Penerapan Algoritma Boyer-Moore Dalam *Speech Recognition*

Shelma Salsabila - 13521115
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail) : 13521115@std.stei.itb.ac.id

***Abstract*—*Speech Recognition* merupakan teknologi yang memungkinkan komputer untuk mengenali dengan mengubah ucapan manusia menjadi teks tertulis. *Speech Recognition* ini di dalam dunia AI digunakan untuk banyak hal. Salah satunya adalah dalam melakukan pencarian jawaban dalam sebuah mesin pencarian seperti Google. Bahkan aplikasi *E-commerce* seperti Shopee juga sudah menerapkan fitur Speech Recognition ini. Serta beberapa penerapan lain. Dalam makalah ini spesifiknya akan membahas mengenai penerapan Speech Recognition di dalam mesin pencarian dengan algoritma *string matching* hasil input suara dengan jawaban yang ada menggunakan algoritma Boyer-Moore.**

Keywords—string matching; boyer-moore; speech recognition; mesin pencarian

I. PENDAHULUAN

Akhir-akhir ini perkembangan teknologi semakin pesat. Tidak bisa dipungkiri perkembangan ini mampu membantu kehidupan manusia menjadi lebih mudah. Salah satu teknologi yang cukup membantu itu adalah *speech recognition*. *Speech Recognition* adalah teknologi yang memungkinkan komputer untuk mengenali dan mengubah ucapan manusia menjadi teks tertulis. Ini melibatkan pemrosesan sinyal suara dan analisis linguistik untuk menginterpretasikan dan memahami apa yang diucapkan oleh pengguna. Salah satu implementasi dari *speech recognition* ini adalah dalam sebuah mesin pencarian. Ada beberapa tahap dalam proses *speech recognition* di dalam mesin pencarian yaitu, pertama sinyal suara ditangkap oleh mikrofon ditangkap dan diubah menjadi bentuk digital oleh suatu perangkat keras komputer. Kemudian, sinyal suara tersebut dianalisis untuk mengidentifikasi fitur-fitur akustik seperti frekuensi, intensitas, dan durasi suara. Nantinya, ini akan dibentuk menjadi sebuah kata atau string.

Pada dasarnya yang diproses dalam komputer nantinya bukanlah suatu suara tapi dalam tipe string. Dalam sebuah mesin pencarian biasanya disimpan sebuah kata pertanyaan dengan jawabannya. Seperti hal nya cara kerja mesin pencarian yang diketahui. Semua string pertanyaan dan jawaban itu biasanya disimpan dalam sebuah database sql atau mongodb atau basis data lainnya.

Hasil proses input user yang menjadi tipe string dari proses *speech recognition* ini akan dibandingkan dengan data-data bertipe string yang ada di database pada mesin pencarian. Ada beberapa algoritma *string matching* yang bisa digunakan untuk ini. Misalnya seperti KMP (Knuth-Morris-Pratt), BM (Boyer-Moore) atau bahkan Brute Force. Tiap algoritma punya kelebihan dan kekurangannya masing-masing. Dengan algoritma *string matching* kita bisa menemukan apakah pertanyaan ada di database atau tidak, jika terbukti ada maka jawaban dari pasangan yang pertanyaan di database nya cocok dengan pertanyaan user akan di-*output*-kan. Dalam implementasinya nanti penulis akan mengembalikan jawaban dengan mengembalikan dalam bentuk suara dengan cara mengkonversi jawaban hasil proses di atas menjadi sebuah suara.

Seperti yang telah dijelaskan tadi ada beberapa algoritma yang cocok untuk melakukan *string matching*. Namun, dalam implementasinya penulis akan menggunakan algoritma string matching Boyer-Moore. Hal ini penulis lakukan karena algoritma *string matching* Boyer-Moore merupakan algoritma yang cukup efektif dalam proses pencarian string. Hal lain yang menjadi pertimbangan adalah dalam penggunaan memori proses pencarian ini juga cukup baik. Dalam penggunaan pola atau pattern yang panjang juga bisa efektif.

Implementasi dari proses di atas nantinya akan dilakukan dengan menggunakan bahasa python. Hal ini penulis lakukan dengan alasan ada library di python yang bisa mengurus berkaitan dengan proses menjadikan suara menjadi sebuah pertanyaan bertipe string.

II. TEORI DASAR

A. *Speech Recognition*

Speech Recognition biasa dikenal dengan Automatic Speech Recognition (ASR) adalah teknologi yang diciptakan untuk mengubah secara otomatis dari ucapan manusia menjadi teks tertulis. Tujuan dari *speech recognition* ini untuk mengenali dan mengartikan kata-kata dan frasa yang diucapkan oleh seseorang melalui suara dan mengubahnya menjadi bentuk teks yang dipahami komputer. Menurut penelitian yang dilakukan oleh Microsoft pada tahun 2019 dihasilkan sebuah fakta bahwa penggunaan suara dalam interaksi dengan perangkat elektronik semakin meningkat. Dari 5000 responden 72% lebih suka menggunakan suara dalam proses pencarian disbanding dengan mengetik pertanyaan itu. Berangkat dari hal ini dapat disimpulkan bahwa *speech recognition* sangatlah penting dan harus terus dikembangkan mengingat kebermanfaatannya juga cukup besar.

Dalam kehidupan sehari-hari *speech recognition* ini banyak dimanfaatkan yaitu sebagai berikut.

- Asisten virtual, Implementasi yang paling umum dari *speech recognition* adalah dalam asisten virtual seperti Siri (Apple), Google Assistant (Google), dan Alexa (Amazon). Asisten ini dapat memahami perintah suara pengguna, menjawab pertanyaan, menjalankan tugas, dan melakukan interaksi suara-manusia.
- Transkripsi Otomatis: *Speech recognition* digunakan dalam aplikasi transkripsi otomatis yang mengubah rekaman suara atau file audio menjadi teks secara otomatis.
- Aplikasi Pencarian: *Speech recognition* digunakan dalam aplikasi pencarian suara yang memungkinkan pengguna untuk melakukan pencarian menggunakan suara mereka.

Masih banyak lagi aplikasi dari *speech recognition* ini dalam kehidupan sehari-hari.

B. Proses *Speech Recognition*

Faktanya, komputer tidak bisa memproses suara secara langsung. Sehingga perlu adanya sebuah proses yang mengubah tipe suara ini menjadi suatu hal yang bisa diproses computer yaitu salah satunya adalah tipe string. Untuk mengubah itu maka dilakukan beberapa tahapan sebagai berikut.

- Perekaman Suara

  Proses ini direkam dengan menggunakan perangkat keras seperti mikrofon. Kemudian sinyal suara itu diubah menjadi format digital yang diproses oleh komputer.

- Proses Ekstasi Fitur

  Proses ekstraksi fitur melakukan analisis sinyal suara untuk mengidentifikasi fitur-fitur akustik yang relevan. Fitur-fitur ini dapat mencakup spektrum frekuensi, energi suara, durasi, atau mel-frequency cepstral coefficients (MFCCs).

- Proses Pemodelan Bahasa

  Proses pemodelan bahasa. Pada proses ini digunakan untuk memahami konteks dan tata bahasa ucapan.

- Proses Pemodelan Akustik

  Proses pemodelan akustik yaitu digunakan untuk memodelkan suara yang direkam. Model akustik telah dilatih menggunakan teknik pembelajaran mesin seperti Gaussian Mixture Models(GMMs) atau Deep Neural Networks(DNNs) dengan menggunakan data-data suara yang berisi contoh ucapan manusia dan transkripsi teks yang sesuai.

- Proses Pencocokkan dan Pengenalan,

  Dalam tahap ini model akustik dan model bahasa bekerja sama untuk mencocokkan pola suara dengan kata-kata atau frasa yang paling sesuai. Algoritma pencocokan digunakan untuk menemukan kecocokan terbaik antara pola suara dan model yang ada. Dan hasil akhirnya adalah teks yang nantinya akan diproses lagi dengan algoritma *string matching*.

C. Proses *String Matching*

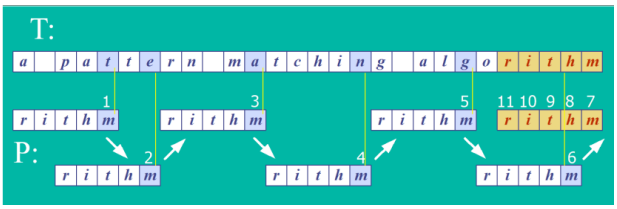
Setelah dalam bentuk teks untuk beberapa penggunaan maka ini akan di cocokkankan. Lebih spesifik akan dijelaskan dalam proses di mesin pencarian. Dalam proses ini bisa digunakan beberapa algoritma *string matching* salah satunya adalah algoritma boyer-moore.

Algoritma boyer-moore adalah algoritma pencocokan pola yang digunakan dalam pemrosesan string untuk mencari kemunculan suatu pola dalam teks. Algoritma ini dikembangkan oleh Robert S. Boyer dan J. Strother Moore pada tahun 1977. Prinsip dasar dari algoritma Boyer-Moore adalah menggunakan informasi dari pola yang dicari untuk mempercepat proses pencocokan dengan menghindari memeriksa setiap karakter dalam teks. Algoritma ini terdiri dari dua langkah utama: tahap pra-pemrosesan (preprocessing) dan tahap pencocokan (matching).

Langkah – Langkah dalam string matching dengan algoritma boyer-moore adalah sebagai berikut.

- Buatlah sebuah tabel yang menyimpan indeks terakhir dari adanya semua karakter yang ada di teks pada pattern
- Setelah itu, lakukan proses pencocokan. Proses pencocokkan dimulai dari karakter paling kanan dalam pola dan berlanjut ke kiri.
- Jika ada ketidakcocokan pada posisi tertentu, tabel di awal yang menyimpan indeks dapat digunakan untuk jumlah pergeseran yang harus dilakukan.
- Algoritma kemudian melanjutkan pencocokan dari posisi baru hasil pergeseran pada tahap selanjutnya.
- Pencocokan terus berlangsung hingga keseluruhan teks dicoba apakah ada yang match atau tidak.

Sebagai contoh akan dilampirkan suatu kasus pencocokkan dengan boyer-moore



contoh penerapan algoritma BM

(source : https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf)

Adapun output yang dihasilkam dari *string matching* itu adalah apakah teks yang berbentuk hasil konversi dari suara

ada kecocokan dengan string di database jika ada maka jawaban dari pertanyaan yang cocok dengan input user itu akan dikeluarkan.

D. Konversi Teks Ke Suara

Konversi teks ke suara digunakan untuk melakukan konversi dari hasil jawaban string matching yang berupa teks menjadi dalam bentuk suara lagi. Bisa dilakukan dengan beberapa langkah salah satunya dengan library yang disediakan oleh bahasa pemrograman python.

E. Python

Python adalah bahasa pemrograman tingkat tinggi yang sering digunakan untuk pengembangan perangkat lunak, pengembangan web, analisis data dan kecerdasan buatan. Ada beberapa karakteristik dari bahasa python yang bisa dikatakan sebagai kelebihan disbanding bahasa lainnya yaitu, sintaks yang bersih dan mudah dibaca, bahasa pemrograman yang terinterpretasi, memiliki berbagai library dan framework dan sebagainya.

Seperti yang telah dijelaskan di atas bahwa python mempunyai banyak library dan framework salah satunya adalah untuk mengelola *speech recognition* ini. Beberapa library itu adalah sebagai berikut.

- Playsound

  Library playsound pada Python digunakan untuk memainkan file audio dalam format WAV atau MP3. Ini adalah library sederhana yang menyediakan fungsi tunggal "playsound()" untuk memutar file audio dengan menggunakan media player default pada sistem operasi yang digunakan.

- Gtts

  Library gtts (Google Text-to-Speech) adalah library Python yang memungkinkan untuk mengonversi teks menjadi ucapan menggunakan layanan Text-to-Speech (TTS) dari Google. Dengan gtts, Anda dapat membuat file audio dari teks yang diinginkan dalam berbagai bahasa yang didukung oleh Google TTS.

- Speech_recognition

  Library ini memungkinkan untuk mengonversi audio suara menjadi teks dengan menggunakan berbagai penyedia layanan speech recognition seperti Google Cloud Speech API.

- OS

  Library os adalah library standar dalam Python yang menyediakan berbagai fungsi untuk berinteraksi dengan operasi file seperti penghapusan file.

III. Penerapan Algoritma Boyer-Moore Dalam Speech Recognition

Seperti yang telah dijelaskan bagian dasar teori langkah-langkah dalam proses penerapan algoritma boyer-moore dalam *speech recognition* ini. Berikut untuk membuktikan bahwa algoritma boyer-moore ini benar-benar membantu dalam dalam *speech recognition* akan dibuat sebuah program, dalam bahasa python. Dengan spesifikasi sebagai berikut.

1. Program akan menerima input dari user. Input dari user ini berupa suara.
2. Program dapat mengeksekusi pertanyaan ketika user sudah berhenti berbicara dalam jeda waktu sekitar 1 detik.
3. Dalam program akan disimpan beberapa pertanyaan beserta jawaban. Mengingat keterbatasan waktu yang penulis punya. Maka, data ini hanya akan disimpan di dalam sebuah array. Bukan disebuah basis data seperti SQL atau lainnya.
4. Program mengkonversi input suara dari user menjadi sebuah teks.
5. Hasil konversi teks dicocokkan dengan pertanyaan yang disimpan di dalam array database oleh program proses pencocokkan dilakukan dengan algorima boyer-moore.
6. Jika ditemukan kecocokkan maka pertanyaan yang mirip denga napa yang user katakan. Pasangan jawabannya akan di-*output*-kan

A. Kebutuhan Instalasi

Dalam proses pembuatan ada beberapa hal yang perlu dipersiapkan yakni:

- Pastikan sudah di-install python
- Siapkan text editor seperti visual studio code
- Library gTTS (bisa install dengan perintah pip install gTTS)
- Library playsound (bisa install dengan perintah pip install playsound)
- Library speechrecognition (bisa install dengan perintah pip install speechrecognition)
- Library PyAudio (bisa install dengan perintah pip install pyAudio)

B. Proses Speech Recognition

Pada tahapan ini ada beberapa hal yang harus dilakukan yakni,

1. Proses Perekaman

   Proses perekaman ini adalah proses dimana kita mendapatkan suara dari user. Proses perekamanan ini bisa menggunakan salah satu library dalam python. Nantinya hasil proses perekaman ini yang akan diolah.

2. Proses Ekstrasi Fitur serta Pemodelan Bahasa, dan Pemodelan Akustik

   Sama seperti halnya proses perekaman proses ini juga menggunakan library dari python yaitu speech_recognition. Disini proses yang terjadi adalah suara hasil dari proses perekaman akan diolah

sedemikian rupa sehingga hasilnya menjadi dalam bentuk teks.

Berikut adalah kode yang merekam dan memproses suara hingga menjadi bentuk teks sehingga nanti bisa diproses oleh algoritma *string matching*.

```
import playsound
from gtts import gTTS
import speech_recognition
import os


def rekam():
    rekam = speech_recognition.Recognizer()
    microphone = speech_recognition.Microphone()
    result = ""
    rekam.pause_treshold = 1
    with microphone as source:
        rekaman = rekam.listen(source)
    try:
        result = rekam.recognize_google(rekaman, language = 'id')
        return result
    except rekam.UnknownValueError:
        result = "Tidak valid"
        return result
    except Exception as e:
        result = e
        return result
```

Kode Proses *Speech Recognition*

(source : Penulis)

Berikut ini, akan ditampilkan hasil dari test case ketika input suara yang diberikan adalah “Selamat Pagi”.

```
print("Hallo ini test case untuk proses speech recognition")
input = rekam()
print("Hasilnyaaaaaaaa")
print(input)
```

```
PS C:\Users\Shelma\Desktop\TugasMakalah> python -u "c:\Users
\Shelma\Desktop\TugasMakalah\suara.py"
Hallo ini test case untuk proses speech recognition
Hasilnyaaaaaaaa
Selamat pagi
PS C:\Users\Shelma\Desktop\TugasMakalah>
```

Test Case Speech Recognition

(source : Penulis)

C. *Proses String Matching*

Seperti yang telah dijelaskan di awal, dalam pengerjaan tugas ini algoritma *string matching* yang digunakan adalah boyer-moore.

Di dalam implementasi algoritma ini diperlukan yang pertama, fungsi buildLast yaitu fungsi yang mendata karakter pada text terakhir ada di index ke berapa di pattern. Adapun untuk implementasi kode buildLast ini dalam python adalah sebagai berikut.

```
def buildLast(pattern):
    table = {}
    for i in range(len(pattern)):
        table[pattern[i]] = i
    return table
```

Kode buildLast

(source : Penulis)

Table hasil buildLast akan digunakan untuk proses string matching dengan boyer-moore Adapun implementasi dari fungsi ini adalah sebagai berikut.

```
def boyer_moore(text, pattern):
    m = len(pattern)
    n = len(text)
    if m == 0:
        return 0

    buildLast_table = buildLast(pattern)
    shift = 0
    while shift <= n - m:
        j = m - 1
        while j >= 0 and pattern[j] == text[shift + j]:
            j -= 1
        if j < 0:
            return shift
        else:
            if text[shift + j] in buildLast_table:
                shift += max(1, j - buildLast_table[text[shift + j]])
            else:
                shift += j + 1
    return -1
```

Kode Proses *String Matching*

(source : Penulis)

Adapun test case untuk men-test apakah fungsi ini berjalan dengan baik adalah sebagai berikut.

1. Untuk pattern yang ada di dalam teks

```
print("Hallo ini test case untuk proses string matching")
text = "Hallo Selamat pagi"
pattern = "Selamat"
print(boyer_moore(text, pattern))
```

```
PS C:\Users\Shelma\Desktop\TugasMakalah> python -u "c:\Users
\Shelma\Desktop\TugasMakalah\bm.py"
Hallo ini test case untuk proses string matching
6
PS C:\Users\Shelma\Desktop\TugasMakalah>
```

Test Case 1 String Matching

(source : Penulis)

2. Untuk pattern yang tidak ada di dalam teks

```
print("Hallo ini test case untuk proses string matching")
text = "Hallo Selamat pagi"
pattern = "Selamate"
print(boyer_moore(text, pattern))
```

```
PS C:\Users\Shelma\Desktop\TugasMakalah> python -u "c:\Users
\Shelma\Desktop\TugasMakalah\bm.py"
Hallo ini test case untuk proses string matching
-1
PS C:\Users\Shelma\Desktop\TugasMakalah>
```

Test Case 2 String Matching

Identify applicable sponsor/s here. If no sponsors, delete this text box (*sponsors).*

(source : Penulis)

Dari sini kita mengetahui jika pattern tidak ada dalam pola-pola teks di database maka return nilainya akan -1. Sedangkan jika ditemukan maka akan me return nilai indeks ketemu pola itu.

D. Proses Konversi Hasil Akhir Menjadi Suara

Karena hasil akhir yang diinginkan penulis adalah seperti asisten google. Dimana jawaban juga disajikan dalam bentuk suara maka hasil proses jawaban akan diubah ke dalam suara hal ini juga memanfaatkan library yang ada di python. Library dan fungsinya adalah sebagai berikut.

1. Library gTTS, library ini untuk mengkonversi dari teks ke dalam suara
2. Library playsound untuk mem-play hasil convert oleh library gTTS

Adapun kode dalam proses ini adalah sebagai berikut

```
def convert_suara(text):
    output_suara = gTTS(text=text, lang = 'id', slow = False)
    output_suara.save("answer.mp3")
    playsound.playsound("answer.mp3", True)
    os.remove("answer.mp3")
```

Kode Proses Konversi Ke Suara

(source : Penulis)

Adapun test case untuk kode ini adalah sebagai berikut.

```
print("Hallo ini test case untuk proses convert suara")
input = "Hallo Selamat Pagi"
convert_suara(input)
```

Test Case Konversi Suara

(source : Penulis)

Dan hasilnya dapat dilihat pada link tautan berikut https://drive.google.com/file/d/1DVejZo0gyoE469hmEQVagbWEYf_xcsU/view?usp=sharing

E. Proses Inti

Proses inti dari program ini adalah sebagai berikut.

- Membuat sebuah database yang nantinya akan diproses pada bagian *string matching* bersama hasil prosess *speech recognition* dari input user. Umumnya, database ini disimpan dalam sebuah basis data seperti SQL atau mongodb. Namun, karena keterbatasan yang penulis miliki, penulis menyimpan database ini dalam dua buah array yaitu, array questions yang berisi pertanyaan-pertanyaan yang disimpan di database dan array answer untuk jawaban dari pertanyaan itu. Pertanyaan dan jawaban dikaitkan dengan kepemilikan indeks yang sama.
- Tugas lain dari program adalah meminta input suara dari user, proses ini dilakukan dengan memanggil fungsi rekam() yang sudah didefinisikan sebelumnya. Dari fungsi rekam ini maka akan diperoleh input dari user.
- Selanjutnya input itu diproses dan menghasilkan sebuah teks. Dalam kode main ini input user dalam bentuk teks disimpan dalam variabel input.
- Kemudian dilakukan *looping* sepanjang array database questions dan dilakukan *string matching* dengan algoritma boyer-moore dengan input tadi yang dilakukan dengan cara memanggil fungsi "boyer_moore" yang sudah didefinisikan sebelumnya.
- Jika dalam proses *looping* ditemukan nilai hasil dari pemanggilan fungsi bukan -1. Maka jawaban dari pertanyaan itu akan diakses dan kemudian di konversi ke dalam bentuk suara. Lalu di play secara otomatis.

```
from bm import boyer_moore
from suara import *

# Database pertanyaan dan jawaban
questions = ["Apa itu wanita?", "Halo siri", "Hai siri bisakah membantu saya",
"Selamat Pagi"]
answers = ["Wanita adalah makhluk ciptaan Tuhan.",
"Hai. selamat menggunakan saya", "Halo saya bisa membantu anda", "Pagi juga"]

# Rekam suara
input = rekam().lower()  # Mengubah input menjadi huruf kecil

# Pencocokan pertanyaan dan jawaban (case-insensitive)
for i, question in enumerate(questions):
    if (boyer_moore(question.lower(), input) != -1):  # Mengubah pertanyaan menjadi huruf kecil
        convert_suara(answers[i])
        break
```

Kode Utama

(source : Penulis)

Agar lebih jelas dilakukan sebuah test case yang dapat dilihat pada tautan ini.
https://drive.google.com/file/d/1mCeeFx7e8m3TRzjqRF0wpyvBBYjMuHVB/view?usp=sharing

Dari hasil implementasi yang telah dilakukan penulis ini membuntikkan bahwa algoritma *string matching* khususnya boyer-moore dapat dimanfaatkan dalam membantu *speech recognition* khususnya dalam menjawab beberapa pertanyaan seperti pada mesin pencarian.

IV. KESIMPULAN DAN SARAN

Algortima string matching dan speech recognition menjadi sebuah kesatuan yang tidak bisa dipisahkan. Pada dasarnya komputer memproses semua informasi dalam bentuk teks atau yang lainnya. Adapun suara tidak bisa diproses langsung oleh komputer. Sehingga, kombinasi diantara keduanya pastilah terjadi. Hal ini menjadi sebuah bukti bahwa lagi-lagi algortima string matching sangat dapat diandalkan dan pasti dibutuhkan terlebih dalam proses-proses *speech recognition*. Tulisan ini hanya salah satu dari aplikasi penerapan antara speech recognition dan string matching kedepannya mungkin bisa dicoba tidak hanya dari input suara itu bisa menjawab pertanyaan mungkin bisa canggih lagi seperti apa yang dilakukan perusahan-perusahaan teknologi sekarang. Seperti siri yang dikembangkan oleh Siri (Apple), Google Assistant (Google), dan Alexa (Amazon). Penulis juga menyarankan

nantinya jika akan melakukan penulisan suatu makalah hendaknya tidak dilakukan dalam waktu yang mepet sehingga hasilnya bisa lebih maksimal.

LINK VIDEO YOUTUBE DAN GITHUB

Untuk mempermudah konsumsi artikel dan lebih menyebarluaskan hasil temuan dari makalah ini, serta beberapa penjelasan mengenai kode secara spesifik penulis juga mencantumkan pranala svideo YouTube untuk ditonton: https://youtu.be/OPuPHmv48ro Adapun untuk source kode lengkap dapat diakses pada link https://github.com/shelmasalsa17/Tugas_Makalah.git

UCAPAN TERIMA KASIH

Penulis mengucapkan puji dan syukur kepada Tuhan Yang Maha Esa atas berkah dan rahmatnya sehingga makalah ini yang berjudul "Penerapan Algoritma Boyer-Moore dalam *Speech Recognition*" dapat diselesaikan dengan baik dan lancar. Penulis juga mengucapkan terima kasih kepada seluruh dosen mata kuliah IF2211 Strategi Algoritma tahun akademik 2022/2023 yang sudah membagikan ilmunya kepada seluruh mahasiswa Teknik Informatika ITB angkatan 2021. Ucapan terima kasih tak lupa penulis ucapkan kepada orangtua dan keluarga yang selalu mendukung penulis, dan teman-teman terdekat penulis yang memberikan dukungan baik dari segi mental maupun materil.

REFERENCES

[1] Munir, Rinaldi. 2021. Pencocokan string (string matching). https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf . Diakses pada 20 Mei 2023.

[2] Laksana, M.Fajar. 2022.Program Sederhana Speech Recognition dengan Python. https://www.youtube.com/watch?v=9N_ZD_IarGU&pp=ygUZc3BlZWNoIHJlY29nbml0aW9uIHB5dGhvbg%3D%3D diakses pada 21 Mei 2023.

[3] Al Faruq B, Herlianto H. Rahendra, Hendrik S.H. Sihar. 2019. Speech Recognition. https://mti.binus.ac.id/2019/05/08/speech-recognition/ diakses pada 20 Mei 2023.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 22 Mei 2023

Ttd

Shelma Salsabila (13521115)