Perform dan iPrice Group,

yang dilakukan dengan lebih dari 80.000 konsumen di Malaysia, Singapura, Indonesia, Vietnam, dan Thailand. Parcel Perform adalah software-as-a-service (SaaS) pelacakan paket yang mampu melacak lebih dari 600 operator logistik secara global. Survei menunjukkan 35% konsumen terus melihat pengiriman sebagai masalah paling besar dalam e-commerce. Selain itu lebih dari 90% keluhan dan tanggapan negatif dari pelanggan terkait dengan keterlambatan pengiriman atau kurangnya komunikasi tentang status pengiriman.

Menurut data dari Yayasan Lembaga Konsumen Indonesia (YLKI) tahun 2020, pengaduan konsumen meningkat menjadi 3.692 dari tahun sebelumnya yang hanya 1.871. Ada 5 konteks pengaduan yang menduduki peringkat teratas. Mengutip data YLKI, lingkup yang dimaksud mencakup produk jasa keuangan sebesar 33,50 persen, e-commerce 12,70 persen, telekomunikasi 8,30 persen, kelistrikan 8,20 persen dan perumahan 5,70 persen. Hal ini menjadi permasalahan serius yang harus segera diselesaikan. Solusi yang tepat untuk mengatasi meningkatnya keluhan konsumen di Indonesia yaitu dengan melakukan survey dan analisis data-data keluhan pelanggan yang ada, agar dapat diambil kebijakan atau keputusan yang tepat kedepannya, agar tingkat keluhan pelanggan atau konsumen di Indonesia dapat menurun.

1.2 Permasalahan

Rumusan permasalahan dari penyusunan makalah ini adalah sebagai berikut :

1. Bagaimana cara untuk meningkatkan layanan customers atau pelanggan suatu perusahaan ?
2. Bagaimana cara menerapkan Teknologi Speech Recognition sebagai transkrip video youtube keluhan pelanggan ?
3. Bagaimana cara menganalisis data keluhan pelanggan dengan menggunakan bahasa pemrograman python ?

1.3 Tujuan

Tujuan dari penyusunan makalah ini adalah sebagai berikut :

1. Mengetahui cara untuk meningkatkan kepuasan pelanggan perusahaan dengan menerapkan teknologi Speech Recognition.
2. Mengetahui cara menerapkan teknologi Speech Recognition untuk transkrip suara video youtube keluhan pelanggan.
3. Mengetahui cara menganalisis data keluhan pelanggan dengan menggunakan bahasa pemrograman python

1.4 Manfaat

Manfaat dari penyusunan tulisan ini adalah sebagai berikut :

1. Manfaat bagi Penulis

Proyek ini dapat dijadikan wadah dalam kontribusi untuk mengatasi permasalahan keluhan pelanggan atau *customers* suatu perusahaan. Selain itu, proyek ini juga dapat dijadikan sebagai wawasan bagi penulis terutama mengenai penerapan Speech Recognition dalam kehidupan sehari-hari.

2. Manfaat bagi Masyarakat

Project ini memberikan dampak langsung bagi masyarakat yaitu menambah wawasan mengenai penerapan teknologi Speech Recognition dalam kehidupan sehari-hari. Selain itu project ini juga dapat membantu menampung suara atau keluhan masyarakat sehingga kepuasan atau keinginan masyarakat dapat terealisasi.

3. Manfaat bagi Perusahaan

Project ini dapat membantu pihak-pihak terkait terutama perusahaan atau pelaku ekonomi. Project ini dapat menjadi salah satu alternatif yang baik dalam meningkatkan kepuasan pelanggan bagi perusahaan.

2. BAB II: TINJAUAN PUSTAKA

2.1 Teori Penunjang

2.1.1 Pelanggan dan Keluhan Pelanggan

Pelanggan adalah individu orang, kelompok, atau perusahaan yang membeli menerima, mengkonsumsi, atau juga menggunakan suatu produk atau jasa yang dimana pelanggan dapat memilih barang atau jasa yang ingin digunakan atau dibeli. Dalam setiap kegiatan ekonomi baik jasa maupun barang peran pelanggan menjadi yang terpenting. Suatu pihak atau perusahaan yang menjual tentunya tidak akan berhasil jika tidak ada pelanggan. Oleh karena itu dalam kegiatan ekonomi pelanggan menjadi prioritas utama.

Perilaku komplain/keluhan pelayanan merupakan suatu proses evaluasi pelanggan terhadap pengalaman konsumsi mereka yang mungkin menyebabkan ketidakpuasan. Keluhan pelanggan selain sebagai bahan perbaikan atau evaluasi terhadap pelayanan yang diberikan juga merupakan suatu hal yang tidak dapat diabaikan. Karena dengan mengabaikan hal tersebut konsumen akan merasa tidak diperhatikan dan pada akhirnya meninggalkan perusahaan. Maka perusahaan/institusi harus benar-benar mencari jalan keluar/solusi sebagai langkah penyelesaian keluhan pelanggan.

2.1.2 Automatic Speech Recognition

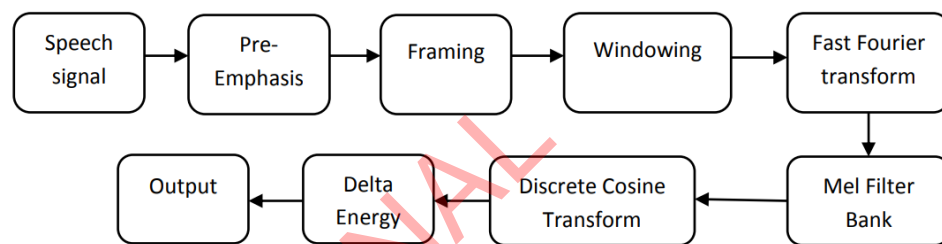
Automatic Speech Recognition (ASR) merupakan suatu pengembangan teknik dan sistem yang memungkinkan komputer untuk menerima masukan berupa kata yang diucapkan. Teknologi ini memungkinkan suatu perangkat untuk mengenali dan memahami kata-kata yang diucapkan dengan cara digitalisasi kata dan mencocokkan sinyal digital tersebut dengan suatu pola tertentu yang tersimpan dalam suatu perangkat. Kata-kata yang diucapkan diubah bentuknya menjadi sinyal digital dengan cara mengubah gelombang suara menjadi sekumpulan angka yang kemudian disesuaikan dengan kode-kode tertentu untuk mengidentifikasi kata-kata tersebut. Hasil dari identifikasi kata yang diucapkan dapat ditampilkan dalam bentuk tulisan atau dapat dibaca oleh perangkat teknologi sebagai sebuah komando untuk melakukan suatu pekerjaan.

Metode yang digunakan untuk melakukan automatic speech recognition adalah metode MFCC untuk feature extraction dan metode HMM untuk pemodelan/pengenalan kata dalam aplikasi Speech to Text (STT).

2.1.3 Mel Frequency Cepstral Coefficients (MFCC)

MFCC (*Mel Frequency Cepstral Coefficients*) merupakan salah satu metode yang banyak digunakan dalam bidang *speech technology*, baik *speaker recognition* maupun *speech recognition*. Metode ini digunakan untuk melakukan *feature extraction*, sebuah proses yang mengkonversikan sinyal suara menjadi beberapa parameter.

MFCC memiliki beberapa tahap untuk ekstraksi suara, berikut diagramnya:



2.1.4 Transkripsi Otomatis

Transkripsi Otomatis adalah proses dimana komputer maupun laptop dapat mendengarkan rekaman suara dan menyalinnya dalam bentuk transkrip yang otomatis dihasilkan dari apa yang didengarnya. Untuk membuat hal ini terjadi, perangkat lunak transkripsi otomatis didukung dengan Teknologi Speech to Text. Cara kerjanya ialah dengan menganalisa rekaman suara detik demi detik, menentukan kata apa yang diucapkan pada tiap detiknya, dan menyimpan setiap kata dalam suatu transkrip rekaman suara.

2.1.5 Speech to Text

Speech to Text (STT) adalah sistem yang dapat mengubah suara menjadi tulisan atau teks. Sistem STT sudah dilatih untuk mengenali suara manusia, mengolahnya, menginterpretasikan nya, dan akan mengonversi suara tersebut menjadi tulisan, sehingga memungkinkan berbagai perangkat seperti komputer dan laptop memahami kebutuhan manusia.

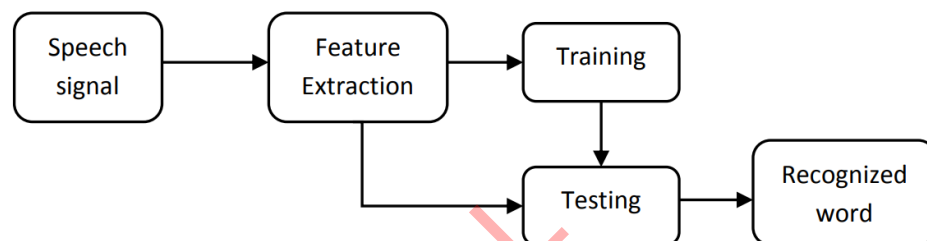
Algoritma untuk Speech to Text adalah sebagai berikut:

1. Sinyal suara (Speech Signal) masuk melalui mikrofon.
2. Fitur ekstrak ciri (Feature extraction) dari sinyal suara.
3. Training dataset dari Feature extraction.
4. Testing dataset dari Feature extraction .
5. Output berupa teks dari sinyal yang dikenali.

Komponen penyusun

1. Sinyal suara.
2. Dataset training.
3. Dataset testing.

Diagram sistem



Cara kerja:

1. Fitur ekstrak ciri dari sinyal suara yang masuk menggunakan MFCC.
2. Training adalah proses clustering dan pembuatan codebook dari sinyal sample menggunakan K-means Clustering.
3. Testing adalah fitur ekstrak ciri dari sinyal test dan menghitung semua likelihood ke sinyal sample yang ada pada codebook.
4. Didapat semua nilai likelihood, nilai yang paling mendekati ke probabilitas sinyal sample dijadikan keluaran berupa teks.

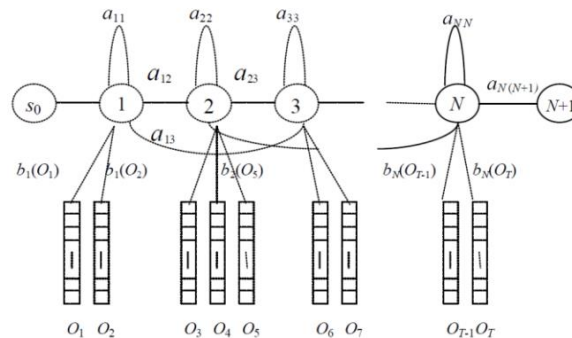
2.1.6 Hidden Markov Model (HMM)

Hidden Markov model (HMM) adalah model statistik di mana sistem yang dimodelkan diasumsikan sebagai proses Markov dengan parameter yang tidak diketahui, dan tantangannya adalah menentukan parameter tersembunyi dari parameter yang dapat diamati. Parameter model yang diekstraksi kemudian dapat digunakan untuk melakukan analisis lebih lanjut, contohnya untuk aplikasi pengenalan pola / ucapan. Dalam HMM, state tidak langsung terlihat, tetapi variabel yang dipengaruhi oleh state terlihat. Setiap state bagian memiliki distribusi probabilitas terhadap kemungkinan keluaran

token. Transisi state juga bersifat probabilistik. Oleh karena itu, urutan token yang dihasilkan oleh HMM memberikan beberapa informasi tentang urutan keadaan. Model HMM lengkap dilambangkan sebagai $\lambda = (A, B, \pi)$.

Parameter Model HMM

1. A, distribusi probabilitas keadaan, $A = \{a_{ij}\}$.
2. B, densitas probabilitas simbol pengamatan, $B = \{b_j(k)\}$.
3. π , distribusi keadaan awal, $\pi = \{\pi_i\}$.



2.1.7 Sentiment Analysis

Menurut KBBI, Sentimen merupakan pendapat atau pandangan yang didasarkan pada perasaan yang berlebih-lebihan terhadap sesuatu. Sedangkan Analisis merupakan aktivitas yang terdiri dari serangkaian kegiatan seperti mengurai, membedakan, dan memilah sesuatu untuk dikelompokkan kembali menurut kriteria tertentu dan kemudian dicari kaitannya lalu ditafsirkan maknanya.

Sentiment Analysis atau yang sering disebut dengan *Opinion Meaning* merupakan salah satu bidang dari *Natural Language Processing* (NLP) yang membangun sistem untuk mengenali dan mengekstraksi opini dalam bentuk teks. Analisis sentimen diberikan oleh pengguna internet pada media sosial untuk memberikan suatu penilaian atau opini pribadi. Tugas analysis sentiment yaitu mengelompokkan teks ke dalam kalimat atau dokumen kemudian menentukan pendapat yang dikemukakan dalam kalimat atau dokumen yang dianalisis apakah bersifat positif, negatif, atau netral.

2.1.8 VADER Sentiment Analysis

VADER (Valence Aware Dictionary and Sentiment Reasoner) merupakan metode analisis lexicon based. Vader akan menganalisis text berdasarkan lexicon (a library) yang menghasilkan class sentiment berupa positif, negatif dan netral dengan tambahan skor total atau compound score. Nilai compound (Compound Score) dihitung dengan menjumlahkan skor valensi setiap kata dalam lexicon, disesuaikan dengan aturan VADER, kemudian dinormalisasi menjadi antara -1 (negatif paling ekstrim) dan +1 (positif paling ekstrim). Nilai compound ini dapat digunakan sebagai satuan standar untuk mengklasifikasikan kalimat dengan ketentuan positif untuk $\text{compound} \geq 0.05$, negatif untuk $\text{compound} \leq -0.05$ dan netral untuk $-0.05 < \text{compound} < 0.05$.

Vader Sentiment Lexicon merupakan salah satu kamus lexicon, kamus ini mempunyai 7.500 tokens yang mengandung kata berbahasa inggris emoticon serta sentimen yang terkait dengan akronim dan inisial.

2.1.9 Video to WAV

Video adalah suatu bentuk teknologi untuk merekam, menangkap, memproses dan mentransmisikan serta mengatur ulang gambar yang bisa bergerak. Video tersebut dapat disimpan menggunakan signal dari film, video, televisi, video tape atau media non komputer lainnya. Setiap frame tersebut dipresentasikan menggunakan sinyal listrik yang disebut dengan gelombang analog atau video komposit yang telah mempunyai komponen-komponen dalam video seperti warna, penerangan dan kesinkronan dari setiap gambarnya.

Sedangkan, WAV adalah format audio file. Ini adalah sumber daya Interchange File Format standar dan menyimpan data dalam potongan-potongan. WAV adalah format audio utama yang digunakan untuk diproses audio pada Windows platform.

Video dan WAV adalah dua file audio yang populer. Terkadang, kemungkinan terdapat keinginan untuk mengkonversi video ke WAV atau sebaliknya untuk mendapatkan kualitas lebih baik. Untuk melakukan hal ini, diperlukan perangkat lunak yang handal yang bekerja secara efisien dan mempertahankan kualitas file audio yang sedang dikonversi.

2.2 Penelitian Terkait

2.2.1 Speech Recognition Untuk Mengendalikan Pagar Menggunakan Arduino Nano

Dari penelitian (Muhammad Iqbal Nurpadilah, 2019) dapat diketahui adanya beberapa tujuan dari penelitian ini adalah sistem pengendali pagar diharapkan dapat menjadi solusi untuk mengendalikan pagar, diharapkan dengan adanya aplikasi ini proses mengendalikan pagar dikendalikan dengan perintah suara, diharapkan dengan adanya alat ini proses pengendalian pagar dapat dikendalikan berdasarkan objek yang melintas. Metodologi penelitian yang digunakan terdiri dari studi literatur, perancangan (design), unit check, implementasi perancangan, dan pengujian. Implementasi hasil dari perancangan hardware menjadi sebuah sistem yang terhubung satu sama lain. Implementasi konfigurasi hardware menjelaskan tentang perancangan perangkat keras dari awal alat sistem pengendali pagar menggunakan perintah suara dengan cara menghubungkan komponen-komponen seperti arduino nano, modul bluetooth HC-05, microservo, dan Motor DC menjadi satu kesatuan.

2.2.2 Speech to Text Untuk Bahasa Indonesia

Adapun tujuan dari penelitian (Teguh Puji Laksono, 2018) adalah membangun Speech to Text untuk mengambil text pada rekaman suara Bahasa Indonesia dan Bahasa Jawa. Pada penelitian ini menggunakan metode Mel Frequency Cepstral Coefficient (MFCC), Deep Learning (Deep Neural Network) dan Connectionist Temporal Classification (CTC). Hasil pengujian speech to text menunjukkan akurasi 65% pada data Bahasa Indonesia dan 57% pada data Bahasa Jawa. Pada data uji 50 kata. Hal tersebut dapat disimpulkan speech to text dengan metode deep learning dapat digunakan sebagai solusi dari mengubah rekaman suara menjadi text.

2.2.3 Aplikasi Speech Recognition Sebagai Pengenalan Ucapan Tunawicara Menggunakan Google Cloud Speech API Berbasis Android

Dari penelitian (Angga Kurniawan, 2017) terdapat beberapa tujuan, yaitu membuat aplikasi pengenalan dan penerjemah ucapan tunawicara yang dapat membantu berkomunikasi lebih mudah dengan orang normal,

menyediakan teknologi komunikasi yang mudah untuk digunakan tunawicara dengan menggunakan aplikasi berbasis Android, dan melakukan pengembangan teknologi tools speech recognition dari Google. Metodologi penelitian meliputi pengumpulan data hingga terbentuknya sebuah perangkat lunak. Berdasarkan hasil dari penelitian yang penulis amati, Aplikasi Speech Recognition dengan menggunakan Google Cloud Speech dapat mengenali dan menerjemahkan ucapan tunawicara. Recognition rate untuk tunawicara 80% dan untuk pengenalan ucapan orang normal 100% dalam penyebutan digit angka 1 – 10. Ada tiga faktor yang mempengaruhi recognition rate: nada, pengucapan(pronunciation), dan kecepatan bicara.

2.2.4 Aplikasi Speech Recognition dan Text to Speech pada Messenger Berbasis Android

Adapun tujuan dari penelitian (Irfan Rinaldy, 2017) adalah untuk membangun aplikasi messenger yang mendukung fitur speech recognition dan text to speech berbasis Android yang dapat membantu meningkatkan mobilitas pengguna smartphone. Metode penelitian yang digunakan meliputi dua metode, yaitu metode pengumpulan data dan metode perancangan aplikasi. Berdasarkan implementasi dan hasil pengujian diantaranya black box testing dan white box testing, serta hasil kuesioner ke target pengguna, maka tujuan penelitian untuk membangun aplikasi messenger yang mendukung fitur speech recognition dan text to speech berbasis Android yang dapat membantu meningkatkan mobilitas pengguna smartphone, dinyatakan telah tercapai karena mampu memanfaatkan teknologi speech recognition dan text to speech dalam proses berkirim pesan, khususnya dalam proses menginput pesan dan membaca pesan yang telah diterima, sehingga berkirim pesan tetap dapat dilakukan meskipun pengguna berada dalam kondisi tidak memungkinkan untuk mengetik pesan maupun membaca pesan secara langsung. Dikarenakan aplikasi ini dikhususkan untuk berkirim pesan singkat (SMS). maka aplikasi ini tidak bergantung pada koneksi internet untuk menjalankan fungsi konversi suara pada saat menginput pesan maupun membaca pesan (offline). Akan tetapi penggunaan bahasa masih terbatas pada bahasan baku dan tidak bisa

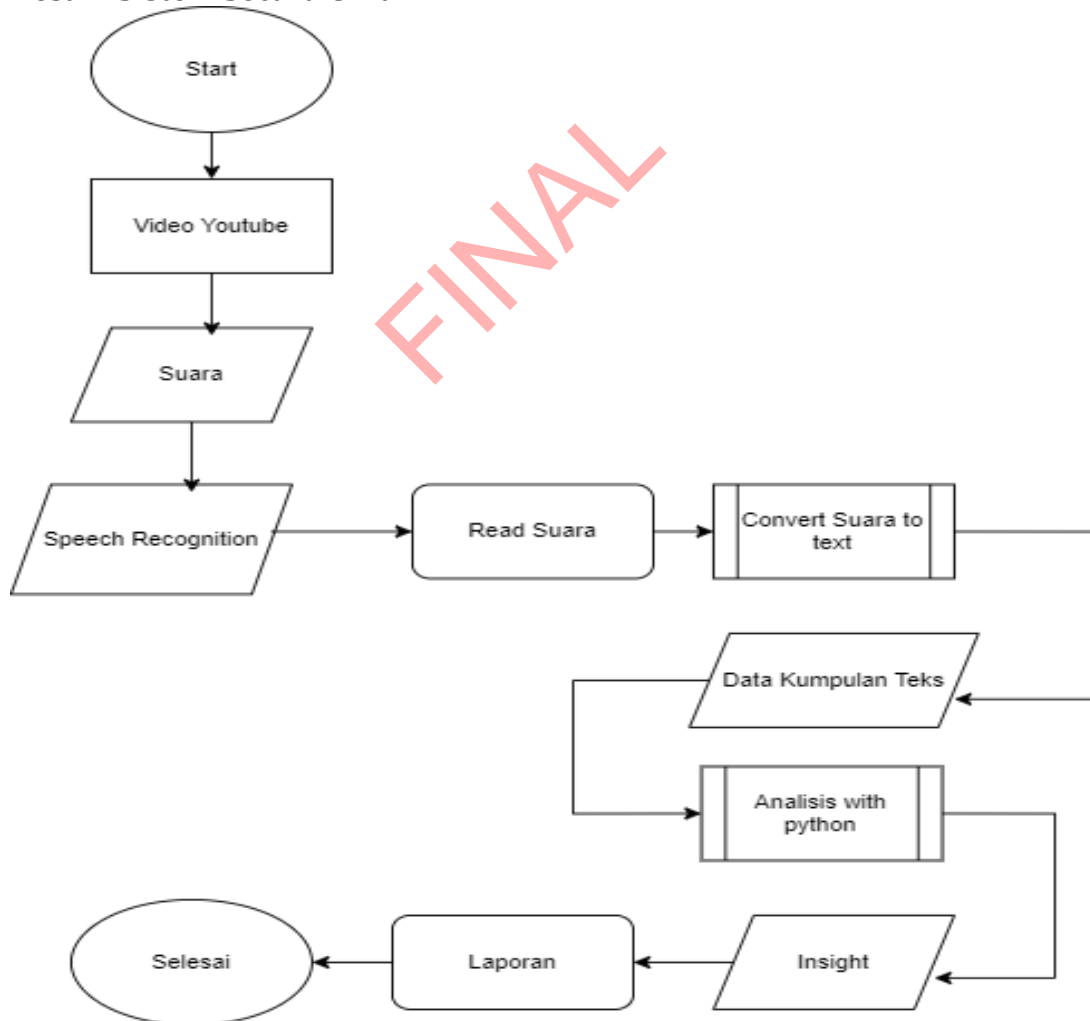
mengakomodasi singkatan, sehingga masih memungkinkan terdapat kekeliruan dalam mengonversi pesan.

FINAL

3. BAB III: METODOLOGI PENELITIAN

Secara umum pada project analisis keluhan pelanggan ini data-data yang digunakan untuk analisis adalah video yang bersumber dari platform Youtube. Alasan menggunakan Youtube adalah karena data dapat diambil dengan mudah dan efisien. Data yang diperlukan dalam project ini berupa audio atau suara yang kemudian akan diubah menjadi sekumpulan teks dengan menerapkan Speech Recognition dengan Python. Setelah data terkumpul selanjutnya adalah dilakukannya analisis terhadap teks tersebut. Di sini kami akan menganalisa data-data tersebut dengan menggunakan library vaderSentiment pada Python. Dari analisis tersebut akan didapatkan analisis berupa positif, netral, dan negatif suatu kalimat.

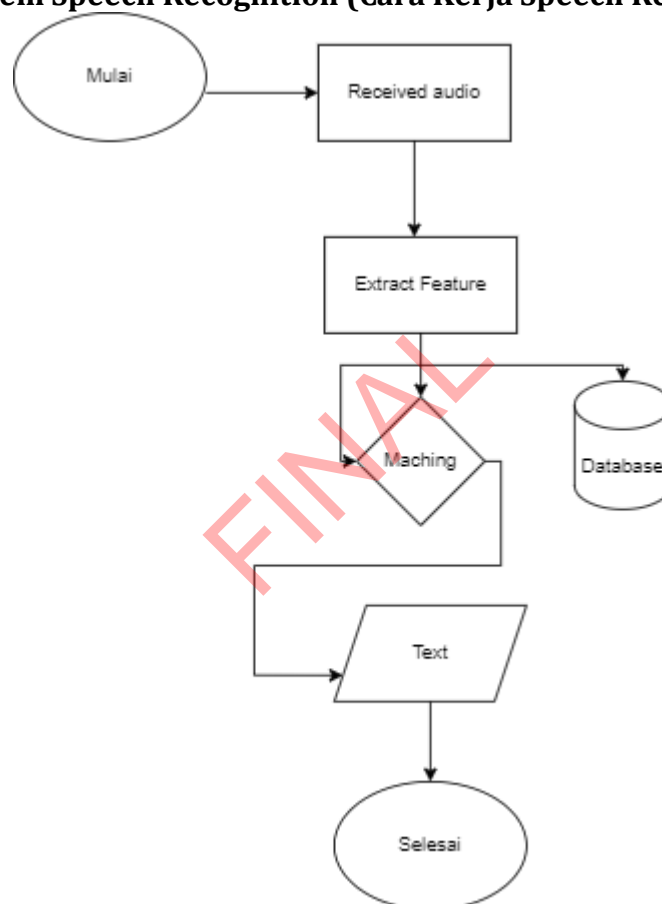
A. Desain Sistem Secara Umum



Gambar 1. Desain sistem secara umum

Pada project analisis keluhan pelanggan ini sumber suara atau audio yang digunakan berasal dari Youtube. Suara dari video Youtube ini kemudian akan diubah menjadi bentuk wav. Suara dalam bentuk wav ini kemudian akan dibaca oleh sistem atau perangkat speech recognition yang kemudian akan dikonversi menjadi sebuah kumpulan-kumpulan teks atau kalimat. Data-data kumpulan teks tersebut kemudian akan dianalisis menggunakan library pada Python yang nantinya hasilnya dapat berupa sebuah insight atau keputusan yang kemudian akan didokumentasikan atau dituliskan di dalam sebuah laporan.

B. Desain Sistem Speech Recognition (Cara Kerja Speech Recognition)



Gambar 2. Desain sistem speech recognition

Tahap pertama dalam proses speech recognition adalah penerimaan masukan yaitu sumber suara diterima melalui media perantara. Kemudian yaitu tahap kedua atau biasa disebut tahap ekstraksi. Pada tahap ini terjadi penyimpanan masukan yang berupa suara dan sekaligus pembuatan basis data. Terakhir sebelum data dikonversi menjadi teks masukan atau audio tersebut harus melalui tahap perbandingan. Pada tahap ini terjadi pencocokan data baru dengan data suara (pencocokan tata bahasa) pada pola. Tahap ini dimulai dengan proses konversi sinyal suara digital hasil dari

proses ekstraksi ke dalam bentuk spektrum suara yang akan dianalisa dengan membandingkannya dengan pola suara pada basis data. Sebelumnya, data suara masukan dipilah-pilah dan diproses satu per satu berdasarkan urutannya. Pemilihan ini dilakukan agar proses analisis dapat dilakukan secara paralel.

C. Rencana Analisis dan Visualisasi

Analisis dan visualisasi yang akan dilakukan pada project ini yaitu dengan menggunakan python. Dalam project ini kami memanfaatkan fungsi bawaan dari python untuk membaca data, melakukan visualisasi, dan analisis. Alasan kami menggunakan visualisasi dan analisis dengan python karena lebih efektif dan fungsi python yang kompleks sehingga dapat digunakan lebih mudah.

Analisis pada project ini akan menggunakan library vaderSentiment pada Python. Setiap hasil transkripsi dari file wav akan dianalisis seberapa besar unsur positif, netral, dan negatif suatu kalimat.

Visualisasi pada project ini akan menggunakan library wordcloud pada Python. Dengan visualisasi ini, akan ditampilkan kata-kata yang banyak muncul dalam suatu video.

FINAL

4. BAB IV: HASIL DAN PEMBAHASAN

4.1 Code program Speech Recognition untuk mendeteksi suara secara manual.

```
import speech_recognition as sr
def main():
    r = sr.Recognizer()

    with sr.Microphone() as source:
        r.adjust_for_ambient_noise(source)

        print("Please say something")

        audio = r.listen(source)

        print("Recognizing Now .... ")

    # recognize speech using google

    try:
        print("You have said \n" + r.recognize_google(audio))
        print("Audio Recorded Successfully \n ")

    except Exception as e:
        print("Error tidak ada suara terdeteksi, mohon mencoba kembali : " +
str(e))

    # write audio
    with open("recorded.wav", "wb") as f:
        f.write(audio.get_wav_data())

if __name__ == "__main__":
    main()
```

Gambar 3. Potongan *script* pada python sequential

Sebelum membuat kode programnya langkah pertama agar kode dapat berjalan dengan baik yaitu dengan menginstall pip python yang diperlukan seperti pip install SpeechRecognition. Untuk menginstall pip tersebut dapat menggunakan prompt perintah atau terminal di perangkat komputer. Pip Speech Recognition tidak dapat bekerja sendiri jika tidak ada audio yang dimasukkan sehingga diperlukan pip PyAudio untuk menangkap suara lewat microphone. Pip PyAudio dapat diinstall menggunakan perintah install PyAudio. Setelah semua pip berhasil di install langkah berikutnya adalah mengimport Speech Recognition as sr.

Dalam penulisan kode langkah pertama yaitu dengan pembuatan fungsi dengan

def main(). Setelah itu membuat Recognizer Class dengan sr.Recognizer() yang dimana perintah ini juga akan dikaitkan dengan beberapa fungsi dasar lain untuk menangkap suara misalnya, listen() dan record(). Untuk menulis perintah Recognizer dapat didefinisikan dalam variabel r . Perintah Recognizer() ini memiliki 7 metode untuk mengenali atau menerjemahkan audio yang ditangkap dalam bentuk teks.

Setelah membuat perintah Recognize() seperti pada langkah sebelumnya, selanjutnya adalah membuat Microphone Class dengan perintah sr.Microphone() dan digabung dengan fungsi with.jika di breakdown fungsi with satu-persatu maka, yang pertama perintah sr.Microphone didefinisikan sebagai source untuk memudahkan dalam penulisan, kemudian fungsi print berguna untuk memberikan keterangan atau perintah kepada pembicara untuk mengucapkan kata atau kalimat melalui microphone, sedangkan audio = r.listen(source) berfungsi untuk menangkap audio suara yang diucapkan.

Setelah membuat microphone class, selanjutnya membuat fungsi kondisi try except. fungsi ini dibuat untuk mendeteksi apakah program yang dibuat berhasil menerjemahkan ucapan ke dalam teks atau malah terjadi error, dalam kondisi try, diisi dengan perintah r.recognize_google(audio) yang artinya perintah atau metode ini yang akan menerjemahkan ucapan yang ditangkap, jika berhasil maka akan di print you said: text, dengan text adalah variabel yang mendefinisikan ucapan yang ditangkap.

Untuk except diisi ini berguna ketika ucapan tidak dapat pahami oleh mesin, maka akan di print ("Error tidak ada suara yang terdeteksi mohon coba kembali") artinya bahwa mesin tidak dapat menampilkan hasil apapun.

Program atau kode terakhir yaitu write audio yang dimana kode ini digunakan untuk merekam dan menyimpan file audio atau suara secara otomatis menjadi file wav yang tersimpan di perangkat.

```

import speech_recognition as sr
import time

class SpeechRecognition():
    def __init__(self):
        self.r = sr.Recognizer()
        masukan_suara = self.r

    def record_audio(self, ask=False):
        with sr.Microphone() as source:
            if ask:
                print(ask)
            audio = self.r.listen(source)
            voice_data = "
            try:
                voice_data = self.r.recognize_google(audio)
            except sr.UnknownValueError:
                print('Suara tidak dapat dikenali, mohon coba kembali')
            except sr.RequestError:
                print('Mohon maaf Recognizer tidak terhubung')
            print(f">> {voice_data.lower()}")
            return voice_data.lower()

    def init_speech_recognizer(self):
        time.sleep(1)
        print('Ucapkan sesuatu')

        while 1:
            voice_data = self.record_audio("Recording")
            print("Done")

speechObj = SpeechRecognition()
speechObj.init_speech_recognizer()

```

Gambar 3.1 Potongan *script* pada python dengan konsep OOP dan Functional

4.2 Code program untuk mengkonversi file audio wav ke text

```
import speech_recognition as sr

r=sr.Recognizer()

text=sr.AudioFile('Headling Complaints Hotel Service_6.wav')
with text as source:
    audio=r.record(source)
    hasil= str(audio)
val=r.recognize_google(audio)
print(val,end='\n')
```

Gambar 4. Potongan *script* pada python

Sama halnya dengan Speech Recognition untuk mendeteksi suara otomatis pada Speech Recognition untuk mengkonversi file audio wav ke text juga memerlukan pip Speech Recog dan PyAudio. Dan juga harus mengimport Speech Recognition terlebih dahulu sebelum meneruskan kodenya. Setelah itu membuat Recognizer Class dengan `sr.Recognizer()` yang dimana perintah ini juga akan dikaitkan dengan beberapa fungsi dasar lain untuk menangkap suara misalnya, `listen()` dan `record()`. Untuk menulis perintah Recognizer dapat didefinisikan dalam variabel `r`. Perintah `Recognizer()` ini memiliki 7 metode untuk mengenali atau menerjemahkan audio yang ditangkap dalam bentuk teks.

Setelah itu mendeklarasikan variabel `text` dengan perintah `sr.AudioFile("nama filenya.Wav")` yang dimana kode ini berguna untuk membaca file wav audio yang tersimpan di perangkat. Untuk dapat menangkap file wav audionya maka file ini harus tersimpan di penyimpanan kode programnya dan juga harus menuliskan nama yang sesuai antara di perangkat dan juga di kode programnya.

Berikutnya `r.record(source)` ini berfungsi untuk merecord data audio dan nantinya akan diolah dengan metode Speech Recognition dengan perintah `r.recognize_google(audio/file atau data audio yang sudah tersimpan atau terecord tadi)` Terakhir yaitu dengan membuat perintah `print` untuk mencetak teks yang sudah terkonversi.

```

import wave, contextlib, math, time
import speech_recognition as sr
from moviepy.editor import AudioFileClip
from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtWidgets import QFileDialog, QMessageBox
from PyQt5.QtCore import QThread, pyqtSignal
class Ui_MainWindow(object):
    """Fungsi Tampilannya dengan GUI."""
    def __init__(self):
        """Inisialisasikan function."""
        self.mp4_file_name = ""
        self.output_file = ""
        self.audio_file = "speech.wav"
    def setupUi(self, MainWindow):
        """Definisikan visual komponen dan posisi."""
        # Main window
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(653, 836)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.label = QtWidgets.QLabel(self.centralwidget)
        self.label.setGeometry(QtCore.QRect(50, 20, 161, 41))
        # Selected video file label
        font = QtGui.QFont()
        font.setPointSize(14)
        self.label.setFont(font)
        self.label.setObjectName("label")
        self.selected_video_label = QtWidgets.QLabel(self.centralwidget)
        self.selected_video_label.setGeometry(QtCore.QRect(230, 20, 371, 41))
        font = QtGui.QFont()
        font.setPointSize(8)
        self.selected_video_label.setFont(font)
        self.selected_video_label.setFrameShape(QtWidgets.QFrame.Box)
        self.selected_video_label.setText("")
        self.selected_video_label.setObjectName("selected_video_label")
        self.label_3 = QtWidgets.QLabel(self.centralwidget)
        self.label_3.setGeometry(QtCore.QRect(50, 90, 161, 41))
        # Transcribed text box
        font = QtGui.QFont()
        font.setPointSize(14)
        self.label_3.setFont(font)
        self.label_3.setObjectName("label_3")
        self.transcribed_text = QtWidgets.QTextBrowser(self.centralwidget)
        self.transcribed_text.setGeometry(QtCore.QRect(230, 320, 381, 431))
        self.transcribed_text.setObjectName("transcribed_text")
        self.label_5 = QtWidgets.QLabel(self.centralwidget)
        self.label_5.setGeometry(QtCore.QRect(230, 280, 161, 41))
        font = QtGui.QFont()
        font.setPointSize(14)
        self.label_5.setFont(font)

```

```

        self.label_5.setObjectName("label_5")
        self.transcribe_button = QtWidgets.QPushButton(self.centralwidget)
        self.transcribe_button.setEnabled(False)
        self.transcribe_button.setGeometry(QtCore.QRect(230, 150, 221, 81))
        # Transcribe button
        font = QtGui.QFont()
        font.setPointSize(14)
        self.transcribe_button.setFont(font)
        self.transcribe_button.setObjectName("transcribe_button")
        self.transcribe_button.clicked.connect(self.process_and_transcribe_audio)
        # progress bar
        self.progress_bar = QtWidgets.QProgressBar(self.centralwidget)
        self.progress_bar.setGeometry(QtCore.QRect(230, 250, 381, 23))
        self.progress_bar.setProperty("value", 0)
        self.progress_bar.setObjectName("progress_bar")
        self.message_label = QtWidgets.QLabel(self.centralwidget)
        self.message_label.setGeometry(QtCore.QRect(0, 760, 651, 21))
        # Message label (for errors and warnings)
        font = QtGui.QFont()
        font.setPointSize(8)
        self.message_label.setFont(font)
        self.message_label.setFrameShape(QtWidgets.QFrame.Box)
        self.message_label.setText("")
        self.message_label.setObjectName("message_label")
        self.output_file_name = QtWidgets.QPlainTextEdit(self.centralwidget)
        self.output_file_name.setGeometry(QtCore.QRect(230, 90, 371, 41))
        # Output file name
        font = QtGui.QFont()
        font.setPointSize(14)
        self.output_file_name.setFont(font)
        self.output_file_name.setObjectName("output_file_name")
        # Menubar options
        MainWindow.setCentralWidget(self.centralwidget)
        self.menubar = QtWidgets.QMenuBar(MainWindow)
        self.menubar.setGeometry(QtCore.QRect(0, 0, 653, 21))
        self.menubar.setObjectName("menubar")
        self.menuFile = QtWidgets.QMenu(self.menubar)
        self.menuFile.setObjectName("menuFile")
        self.menuAbout = QtWidgets.QMenu(self.menubar)
        self.menuAbout.setObjectName("menuAbout")
        MainWindow.setMenuBar(self.menubar)
        self.statusbar = QtWidgets.QStatusBar(MainWindow)
        self.statusbar.setObjectName("statusbar")
        MainWindow.setStatusBar(self.statusbar)
        self.actionOpen_mp4_video_recording = QtWidgets.QAction(MainWindow)

self.actionOpen_mp4_video_recording.setObjectName("actionOpen_mp4_video_recording")
        self.actionOpen_mp4_video_recording.triggered.connect(self.open_audio_file)
        self.actionAbout_vid2text = QtWidgets.QAction(MainWindow)
        self.actionAbout_vid2text.setObjectName("actionAbout_vid2text")
        self.actionAbout_vid2text.triggered.connect(self.show_about)

```



```

self.actionNew = QtWidgets.QAction(MainWindow)
self.actionNew.setObjectName("actionNew")
self.actionNew.triggered.connect(self.new_project)
self.menuFile.addAction(self.actionOpen_mp4_video_recording)
self.menuFile.addAction(self.actionNew)
self.menuAbout.addAction(self.actionAbout_vid2text)
self.menubar.addAction(self.menuFile.menuAction())
self.menubar.addAction(self.menuAbout.menuAction())
self.retranslateUi(MainWindow)
QtCore.QMetaObject.connectSlotsByName(MainWindow)
def retranslateUi(self, MainWindow):
    Metode untuk menerjemahkan UI nya.
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow"))
    self.label.setText(_translate("MainWindow", "Selected video file:"))
    self.label_3.setText(_translate("MainWindow", "Output file name:"))
    self.label_5.setText(_translate("MainWindow", "Transcribed text:"))
    self.transcribe_button.setText(_translate("MainWindow", "Transcribe"))
    self.output_file_name.setPlaceholderText(_translate("MainWindow",
"interview1.txt"))
    self.menuFile.setTitle(_translate("MainWindow", "File"))
    self.menuAbout.setTitle(_translate("MainWindow", "About"))
    self.actionOpen_mp4_video_recording.setText(_translate("MainWindow",
"Open mp4 video recording"))
    self.actionAbout_vid2text.setText(_translate("MainWindow", "About
vid2text"))
    self.actionNew.setText(_translate("MainWindow", "New"))
def open_audio_file(self):
    Open the audio (*.mp4) file.
    file_name = QFileDialog.getOpenFileName()
    if file_name[0][-3:] == ".mp4":
        self.transcribe_button.setEnabled(True)
        self.mp4_file_name = file_name[0]
        self.message_label.setText("")
        self.selected_video_label.setText(file_name[0])
    else:
        self.message_label.setText("Please select an *.mp4 file")
def convert_mp4_to_wav(self):
    Convert the mp4 video file into an audio file.
    self.message_label.setText("Converting mp4 to audio (*.wav)...")
    self.convert_thread = convertVideoToAudioThread(self.mp4_file_name,
self.audio_file)
    self.convert_thread.finished.connect(self.finished_converting)
    self.convert_thread.start()
def get_audio_duration(self, audio_file):
    Determine the length of the audio file.
    with contextlib.closing(wave.open(audio_file, 'r')) as f:
        frames = f.getnframes()
        rate = f.getframerate()
        duration = frames / float(rate)

```

```

        return duration
def transcribe_audio(self, audio_file):
    """Transcribe the audio file."""
    total_duration = self.get_audio_duration(audio_file) / 10
    total_duration = math.ceil(total_duration)
    self.td = total_duration
    if len(self.output_file_name.toPlainText()) > 0:
        self.output_file = self.output_file_name.toPlainText()
    else:
        self.output_file = "my_speech_file.txt"
    # Use thread to process in the background and avoid freezing the GUI
    self.thread = transcriptionThread(total_duration, audio_file, self.output_file)
    self.thread.finished.connect(self.finished_transcribing)
    self.thread.change_value.connect(self.set_progress_value)
    self.thread.start()

def finished_converting(self):
    """Reset message text when conversion is finished."""
    self.message_label.setText("Transcribing file...")
    self.transcribe_audio(self.audio_file)
def finished_transcribing(self):
    """This run when transcription finished to tidy up UI."""
    self.progress_bar.setValue(100)
    self.transcribe_button.setEnabled(True)
    self.message_label.setText("")
    self.update_text_output()
    #self.sentimen_analis()
    #self.wordcloud()

def set_progress_value(self, val):
    """Update progress bar value."""
    increment = int(math.floor(100*(float(val)/self.td)))
    self.progress_bar.setValue(increment)
def process_and_transcribe_audio(self):
    """Process the audio into a textual transcription."""
    self.transcribe_button.setEnabled(False)
    self.message_label.setText("Converting mp4 to audio (*.wav)...")
    self.convert_mp4_to_wav()
def update_text_output(self):
    """Update the text box with the transcribed file."""
    f = open(self.output_file, "r")
    tr = self.transcribed_text.setText(f.read())
    return tr

    f.close()
# def sentimen_analis(self):
#     sentiment_result=[]
#     #Sentiment analysis
#     # sent_list = [self.tr]
#     # analyser = SentimentIntensityAnalyzer()

```

```

# print(sent_list)

#for i in sent_list:
#    sentiment_result.append(analyser.polarity_scores(i))
#print(sentiment_result)
#self.sentiment_text.setText(sentiment_result)

#def wordcloud(self):
#    comment_words = "
#    stopwords = set(STOPWORDS)

#    wordcloud = WordCloud(width=800, height=800,
#                            # background_color='white',
#                            # stopwords=stopwords,
#                            # min_font_size=10).generate(self.output_file)

#    #plot the WordCloud image
#    plt.figure(figsize=(8, 8), facecolor=None)
#    plt.imshow(wordcloud)
#    plt.axis("off")
#    plt.tight_layout(pad=0)

#    plt.show()
def new_project(self):
    """Clear existing fields of data."""
    self.message_label.setText("")
    self.transcribed_text.setText("")
    self.selected_video_label.setText("")
    self.output_file_name.document().setPlainText("")
    self.sentiment_text.setText("")
    self.progress_bar.setValue(0)
def show_about(self):
    """Show about message box."""
    msg = QMessageBox()
    msg.setWindowTitle("About vid2speech")
    msg.setText(" Created by Dr. Alan Davies,\n Senior Lecturer,\n Health Data Science,\n Manchester University, UK")
    msg.setIcon(QMessageBox.Information)
    msg.exec_()
class convertVideoToAudioThread(QThread):
    """Thread to convert mp4 video file to wav file."""
    def __init__(self, mp4_file_name, audio_file):
        """Initialization function."""
        QThread.__init__(self)
        self.mp4_file_name = mp4_file_name
        self.audio_file = audio_file
    def __del__(self):
        """Destructor."""
        self.wait()

```

```

def run(self):
    """Run video conversion task."""
    audio_clip = AudioFileClip(self.mp4_file_name)
    audio_clip.write_audiofile(self.audio_file)
class transcriptionThread(QThread):
    """Thread to transcribe file from audio to text."""
    change_value = pyqtSignal(int)
    def __init__(self, total_duration, audio_file, output_file):
        """Initialization function."""
        QThread.__init__(self)
        self.total_duration = total_duration
        self.audio_file = audio_file
        self.output_file = output_file
    def __del__(self):
        """Destructor."""
        self.wait()
    def run(self):
        """Run transcription, audio to text."""
        r = sr.Recognizer()
        for i in range(0, self.total_duration):
            try:
                with sr.AudioFile(self.audio_file) as source:
                    audio = r.record(source, offset=i*10, duration=10)
                    f = open(self.output_file, "a")
                    f.write(r.recognize_google(audio))
                    f.write(" ")
                self.change_value.emit(i)
            except:
                print("Unknown word detected...")
                continue
            f.close()
if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())

```

Gambar 5. Potongan script UI Transcribe pada python

Kodingan Transcript, word cloud, dan sentimen analisis pada python:

```
import moviepy.editor as mp
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd
import speech_recognition as sr
from textblob import Sentence
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

my_clip = mp.VideoFileClip(input("Silahkan Masukkan File Video.mp4: "))
my_clip.audio.write_audiofile(r"my_result.wav")

r=sr.Recognizer()

text= sr.AudioFile(r"my_result.wav")
with text as source:
    audio=r.record(source)
    hasil= str(audio)
Sentence=r.recognize_google(audio)
print(Sentence,end='\n')
file = open("data.txt", "w")
file.write(Sentence)
file.close()
#Sentiment analysis
sent_list=[Sentence]
analyser=SentimentIntensityAnalyzer()

for i in sent_list:
    print(analyser.polarity_scores(i))

comment_words = "
stopwords = set(STOPWORDS)

wordcloud = WordCloud(width=800, height=800,
                        background_color='white',
                        stopwords=stopwords,
                        min_font_size=10).generate(Sentence)

# plot the WordCloud image
plt.figure(figsize=(8, 8), facecolor=None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad=0)

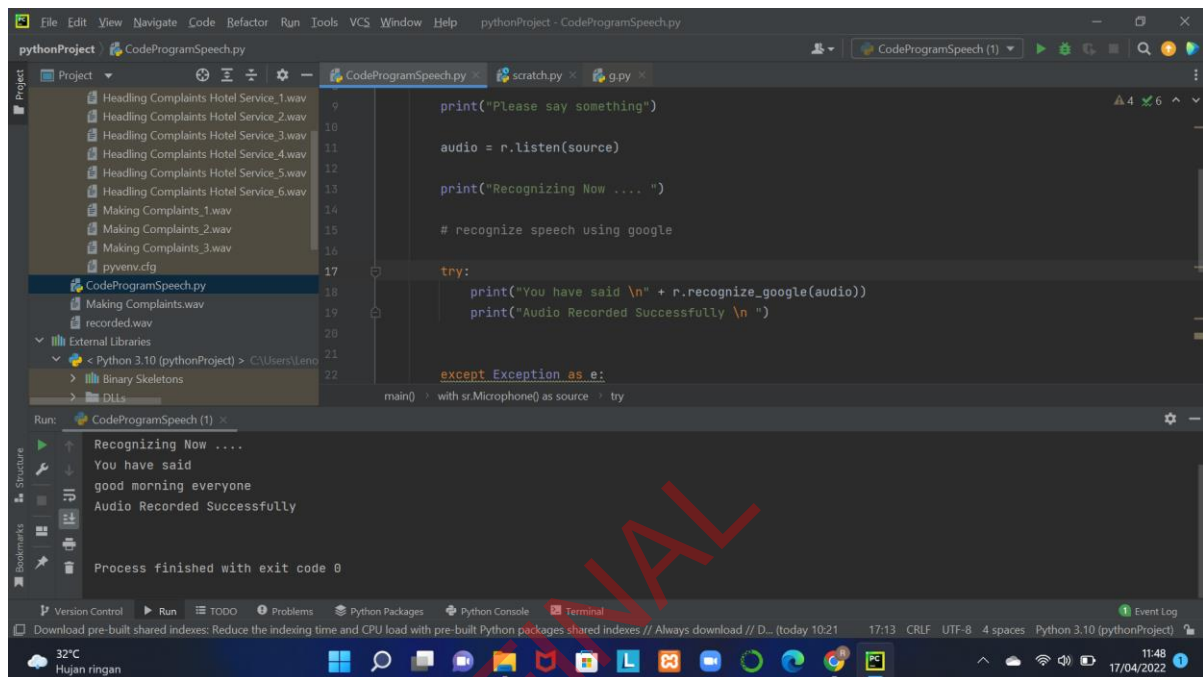
plt.show()
```

Gambar 6. Potongan *script* Transcript, Sentimen analisis, dan Word Cloud pada python

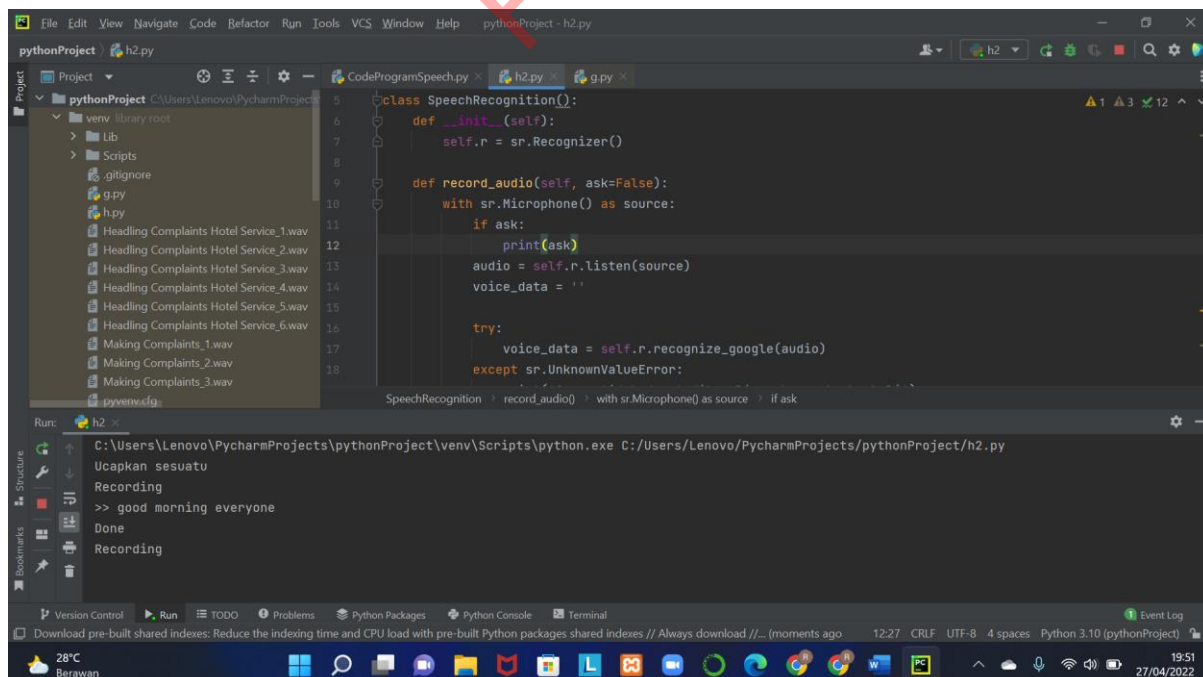
4.3 Hasil Deteksi suara manual dengan Speech Recognition Python

Hasil dari program deteksi suara manual dengan Speech Recognition ini berupa kalimat atau text yang telah terecord di python. Selain merecord teks, python juga dapat merecord data audio secara otomatis di penyimpanan internal komputer.

Execution :



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject - CodeProgramSpeech.py
pythonProject
├── CodeProgramSpeech.py
├── scratch.py
├── g.py
├── Project
│   ├── Heading Complaints Hotel Service_1.wav
│   ├── Heading Complaints Hotel Service_2.wav
│   ├── Heading Complaints Hotel Service_3.wav
│   ├── Heading Complaints Hotel Service_4.wav
│   ├── Heading Complaints Hotel Service_5.wav
│   ├── Heading Complaints Hotel Service_6.wav
│   ├── Making Complaints_1.wav
│   ├── Making Complaints_2.wav
│   ├── Making Complaints_3.wav
│   ├── pyvenv.cfg
│   ├── CodeProgramSpeech.py
│   ├── Making Complaints.wav
│   └── recorded.wav
├── External Libraries
│   ├── Python 3.10 (pythonProject) > C:\Users\Leno...
│   ├── Binary Skeletons
│   └── DLLs
└── Run: CodeProgramSpeech (1)
    ├── Recognizing Now ....
    ├── You have said
    ├── good morning everyone
    └── Audio Recorded Successfully
    └── Process finished with exit code 0
```

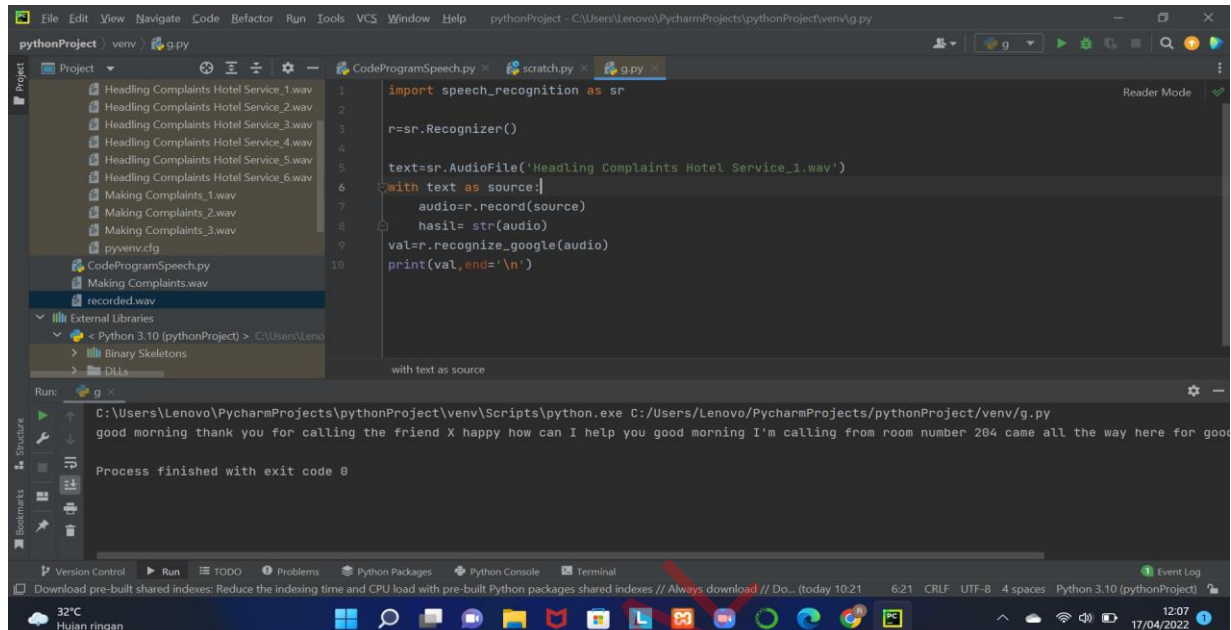


```
File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject - h2.py
pythonProject
├── h2.py
├── Project
│   ├── Heading Complaints Hotel Service_1.wav
│   ├── Heading Complaints Hotel Service_2.wav
│   ├── Heading Complaints Hotel Service_3.wav
│   ├── Heading Complaints Hotel Service_4.wav
│   ├── Heading Complaints Hotel Service_5.wav
│   ├── Heading Complaints Hotel Service_6.wav
│   ├── Making Complaints_1.wav
│   ├── Making Complaints_2.wav
│   ├── Making Complaints_3.wav
│   ├── pyvenv.cfg
│   ├── CodeProgramSpeech.py
│   ├── Making Complaints.wav
│   └── recorded.wav
├── External Libraries
│   ├── Python 3.10 (pythonProject) > C:\Users\Leno...
│   ├── Binary Skeletons
│   └── DLLs
└── Run: h2
    ├── Ucapkan sesuatu
    ├── Recording
    ├── >> good morning everyone
    ├── Done
    └── Recording
    └── Process finished with exit code 0
```

4.4 Hasil Konversi Audio Wav ke Text dengan Speech Recognition Python

Speech Recognition python juga memiliki metode untuk mengkonversikan suara atau audio dalam bentuk Wav menjadi sebuah text. Hasil yang sudah di konversikan akan di cetak dan ditampilkan di terminal run di code python.

Execution :



The screenshot shows the PyCharm IDE with a project named 'pythonProject'. The file explorer on the left shows a directory structure with various audio files (e.g., 'Headling Complaints Hotel Service_1.wav') and Python files. The main editor window displays a Python script named 'g.py' with the following code:

```
1 import speech_recognition as sr
2
3 r=sr.Recognizer()
4
5 text=sr.AudioFile('Headling Complaints Hotel Service_1.wav')
6 with text as source:
7     audio=r.record(source)
8     hasil= str(audio)
9     val=r.recognize_google(audio)
10    print(val,end='\n')
```

The Run window at the bottom shows the command executed: `C:\Users\Lenovo\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/Lenovo/PycharmProjects/pythonProject/venv/g.py`. The output in the terminal is:

```
good morning thank you for calling the friend X happy how can I help you good morning I'm calling from room number 204 came all the way here for good
```

Below the terminal output, there are several lines of text that appear to be a transcription of the audio file, with some parts crossed out by a red 'X' mark:

```
C:\Users\Lenovo\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/Lenovo/PycharmProjects/pythonProject/venv/g.py
but I found out that the water is not even running and the guest next door is very noisy can you do something about this sorry to hear that man I wi
okay thank you for reading we have to get back about the teachers and we would like to offer you and I'm really true in the
well I love that but how about the cost I've already spent a lot didn't worry ma'am
would be free of charge as an competition man
else I can help you with no that's all thank you for your help you're welcome and thank you for a patient and have a nice day
```

Hasil :

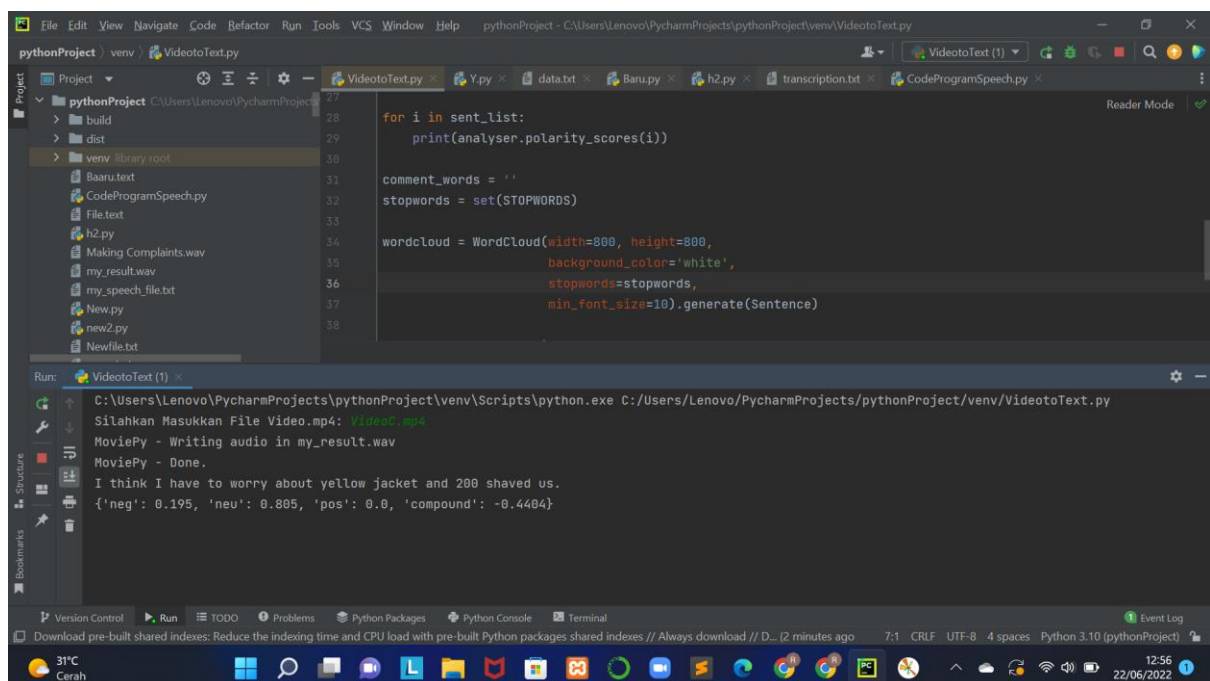
“good morning thank you for calling the friend X happy how can I help you good morning I'm calling from room number 204 came all the way here for good holiday but I found out that the water is not even running and the guest next door is very noisy can you do something about this sorry to hear that man I will talk to my husband right now okay thank you for reading we have to get back about the teachers and we would like to offer you and I'm really true in the well I love that but how about the cost I've already spent a lot didn't worry ma'am would be free of charge as an competition man else I can help you with no that's all thank you for your help you're welcome and thank you for a patient and have a nice day.”

Indonesian :

“Selamat pagi terima kasih telah menelepon teman X senang bagaimana saya bisa membantu Anda selamat pagi saya menelepon dari nomor kamar 204 datang jauh-jauh kesini untuk liburan yang baik tetapi saya menemukan bahwa airnya bahkan tidak mengalir dan tamu di sebelah sangat bising dapatkah kamu melakukan sesuatu tentang ini, maaf mendengar pria itu Saya akan berbicara dengan suami saya sekarang oke terima kasih telah membaca kita harus kembali tentang dan kami ingin menawarkan Anda dan saya benar-benar benar dalam baik saya suka itu tapi bagaimana dengan biaya yang sudah saya keluarkan banyak jangan khawatir Bu akan bebas biaya sebagai orang lain saya dapat membantu Anda dengan tidak itu saja terima kasih atas bantuan Anda sama-sama dan terima kasih untuk pelayannya dan semoga harimu menyenangkan.”

4.5 Hasil Analisis Data dengan Sentimen Analisis

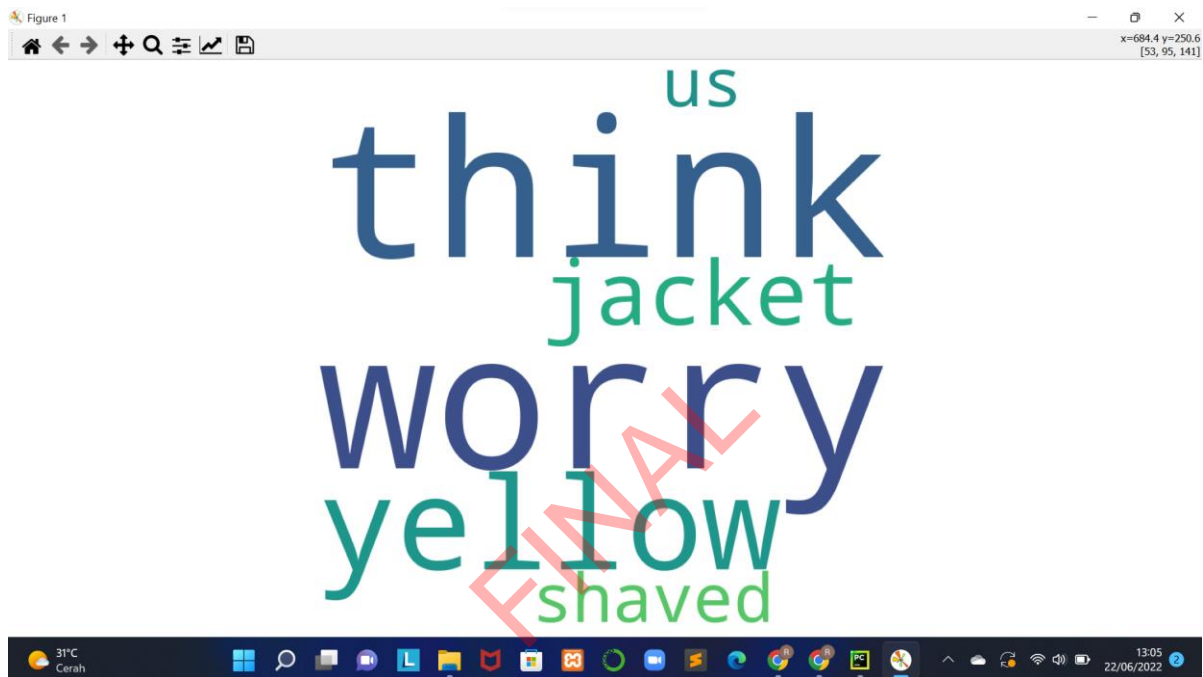
Proses analisis ini dilakukan berdasarkan analisis tiap kata yang dilakukan oleh python. Analisis ini dilakukan dengan mengelompokkan kategori kata atau kalimat (Sentimen analisis) menjadi 3 jenis, yaitu netral, positif, dan negatif. Analisis ini didasarkan pada skor-skor kategori seperti skor untuk negatif, positif, dan netral. Sehingga nanti setelah kalimat dari video di transkrip akan dianalisis dan menghasilkan skor untuk masing-masing kategori, misalnya pada kalimat dibawah ini menghasilkan analisis sentimen dengan kategori netral (dengan skor 0,195), netral (dengan skor 0,805), positif (dengan skor 0). Dari hasil analisis tersebut dapat disimpulkan bahwa pada kalimat ini cenderung netral, dan tidak ada unsur positifnya.



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject - C:\Users\Lenovo\PycharmProjects\pythonProject\venv\VideoToText.py
pythonProject venv VideoToText.py
Project
  pythonProject C:\Users\Lenovo\PycharmProjects\pythonProject\venv\VideoToText.py
    build
    dist
    venv library root
    Baaru.txt
    CodeProgramSpeech.py
    File.txt
    h2.py
    Making Complaints.wav
    my_result.wav
    my_speech_file.txt
    New.py
    new2.txt
    Newfile.txt
  Run: VideoToText (1)
    C:\Users\Lenovo\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Users/Lenovo/PycharmProjects/pythonProject/venv/VideoToText.py
    Silahkan Masukkan File Video.mp4: VideoC.mp4
    MoviePy - Writing audio in my_result.wav
    MoviePy - Done.
    I think I have to worry about yellow jacket and 200 shaved us.
    {'neg': 0.195, 'neu': 0.805, 'pos': 0.0, 'compound': -0.4404}
  Run
  Version Control
  Run
  TODO
  Problems
  Python Packages
  Python Console
  Terminal
  Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // D... (2 minutes ago)
  7:1 CRLF UTF-8 4 spaces Python 3.10 (pythonProject)
  31°C
  Cera
  12:56
  22/06/2022
```


4.6 Hasil Word Cloud

Setelah dianalisis dengan sentimen analisis, pada project ini kami juga membuat word cloud untuk mengetahui kata-kata yang sering muncul dalam kalimat. Hal ini dilakukan agar dalam menganalisis suatu kalimat, kita dapat mengetahui kalimat ini cenderung ke kategori apa, positif, negatif, atau netral. Dengan adanya word cloud dapat membantu dalam proses analisis yang dilakukan oleh python dan memudahkan pengguna mengetahui atau menganalisa.



4.7 Hasil UI Transcribe Video

Hasil akhir dari project ini adalah transcribe video menjadi kalimat-kalimat. Pada sistem ini kita dapat mengupload file video yang ingin kita transcribe. Dengan mengklik file kemudian memilih file video yang akan di transcribe . Setelah ini masukkan nama file (misalnya interview1.txt) saat kita ingin menstranscribe lagi dengan nama file yang sama maka secara otomatis teks akan dilanjutkan, dan akan berganti ketika nama filenya diubah.

5. BAB V: KESIMPULAN

Berdasarkan hasil dari Project Based Learning Pemrograman Fungsional yang telah dilakukan, dapat disimpulkan bahwa video Youtube keluhan pelanggan dapat dianalisis dengan teknologi Speech Recognition. Video Youtube akan diubah terlebih dahulu ke bentuk WAV. Dengan demikian, teknologi Speech Recognition dapat mengenali pola suara pelanggan mengenai keluhannya dengan baik dan mentranskrip suara keluhan tersebut secara otomatis. Dari hasil transkrip tersebut, dapat dianalisis tingkat sentiment dari setiap kata dalam satu kalimat atau paragraf dengan menggunakan function vaderSentiment yang terdapat pada python. Jika hasil negatif lebih besar dari positif, maka keluhan tersebut negatif. Selain itu, dari hasil transkrip tersebut akan divisualisasikan dengan wordcloud, sehingga kata yang paling banyak muncul akan divisualisasikan semakin besar. Dengan demikian, perusahaan dapat lebih efektif meningkatkan kepuasan pelanggan sesuai dengan keluhan yang paling banyak terjadi.

FINAL

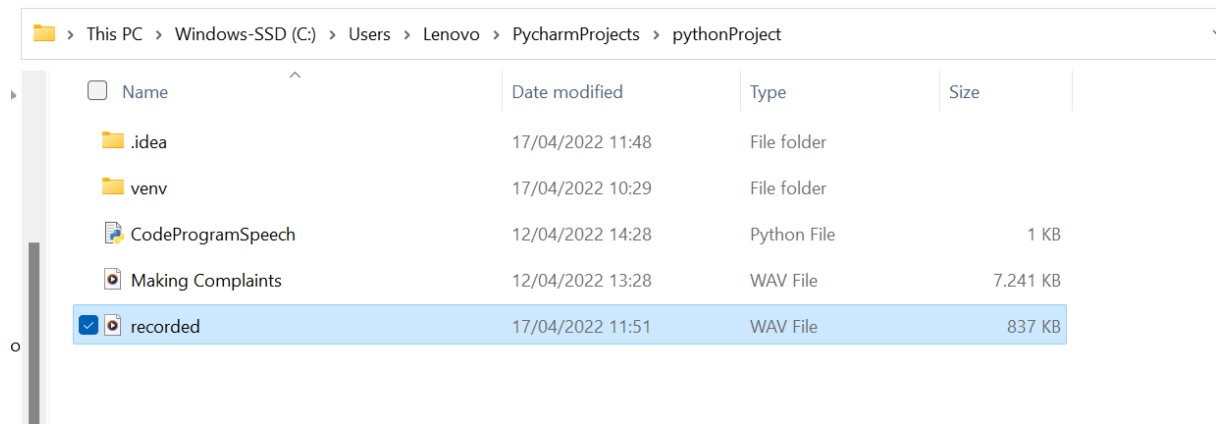
6. DAFTAR PUSTAKA






- Isa, Sani M. (2019, Mei 08). Speech Recognition. Diakses dari <https://mti.binus.ac.id/2019/05/08/speech-recognition/>
- Vocalmatic. 2020. Transkripsi Otomatis Suara dan Video Anda ke Teks. Diakses dari <https://vocalmatic.com/id>
- Widya Wicara. 2021. Bagaimana Teknologi Speech to Text Bekerja dan Apa Kegunaannya?. Diakses dari <https://widyawicara.com/bagaimana-teknologi-speech-to-text-bekerja-dan-apa-kegunaannya/>
- Masridah, Lelis. 2019. Speech Recognition Untuk Mengendalikan Pagar Menggunakan Arduino Nano. Undergraduate thesis. Universitas Siliwangi. Diakses dari <http://repositori.unsil.ac.id/218/>
- Laksono, Teguh Puji. 2018. Speech to Text Untuk Bahasa Indonesia. Undergraduate thesis. Universitas Islam Indonesia. Diakses dari <https://dspace.uui.ac.id/handle/123456789/10756>
- Kurniawan, Angga. 2017. Aplikasi Speech Recognition Sebagai Pengenalan Ucapan Tunawicara Menggunakan Google Cloud Speech Api Berbasis Android. Undergraduate thesis. Universitas Islam Negeri Syarif Hidayatullah Jakarta. Diakses dari <https://repository.uinjkt.ac.id/dspace/handle/123456789/52445>
- Rinaldy, Irfan. 2017. Aplikasi Speech Recognition dan Text to Speech Berbasis Android. Undergraduate thesis. Universitas Islam Negeri Alauddin Makassar. Diakses dari <https://repositori.uin-alauddin.ac.id/10413/>

7. LAMPIRAN

Pada bagian lampiran dapat ditambahkan beberapa set data dan tambahan lampiran penunjang yang membantu dalam proses pengerjaan proyek.

Data audio dari deteksi suara yang tersimpan secara otomatis di perangkat (WAV)



This PC > Windows-SSD (C:) > Users > Lenovo > PycharmProjects > pythonProject				
<input type="checkbox"/> Name	Date modified	Type	Size	
 .idea	17/04/2022 11:48	File folder		
 venv	17/04/2022 10:29	File folder		
 CodeProgramSpeech	12/04/2022 14:28	Python File	1 KB	
 Making Complaints	12/04/2022 13:28	WAV File	7.241 KB	
<input checked="" type="checkbox"/>  recorded	17/04/2022 11:51	WAV File	837 KB	

FINAL