

Implementasi Voice Recognition Berbasis Machine Learning

Fahrizal Adnan¹, Ilya Amelia², dan Sayyid ‘Umar Shiddiq³

¹ Program Studi Teknik Elektro, Fakultas Teknik, Universitas Negeri Semarang
Jl. Taman Siswa, Kampus Pusat : Sekaran, 50229 Gunung Pati, Semarang, Indonesia

^{2,3} Program Studi Teknik Elektro, Fakultas Teknik, Universitas Negeri Semarang
Jl. Taman Siswa, Kampus Pusat : Sekaran, 50229 Gunung Pati, Semarang, Indonesia

fahrizaladnan1@students.unnes.ac.id¹, ilyame9@students.unnes.ac.id², sayyidumarshiddiq@students.unnes.ac.id³

Abstrak— Membuat suatu dokumen yang bersumber dari rekaman suara saat ini bukanlah suatu hal yang merepotkan. Hal tersebut dapat diatasi dengan mudah dan cepat menggunakan suatu program yang disebut Voice Recognition. Program tersebut merupakan perintah yang digunakan untuk membuat dokumen tertulis dengan cara mengubah suara menjadi teks. Percobaan dilakukan dengan menggunakan library Speech recognition serta melakukan pengujian suara secara realtime dan pengujian pada file audio yang sudah tersedia. Baik suara secara realtime maupun file audio mengucapkan kalimat yang sama. Tujuan dari percobaan ini adalah untuk mengukur tingkat akurasi program speech to text menggunakan library Speech recognition diantara suara secara realtime dengan file audio. Tingkat akurasi kemudian diproses menggunakan metode cosine similarity. Hasil menunjukkan bahwa penangkapan audio secara realtime menghasilkan tingkat akurasi sebesar 0.97 dan 0.93 dan pemrosesan audio rekaman memiliki tingkat akurasi sebesar 0.97.

Kata kunci— Voice Recognition, machine learning, cosine similarity.

Abstract— Creating a document that is sourced from sound recordings is not a hassle nowadays. This can be solved easily and quickly using a program called Voice Recognition. The program is a command that is used to create a written document by converting sound into text. The experiment was carried out using the Speech recognition library as well as conducting real-time voice testing and testing on audio files that were already available. Both realtime voice and audio files say the same sentence. The purpose of this experiment is to measure the accuracy of speech to text programs using the Speech recognition library between voices in real time and audio files. The level of accuracy is then processed using the cosine similarity method. The results show that real-time audio capture produces an accuracy rate of 0.97 and 0.93 and audio recording processing has an accuracy rate of 0.97.

Keywords— Pengenalan Suara, pembelajaran mesin, kesamaan kosinus.

I. PENDAHULUAN

Suara merupakan bunyi yang dikeluarkan dari mulut manusia seperti bercakap – cakap, tertawa, menangis dan berteriak yang digunakan manusia sebagai sarana komunikasi utama untuk bertukar informasi antar sesamanya. Namun seiring dengan berkembangnya zaman, komunikasi tidak hanya terjalin antara manusia dengan manusia yang lain. Komunikasi kini sudah bisa dilakukan antara manusia dengan komputer. Komputer terlebih dahulu diberi petunjuk untuk dapat membaca suara yang masuk atau biasa disebut dengan supervised learning.

Di dalam ilmu komputer terdapat bidang ilmu yang disebut dengan Speech to Text. Salah satu keunggulan utama untuk layanan ini adalah pengurangan kesalahan ejaan pada kata – kata yang mungkin terjadi oleh seseorang saat mengetik. Speech to text dapat diterapkan diberbagai bidang seperti membuat alat bantu komunikasi bagi translator, penyandang disabilitas hingga home automation. Beberapa penelitian mengenai speech to text ini telah dilakukan dalam berbagai bahasa seperti bahasa jawa dan bahasa sunda (Fathurrahman, Dede N et al, 2018).

A. Speech Recognition

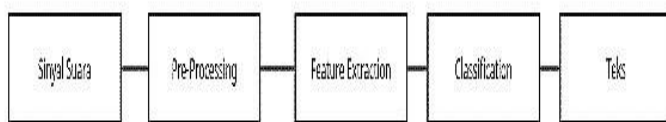
Speech recognition merupakan kemampuan mesin untuk mendengarkan kata – kata yang diucapkan dan mengidentifikasinya. Kemampuan untuk dapat mengubah suara yang masuk pada komputer ke dalam bentuk teks. Speech Recognition menggabungkan ilmu komputer dan linguistik untuk mengidentifikasi kata – kata yang diucapkan dan mengubahnya menjadi teks. Hal ini memungkinkan komputer untuk memahami bahasa manusia. Teknologi speech recognition 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 pada suatu perangkat.

Platform python dapat digunakan untuk mengubah kata – kata yang diucapkan menjadi teks, membuat kueri, atau memberikan balasan. Kata – kata yang terucap lalu diubah bentuknya ke dalam sinyal digital dengan cara mengubah gelombang suara menjadi sekumpulan angka yang kemudian disesuaikan dengan kode – kode tertentu untuk mengidentifikasikan kata-kata tersebut. Hasil dari identifikasi kata yang diucapkan ditampilkan dalam bentuk tulisan atau

dapat dibaca oleh perangkat teknologi sebagai sebuah perintah untuk melakukan suatu pekerjaan.

Proses dalam dunia nyata secara umum menghasilkan observable output yang dapat dikarakterisasikan sebagai sinyal (Rabiner, Lawrence, 1989). Sinyal dapat bersifat diskrit (karakter dalam alfabet) maupun kontinu (pengukuran temperatur, alunan musik). Sinyal bisa bersifat stabil (nilai statistiknya tidak berubah terhadap waktu) maupun tidak stabil (nilai sinyal berubah-ubah terhadap waktu). Dengan melakukan pemodelan terhadap sinyal secara benar, dapat dilakukan simulasi terhadap sumber dan pelatihan sebanyak mungkin melalui proses simulasi tersebut.

Singkatnya, terdapat 3 proses umum bagaimana komputer mengubah sinyal suara ke dalam bentuk teks, yaitu preprocessing, feature extraction dan pengklasifikasi.



Gambar 1. Proses umum Speech recognition

Pada gambar diatas, masukan sinyal suara akan di preprocessing terlebih dulu untuk mempersiapkan dan mengolah data awal sehingga data yang digunakan merupakan data yang sudah siap pakai dan dapat mempermudah proses-proses dalam tahapan berikutnya, lalu data yang sudah di preprocessing ini di ekstrak ciri nya dan dipisahkan, setelah itu, sistem akan membandingkan hasil ekstraksi yang sudah dilakukan dengan database yang tersedia.

Parameter yang dibandingkan ialah tingkat penekanan suara yang kemudian akan dicocokkan dengan template database yang tersedia. Parameter yang dibandingkan ialah tingkat penekanan suara yang kemudian akan dicocokkan dengan template database yang tersedia. Pada zaman sekarang, Speech Recognition sudah banyak digunakan, namun sayangnya aplikasi yang menggunakan teknologi Speech Recognition lebih banyak dirancang untuk ter built-in pada mobile phone tertentu, contoh : (Siri pada IOS), dan pada umumnya, teknologi Automatic Speech Recognition digunakan untuk salah satu alat bantu menjalankan perintah – perintah singkat, seperti mengetik teks, pencarian kontak, dan lain sebagainya.

Ada beberapa faktor yang mempengaruhi kurangnya akurasi hasil dari system ini diantaranya adalah rendahnya rasio sinyal terhadap noise, ucapan yang overlapping, penggunaan daya komputer, juga kata-kata homonim. Pada kasus rendahnya rasio sinyal terhadap noise, ketika pengguna mengucapkan kata-

kata yang dimaksud, ada kemungkinan adanya noise atau suara

tambahan yang mengganggu program untuk menerima kata-kata yang dimaksud. Maka dari itu, pengguna harus berada di ruangan yang tenang dan memposisikan mikrofon sedekat mungkin dengan mulut agar mengurangi noise yang diterima sistem. Sedangkan pada kasus ucapan yang overlapping, sistem pengenalan ucapan saat ini masih kesulitan untuk

mengidentifikasi kata-kata jika ada ucapan lain yang masuk dalam sistemnya, misalnya ketika pengguna menggunakan sistem ini ketika ada percakapan lain di dekatnya sehingga kata - kata tersebut juga masuk ke dalam sistem dan ikut diproses, akurasi dari hasil yang didapat akan sangat buruk karena hal ini.

B. Machine Learning

Machine Learning (ML) atau pembelajaran mesin merupakan pendekatan dalam AI yang banyak digunakan untuk menggantikan atau menirukan perilaku manusia untuk menyelesaikan masalah atau melakukan otomatisasi. Sesuai namanya, ML mencoba menirukan bagaimana proses manusia atau makhluk cerdas belajar dan mengeneralisasi (Tanaka, M et al, 2014). Setidaknya ada dua aplikasi utama dalam ML yaitu, klasifikasi dan prediksi . Ciri khas dari ML adalah adanya proses pelatihan, pembelajaran, atau training. Oleh karena itu, ML membutuhkan data untuk dipelajari yang disebut sebagai data training. Klasifikasi adalah metode dalam ML yang digunakan oleh mesin untuk memilah atau mengklasifikasikan obyek berdasarkan ciri tertentu sebagaimana manusia mencoba membedakan benda satu dengan yang lain. Sedangkan prediksi atau regresi digunakan oleh mesin untuk menerka keluaran dari suatu data masukan berdasarkan data yang sudah dipelajari dalam training. Metode ML yang paling populer yaitu Sistem Pengambil Keputusan, Support Vector Machine (SVM) dan Neural Network. Speech recognition termasuk dalam salah satu penerapan machine learning.

C. Cosine Similarity

Cosine Similarity secara sederhana didefinisikan sebagai alat untuk membandingkan suatu kemiripan diantara dua dokumen. Cosine similarity merupakan metrik yang digunakan untuk mengukur kesamaan dua vektor. Secara khusus, cosine similarity mengukur kesamaan dalam arah atau orientasi vektor dengan mengabaikan perbedaan dalam besaran atau skalanya. Metode ini memiliki konsep normalisasi panjang vektor data dengan membandingkan *N-gram* yang sejajar satu sama lain dari 2 pembandingan (Nurdiana, O et al, 2016).

Cosine similarity digunakan dalam ruang positif, dimana hasilnya dibatasi antara 0 hingga 1 (Ariantini, D.A.R., Lumenta, A.S. and Jacobus, A., 2016). Jika hasilnya 0 maka dokumen tersebut dikatakan tidak mirip dan sebaliknya ketika hasil cosine similarity adalah 1 maka dua dokumen tersebut dikatakan mirip. Dalam Information Retrieval, masing-masing kata/istilah (term) diasumsikan sebagai dimensi yang berbeda dan dokumen ditandai dengan vector dimana nilai masing-masing dimensi sesuai dengan berapa istilah muncul dalam dokumen. Berikut adalah rumus cosine similarity :

$$\text{Similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

Keterangan :

A = vector

B = vector

A_i = bobot *term I* dalam blok A_i
 B_i = bobot *term I* dalam blok B_i
 I = jumlah *term* dalam kalimat

II. METODE PENELITIAN

Percobaan dibagi menjadi dua bagian yaitu pengambilan data dan pengolahan data. Pengambilan data didapat dengan merekam audio secara realtime dan mengambil sampel file audio. Pengambilan suara secara realtime dibantu oleh microphone pada komputer. Kemudian input berupa suara ini akan diolah oleh komputer dengan mengubah energi listrik dari analog ke digital untuk diidentifikasi. Setelah input diproses barulah dihasilkan output berupa teks. Pengolahan data dilakukan oleh program yang dibuat. Modul dan aplikasi yang dibutuhkan adalah Speech Recognition, pyaudio dan jupyter Notebook.

```
pip install speechrecognition
***

pip install pyaudio
***

import speech_recognition as sr

mic = sr.Microphone()
engine = sr.Recognizer()

#Speech to Text
while True :
    engine.pause_threshold = 2
    with mic as source:
        print("katakan sesuatu")
        #menangkap audio
        rekam = engine.listen(source)
        print("Recognizing Now...")

    try:
        teks = engine.recognize_google(rekam, language = "id-ID")
        print(teks)
        if teks == 'berhenti' :
            break
    except engine.UnknownValueError :
        print("Maaf, suara tidak terdengar. Silahkan coba lagi")
```

Gambar 2. Pemrograman Speech Recognition

Percobaan dilakukan menggunakan platform Jupyter Notebook dengan bahasa python. Selanjutnya adalah membuat pemrograman agar komputer dapat memahami masukan. Adapun tahapan – tahapan yang dilakukan adalah sebagai berikut :

1) Installing module

Mengunduh dua module yaitu modul `speech_recognition` dan `pyaudio`. Oleh karena module `speech_recognition` tidak dapat digunakan sendiri, maka digunakan bersamaan dengan `pyaudio` sebagai penangkap suara lewat microphone.

```
[1]: pip install speechrecognition
***

[2]: pip install pyaudio
***
```

Gambar 3. Installing Package

2) Menginisialisasi perintah awal

Perintah yang diperlukan speech recognition adalah `Recognizer()` dan perintah untuk menerima input yaitu `Microphone()`. Perintah `Recognizer()` digunakan dengan perintah lain untuk menangkap suara seperti `listen()`. Sedangkan perintah untuk menangkap suara digunakan `sr.Microphone()`, yang didefinisikan sebagai source untuk memudahkan dalam penangkapan suara. Untuk speech recognition dengan menggunakan data berupa sampel file audio digunakan `sr.AudioFile()` yang memberi instruksi kepada mesin untuk membaca sebuah file audio yang terdapat pada perangkat. Kemudian fungsi print digunakan untuk memberikan keterangan atau aba – aba kepada pembicara untuk mengucapkan kalimat. Sedangkan `engine.listen(source)` berfungsi untuk menangkap audio suara yang diucapkan. Untuk menulis perintah dapat didefinisikan dengan variabel sembarang seperti gambar di bawah ini :

```
mic = sr.Microphone()
engine = sr.Recognizer()

#Speech to Text
while True :
    engine.pause_threshold = 2
    with mic as source:
        print("katakan sesuatu")
        #menangkap audio
        rekam = engine.listen(source)
        print("Recognizing Now...")
```

Gambar 4. Inisialisasi First Command

3) Menggunakan Fungsi Try and Except

Dilakukan untuk mendeteksi apakah program yang dibuat berhasil menerjemahkan ucapan ke dalam teks atau memiliki hasil error. Fungsi try diisi oleh perintah `recognize_google()` sebagai metode pengenalan suara. Sedangkan fungsi except, diisi perintah `sr.UnknownValueError` yang akan mendeteksi jawaban yang tidak dapat dipahami komputer maka akan muncul print yang memberitahu bahwa komputer tidak dapat memahami apa yang diucapkan pembicara.

```
try:
    teks = engine.recognize_google(rekaman, language = "id-ID")
    print(teks)
    if teks == 'berhenti' :
        break
except engine.UnknownValueError :
    print("Maaf, suara tidak terdengar. Silahkan coba lagi")
except Exception as e:
    print(e)
```

Gambar 5. Fungsi try dan except

4) Menambahkan Fungsi Threshold Dan If-Break Terdapat fungsi `engine.pause_threshold` dan fungsi `if-break`

yang bersifat opsional. `engine.pause_threshold` berperan untuk memberi jeda waktu saat pembicara berhenti berbicara dalam jangka waktu yang ditentukan . Dalam percobaan ini, nilai `pause_threshold` adalah 2 detik. Sedangkan fungsi `if-break` berfungsi untuk memberhentikan program simulasi menggunakan kata kunci yang sudah ditentukan. Hal ini dilakukan untuk memberhentikan program simulasi yang terus berjalan selama suara dapat dikenali oleh komputer.

```
#Speech to Text
while True :
    engine.pause_threshold = 2
    with mic as source:
        print("katakan sesuatu")
        #menangkap audio
        rekam = engine.listen(source)
        print("Recognizing Now...")

    try:
        teks = engine.recognize_google(rekam, language = "id-ID")
        print(teks)
        if teks == 'berhenti' :
            break
    except engine.UnknownValueError :
        print("Maaf, suara tidak terdengar. Silahkan coba lagi")
```

Gambar 6. Posisi fungsi engine.pause_threshold dan if-break

5) Menambahkan Perintah Cosine Similarity

Untuk mengukur tingkat akurasi baik pada modul speech recognition berbasis audio realtime maupun sample file audio, digunakan metode cosine similarity dengan pemrograman sebagai berikut :

```
cosine similarity

s1 = "pembuangan sampah yang tidak diurus dengan baik akan mengakibatkan masalah besar karena penumpukan sampah atau membuang
s2 = teks

import math
import pandas as pd

def fungsi cosine similarity
def cosine_sim(NOW1,NOW2):
    vec1 = list(NOW1)
    vec2 = list(NOW2)
    dot_prod = 0
    for i, v in enumerate(vec1):
        dot_prod += v*vec2[i]
    mag_1 = math.sqrt(sum([x**2 for x in vec1]))
    mag_2 = math.sqrt(sum([x**2 for x in vec2]))
    return dot_prod/(mag_1*mag_2)

cosinebow = []
bow1 = s1.split()
bow2 = s2.split()

uniquewords = (set(bow1)|set(bow2))
NOW1 = dict.fromkeys(uniquewords, 0)
for word in bow1:
    NOW1[word] += 1

NOW2 = dict.fromkeys(uniquewords, 0)
for word in bow2:
    NOW2[word] += 1

H = pd.DataFrame([NOW1,NOW2])

#perhitungan cosine similarity
similarity = cosine_sim(NOW1.values(),NOW2.values())
print("cosine similarity", similarity)
```

Gambar 7. Pemrograman Cosine Similarity

III. HASIL DAN PEMBAHASAN

Penelitian dilakukan dengan mengambil dua percobaan yaitu menggunakan audio realtime dan menggunakan file audio (rekaman suara). Percobaan menggunakan audio realtime dilakukan dengan dua cara, yaitu cara pertama dengan mengucapkan kalimat dibawah secara cepat dan yang kedua adalah mengucapkannya secara normal. Sedangkan percobaan menggunakan audio rekaman dilakukan dengan menggunakan suara dua pelaku percobaan yang berbeda gender yang membaca kalimat dengan normal. Berikut merupakan kalimat yang digunakan untuk mengukur tingkat akurasi modul speech recognition :

“pembuangan sampah yang tidak diurus dengan baik akan mengakibatkan masalah besar karena penumpukan sampah atau membuangnya sembarangan ke kawasan terbuka akan mengakibatkan pencemaran tanah yang juga akan berdampak ke saluran air tanah”

Untuk menghitung cosine similarity dari percobaan, perlu diketahui vektor A dan vektor B yang mewakili term kata asal

dan term kata hasil. Hal ini bertujuan untuk melihat setiap unique words dalam dokumen

A. Audio Realtime

1) Membaca Normal

Hasil keluaran yang tertangkap program dan hasil mengukur kesamaan menggunakan metode cosine similarity ditampilkan pada gambar 8 dan gambar 9 dibawah ini.

katakan sesuatu
Recognizing Now...
pembuangan sampah yang tidak diurus dengan baik akan mengakibatkan masalah besar karena penumpukan sampah atau membuangnya sembarangan ke kawasan terbuka akan mengakibatkan pencemaran tanah yang juga akan berdampak ke saluran air tanah

Gambar 8. Output yang tertangkap

cosine similarity 0.9789450103725609

Gambar 9. Hasil perhitungan Cosine Similarity

Kata Asal (A)	Kata Hasil (B)	Kategori
pembuangan	pembuangan	Tepat
sampah	sampah	Tepat
yang	yang	Tepat
tidak	tidak	Tepat
diurus	diurus	Tepat
dengan	dengan	Tepat
baik	baik	Tepat
akan	akan	Tepat
mengakibatkan	mengakibatkan	Tepat
masalah	masalah	Tepat
besar	besar	Tepat
karena	karena	Tepat
penumpukan	penumpukan	Tepat
sampah	sampah	Tepat
atau	atau	Tepat
membuangnya	membuangnya	Tepat
sembarangan	sembarangan	Tepat
ke	Ke	Kurang Tepat
kawasan	kawasan	Tepat
terbuka	terbuka	Tepat
akan	akan	Tepat
mengakibatkan	mengakibatkan	Tepat
pencemaran	pencemaran	Tepat
tanah	tanah	Tepat
yang	yang	Tepat
juga	juga	Tepat
akan	akan	Tepat
berdampak	berdampak	Tepat
ke	ke	Tepat
saluran	saluran	Tepat
air	air	Tepat
tanah	tanah	Tepat

2) Membaca Cepat

Hasil keluaran yang tertangkap program dan hasil mengukur kesamaan menggunakan metode cosine similarity ditampilkan pada gambar 10 dan gambar 11 dibawah ini

Katakan sesuatu
Recognizing How...
pembuangan sampah yang tidak diurus dengan baik akan mengakibatkan masalah besar karena penumpukan sampah sembarangan kawasan terbuka akan mengakibatkan pencemaran tanah yang juga akan berdampak saluran air tanah

Gambar 10. Output yang tertangkap

cosine similarity 0.9354143466934853

Gambar 11. Hasil perhitungan Cosine Similarity

Kata Asal (A)	Kata Hasil (B)	Kategori
pembuangan	pembuangan	Tepat
Sampah	sampah	Tepat
Yang	yang	Tepat
Tidak	tidak	Tepat
Diurus	diurus	Tepat
Dengan	dengan	Tepat
Baik	baik	Tepat
Akan	akan	Tepat
mengakibatkan	mengakibatkan	Tepat
Masalah	Masalah	Tepat
Besar	Besar	Tepat
Karena	Karena	Tepat
penumpukan	Penumpukan	Tepat
Sampah	Sampah	Tepat
Atau	-	Tidak Tepat
membuangnya	-	Tidak Tepat
sembarangan	Sembarangan	Tepat
Ke	-	Tidak Tepat
Kawasan	Kawasan	Tepat
Terbuka	Terbuka	Tepat
Akan	Akan	Tepat
mengakibatkan	Mengakibatkan	Tepat
Pencemaran	Pencemaran	Tepat
Tanah	Tanah	Tepat
Yang	Yang	Tepat
Juga	Juga	Tepat
Akan	Akan	Tepat
Berdampak	Berdampak	Tepat
Ke	-	Tidak Tepat
Saluran	Saluran	Tepat
Air	Air	Tepat
Tanah	Tanah	Tepat

B. Audio Rekaman

1) Audio rekaman dengan suara wanita sebagai pengisi audio

Mendengarkan audio
Hasil: pembuangan sampah yang tidak diurus dengan baik akan mengakibatkan masalah besar karena penumpukan sampah atau pembuangan sembarangan ke kawasan terbuka akan mengakibatkan pencemaran tanah yang juga akan berdampak ke saluran air tanah

Gambar 12. Output yang tertangkap

cosine similarity 0.9789450103725609

Gambar 13. Hasil perhitungan Cosine Similarity

Kata Asal (A)	Kata Hasil (B)	Kategori
Pembuangan	pembuangan	Tepat
Sampah	sampah	Tepat
Yang	yang	Tepat
Tidak	tidak	Tepat
Diurus	diurus	Tepat
Dengan	dengan	Tepat
Baik	baik	Tepat
Akan	akan	Tepat
mengakibatkan	mengakibatkan	Tepat
Masalah	masalah	Tepat
Besar	besar	Tepat
Karena	karena	Tepat
penumpukan	penumpukan	Tepat
Sampah	sampah	Tepat
Atau	atau	Tepat
membuangnya	membuangnya	Tepat
sembarangan	sembarangan	Tepat
ke	Ke	Kurang Tepat
kawasan	kawasan	Tepat
terbuka	terbuka	Tepat
akan	akan	Tepat
mengakibatkan	mengakibatkan	Tepat
pencemaran	pencemaran	Tepat
tanah	tanah	Tepat
yang	yang	Tepat
juga	juga	Tepat
akan	akan	Tepat
berdampak	berdampak	Tepat
ke	ke	Tepat
saluran	saluran	Tepat
air	air	Tepat
tanah	tanah	Tepat

2) Audio Rekaman dengan suara pria sebagai pengisi audio

Pengisi audio
Hasil: pembuangan sampah yang tidak diurus dengan baik akan mengakibatkan masalah besar karena penumpukan sampah atau membuang ya sembarangan ke kawasan terbuka akan mengakibatkan pencemaran tanah yang juga akan berdampak ke saluran air tanah

Gambar 14. Output yang tertangkap
cosine similarity 0.9789450103725609

Gambar 15. Hasil perhitungan Cosine Similarity

Kata Asal (A)	Kata Hasil (B)	Kategori
pembuangan	pembuangan	Tepat
sampah	sampah	Tepat
yang	yang	Tepat
tidak	tidak	Tepat
diurus	diurus	Tepat
dengan	dengan	Tepat
baik	baik	Tepat
akan	akan	Tepat
mengakibatkan	mengakibatkan	Tepat
masalah	masalah	Tepat
besar	besar	Tepat
karena	karena	Tepat
penumpukan	penumpukan	Tepat
sampah	sampah	Tepat
atau	atau	Tepat
membuangnya	membuangnya	Tepat
sembarangan	sembarangan	Tepat
ke	Ke	Kurang Tepat
kawasan	kawasan	Tepat
terbuka	terbuka	Tepat
akan	akan	Tepat
mengakibatkan	mengakibatkan	Tepat
pencemaran	pencemaran	Tepat
tanah	tanah	Tepat
yang	yang	Tepat
juga	juga	Tepat
akan	akan	Tepat
berdampak	berdampak	Tepat
ke	ke	Tepat
saluran	saluran	Tepat
air	air	Tepat
tanah	tanah	Tepat

IV. KESIMPULAN

Dengan menggunakan modul *speechrecognition*, pengenalan suara dapat dilakukan. modul ini memiliki tingkat

akurasi yang sangat baik. Namun disatu sisi, modul memiliki kelemahan yaitu memerlukan koneksi internet yang baik agar output dapat segera terlihat. Apabila perangkat penguji sedang dalam koneksi internet yang buruk, maka akan memakan waktu yang relatif lama untuk mengeluarkan hasilnya. Apabila tidak ada koneksi sama sekali, maka pengenalan suara tidak dapat dilakukan.

REFERENSI

- Ariantini, D.A.R., Lumenta, A.S. and Jacobus, A., 2016. Pengukuran Kemiripan Dokumen Teks Bahasa Indonesia Menggunakan Metode Cosine Similarity. *Jurnal Teknik Informatika*, 9(1).
- Bengio, Yoshua; LeCun, Yann; Hinton, Geoffrey (2015). "Deep Learning". *Nature*. 521 (7553): 436–444.
- Ciresan, D.; Meier, U.; Schmidhuber, J. (2012). "Multi-column deep neural networks for image classification". *2012 IEEE Conference on Computer Vision and Pattern Recognition*. pp. 3642–3649.
- Fathurrahman, Dede N et al. (2018). Deep Neural network Untuk Pengenalan Ucapan Pada Bahasa Sunda Dialek Tengah Timur (Majalengka). *e-Proceeding of Engineering*. 5(3). 6073-6080.
- Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep Learning* (Vol. 1). MIT Press Cambridge.
- Isa, Sani M. (2019). *Speech Recognition*. [Online]. URL : <https://mti.binus.ac.id/2019/05/08/speechrecognition/#:~:text=Speech%20Recognition%20atau%20yang%20biasa%20dikenal%20dengan%20automatic,yang%20diucapkan%20dengan%20cara%20digitalisasi%20kata%20dan%20> . Diakses tanggal 13 April 2022.
- Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey (2012). *"ImageNet Classification with Deep Convolutional Neural Networks"*
- Laksono, Teguh Puji. (2018). *Speech To Text Untuk Bahasa Indonesia*. Skripsi FTI UII, UII Yogyakarta: diterbitkan.
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). *Deep Learning*. *Nature*, 521(7553), 436
- Nurdiana, O., Jumadi., dan Nursantika, D. 2016. Perbandingan Metode Cosine Similarity dengan Metode Jaccard Similarity pada Aplikasi Pencarian Terjemahan AlQur'an dalam Bahasa Indonesia. *Jurnal Online Informatika Volume 1, Nomor 1*.
- Pratama, R.P., Faisal, M. and Hanani, A., 2019. Deteksi Plagiarisme pada Dokumen Jurnal Menggunakan Metode Cosine Similarity. *SMARTICS Journal*, 5(1), pp.22-26.
- Rabiner, Lawrence, R. (1989). *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. *Proceedings of the IEEE*, 77(2), 257–286.
- Tanaka, M.; Okutomi, M. A novel inference of a restricted boltzmann machine. *Pattern Recognition (ICPR)*, 2014 22nd International Conference on. 2014; pp 1526–1531.