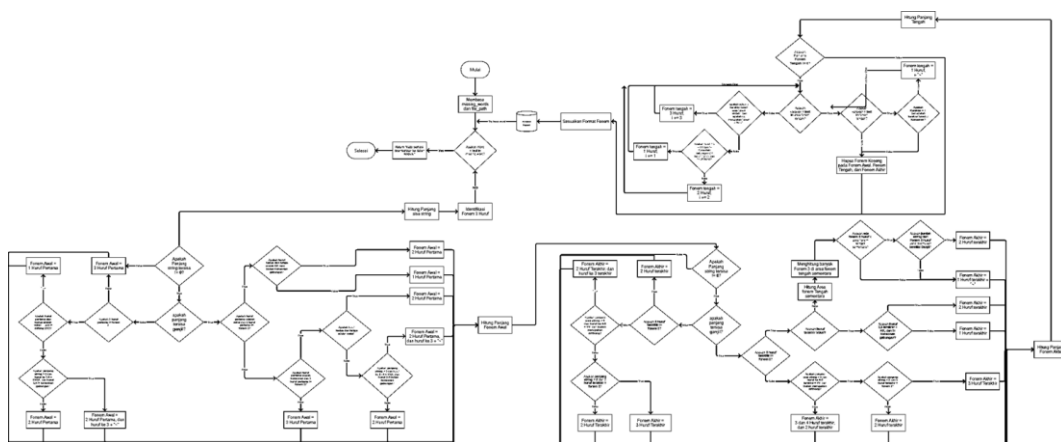


## DAFTAR PUSTAKA

- Atmaja, R. D. (2011). *Perancangan dan Simulasi Kontrol Prosodi Pada Concatenation Synthesizer menggunakan Metode Jaringan Saraf Tiruan Hopfield*. Universitas Telkom.
- Atmanegara, W. S. (2021). *Pedoman umum ejaan Bahasa Indonesia* (T. Kumala, Ed.). Episentrum Books.
- Budiwiyanto, A. (2022, Januari 22). *Korpus dalam Penyusunan Kamus*. Badan Pengembangan dan Pembinaan Bahasa.
- Darwin, D., Anwar, M., & Munir, M. (2021). Paradigma Strukturalisme Bahasa: Fonologi, Morfologi, Sintaksis, dan Semantik. *Jurnal Ilmiah SEMANTIKA*, 2(02). <https://doi.org/10.46772/semantika.v2i02.383>
- Hapsari, J. P. (2014). *Aplikasi Pengenalan Suara Dalam Pengaksesan Sistem Informasi Akademik*.
- Hizbullah, N., Fazlurrahman, & Fauziah, F. (2016). Linguistik Korpus Dalam Kajian Dan Pembelajaran Bahasa Arab Di Indonesia. *Prosiding Konferensi Nasional Bahasa Arab 11*, 1–9.
- Ilham, M., & Mirza, A. H. (2020). Penerapan Algoritma Knuth Morris Pratt Dalam Fitur Pencarian Pengarsipan Dokumen Pada Sma Plus Negeri 17 Palembang. *Journal of Software Engineering Ampera*, 1(2), 110–121. <https://doi.org/10.51519/journalsea.v1i2.49>
- Islamiyah, M., & Fajri, M. S. Al. (2019). Skinny, Slim, dan Thin: Analisis Berbasis Korpus Kata Sifat Identik dan Implikasinya pada Pengajaran Bahasa Inggris. *Ranah: Jurnal Kajian Bahasa*, 8(1), 19. <https://doi.org/10.26499/rnh.v8i1.894>
- Khairan, K. A., & Ahmadian, H. (2019). Penerapan Algoritma Knuth-Morris-Pratt Pada Fitur Pencarian Definisi Istilah Standar Operasional Prosedur (Sop) Pada Lembaga Penjaminan Mutu Uin Ar-Raniry. *Cyberspace: Jurnal Pendidikan Teknologi Informasi*, 3(1), 26. <https://doi.org/10.22373/cj.v3i1.4723>
- Kumara, G. H. (2010). *Visualisasi Beberapa Algoritma Pencocokan String Dengan Java*. Institut Teknologi Bandung.
- Melangi, S. (2018). Text To Speech Bahasa Indonesia Menggunakan Synthesizer Concatenation Berbasis Fonem. *Jurnal Teknik Elektro CosPh*.

- Muslich, M. (2018). *Fonologi bahasa Indonesia Tinjauan Deskripsi Sistem Bunyi bahasa Indonesia* (Cet. 9). Bumi Aksara.
- Suprihatin. (2005, Februari 8). Finite State Automata untuk Parsing (Pemenggalan) Suku Kata dalam Bahasa Indonesia. *Seminar Nasional MIPA* .
- Teunomvira, H., & Gustianingsih. (2022). *Fonem Bahasa Indonesia dan Tamiang: Kajian Komparatif*. 5(2), 1–11.
- Widyawicara. (2021). *Contoh Pemanfaatan Text-to-Speech di Dunia Pendidikan*. [https://widyawicara.com/contoh-pemanfaatan-text-to-speech-di-dunia-  
pendidikan/](https://widyawicara.com/contoh-pemanfaatan-text-to-speech-di-dunia-pendidikan/)
- Wiludjeng, D. F. (2022). *Pembuatan Corpus Berbahasa Indonesia Menggunakan Metode Diphone Concatenation Untuk Sistem Text-to-Speech*. Universitas Hasanuddin.
- Wundari, B. G. (2009). *Sistem Text-To-Speech Dengan Metode Unit Selection Synthesis Untuk Bahasa Indonesia*. Universitas Indonesia.

Lampiran 1. *Flowchart* kategorisasi fonem lengkap



*Flowchart* kategorisasi fonem lengkap dapat diakses di link berikut: [https://bit.ly/flowchart\\_kategorisasi\\_Fonem](https://bit.ly/flowchart_kategorisasi_Fonem)

Lampiran 2. Daftar pengucapan fonem

HURUF	A	B	C	.....	X	Y	Z
A	AA	Batu	Cacing	.....	xamir	yasir	zaman
B	Abu	BB	Coba	.....	xbbua	ybbaa	zbbba
C	Acuan	Bcek	CC	.....	xccar	ycca	zccar
X	Aksara	bq	cx	.....	XX	yxxa	zxxa
Y	Ayam	byur	Cair	.....	xyya	YY	xyya
Z	Azimat	bz	Cize	.....	xzza	yzza	ZZ
.....	.....	.....	.....	.....	.....	.....	.....
Fonem	kan	nga	nya	.....			
3	makan	ngatur	nyayyi	.....			
Fonem	A	B	C	.....	X	Y	Z
1/2	semua	kutub	tac	.....	lex	kuy	laz

Link daftar pengucapan fonem: [https://bit.ly/text\\_pengambilan\\_data](https://bit.ly/text_pengambilan_data)

## Lampiran 3. Format kuesioner

## 1. Kuesioner Skenario 1 Kategorisasi Fonem

No	Data Uji	Jumlah Kata Terkonversi	Akurasi (100%)	Kategorisasi Fonem			Size (Kb)
				<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>	
1	Iya						
2	Ini						
3	Dia						
4	Abu						
5	Air						
6	Kamu						
7	Saya						
8	Ikan						
9	Mama						
10	Lima						
11	Makan						
12	Siapa						
13	Kapan						
14	Lapis						
15	Rumah						
16	Kucing						
17	Anjing						
18	Kancil						
19	Sayang						
20	Negeri						
	Total/Rata-Rata						

## 2. Kuesioner Skenario 2 Kategorisasi Fonem

No	Data Uji	Jumlah Kata Terkonversi	Akurasi (100%)	Kategorisasi Fonem			Size (Kb)
				<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>	
1	Makan Malam						
2	Jalan Sekarang						
3	Minum Kopi						

No	Data Uji	Jumlah Kata Terkonversi	Akurasi (100%)	Kategorisasi Fonem			
				<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>	<i>Size (Kb)</i>
4	Masuk Sekolah						
5	Belok Kanan						
6	Kopi Ini Manis						
7	Kepala Saya Sakit						
8	Saya Mau Makan						
9	Saya Sedang Masak						
10	Ini Kamar Saya						
	Total/Rata-Rata						

### 3. Kuesioner Skenario 3 Kategorisasi Fonem

No	Data Uji	Jumlah Kata Terkonversi	Akurasi (100%)	Kategorisasi Fonem			
				<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>	<i>Size (Kb)</i>
1	Bermain						
2	Menjaga						
3	Menikah						
4	Bersalaman						
5	Berikan						
	Total/Rata-Rata						

### 4. Kuesioner Skenario 4 Kategorisasi Fonem

No	Data Uji	Jumlah Kata Terkonvers i	Akurasi (100%)	Kategorisasi Fonem			
				<i>Intelligibilit y</i>	<i>Fluidit y</i>	<i>Naturalne ss</i>	<i>Size (Kb)</i>
1	Saya pergi ke sekolah untuk belajar dan bermain						
2	Hari ini sangat panas karena telah						

No	Data Uji	Jumlah Kata Terkonvers i	Akurasi (100%)	Kategorisasi Fonem			
				<i>Intelligibilit y</i>	<i>Fluidit y</i>	<i>Naturalne ss</i>	<i>Size (Kb)</i>
	musim kemarau						
3	Kopi yang dibuat sangat manis karena diberi gula banyak						
4	ayah dan ibu membersihka n teras rumah karena banjir						
5	ibu sedang makan ikan gurame yang diberi tetangga						
Total/Rata-Rata							

5. Kuesioner Skenario 1 *Diphone Concatenation*

No	Data Uji	<i>Diphone Concatenation</i>			
		<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>	<i>Size (Kb)</i>
1	Iya				
2	Ini				
3	Dia				
4	Abu				
5	Air				
6	Kamu				
7	Saya				
8	Ikan				
9	Mama				
10	Lima				
11	Makan				
12	Siapa				
13	Kapan				
14	Lapis				
15	Rumah				

No	Data Uji	Diphone Concatenation			
		<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>	<i>Size (Kb)</i>
16	Kucing				
17	Anjing				
18	Kancil				
19	Sayang				
20	Negeri				
Total/Rata-Rata					

6. Kuesioner Skenario 2 *Diphone Concatenation*

No	Data Uji	Diphone Concatenation			
		<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>	<i>Size (Kb)</i>
1	Makan Malam				
2	Jalan Sekarang				
3	Minum Kopi				
4	Masuk Sekolah				
5	Belok Kanan				
6	Kopi Ini Manis				
7	Kepala Saya Sakit				
8	Saya Mau Makan				
9	Saya Sedang Masak				
10	Ini Kamar Saya				
Total/Rata-Rata					

7. Kuesioner Skenario 3 *Diphone Concatenation*

No	Data Uji	Diphone Concatenation			
		<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>	<i>Size (Kb)</i>
1	Bermain				
2	Menjaga				
3	Menikah				

No	Data Uji	Diphone Concatenation			
		<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>	<i>Size (Kb)</i>
4	Bersalaman				
5	Berikan				
	Total/Rata-Rata				

8. Kuesioner Skenario 4 *Diphone Concatenation*

No	Data Uji	Diphone Concatenation			
		<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>	<i>Size (Kb)</i>
1	Saya pergi ke sekolah untuk belajar dan bermain				
2	Hari ini sangat panas karena telah musim kemarau				
3	Kopi yang dibuat sangat manis karena diberi gula banyak				
4	ayah dan ibu membersihkan teras rumah karena banjir				
5	ibu sedang makan ikan gurame yang diberi tetangga				
	Total/Rata-Rata				



## Lampiran 4. Hasil kuisisioner

## 1. Hasil Kuesioner Sesi Kategorisasi Fonem

Responden	Pengujian Keluaran Suara				Mean Opinion Score								
					Skenario 1			Skenario 4					
	1A	1B	.....	4E	Intelligibility	Fluidity	Naturalness	Intelligibility	Fluidity	Naturalness			
					A1 .....	A1 .....	A1 .....	A1 .....	A1 .....	A1 .....			
Pahrul	1	1	.....	8	3 .....	3 .....	3 .....	2 .....	3 .....	3 .....			
Anonim	1	1	.....	8	3 .....	3 .....	3 .....	3 .....	3 .....	3 .....			
Putra	1	1	.....	8	5 .....	4 .....	5 .....	4 .....	4 .....	4 .....			
M. Fadhu Rahman Faisal	1	1	.....	8	4 .....	4 .....	4 .....	3 .....	3 .....	5 .....			
Reski Amalia	1	1	.....	8	2 .....	1 .....	1 .....	4 .....	4 .....	4 .....			
Kyky	1	1	.....	8	5 .....	5 .....	4 .....	4 .....	4 .....	5 .....			
.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....			
.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....			
Agunawan Ali Nur	1	1	.....	8	4 .....	4 .....	4 .....	4 .....	4 .....	4 .....			
Total Akurasi / Nilai	100%	100%	.....	100%	3,62	.....	.....	.....	.....	3,31			

Link Hasil Kuesioner Sesi Kategorisasi Fonem : [https://bit.ly/Survei\\_Kategorisasi\\_Fonem](https://bit.ly/Survei_Kategorisasi_Fonem)

2. Hasil Kuesioner Sesi *Diphone Concatenation*

Responden	Mean Opinion Score											
	Skenario 1						Skenario 4					
	Intelligibility		Fluidity		Naturalness		Intelligibility		Fluidity		Naturalness	
	A1	.....	A1	.....	A1	.....	A1	.....	A1	.....	A1	.....
Pahrul	4	.....	4	.....	3	.....	3	.....	2	.....	3	.....
Anonim	2	.....	3	.....	3	.....	2	.....	3	.....	2	.....
Putra	3	.....	3	.....	3	.....	1	.....	1	.....	1	.....
M. Fadhlul Rahman Faisal	3	.....	4	.....	4	.....	3	.....	4	.....	3	.....
Reski Amalia	2	.....	1	.....	1	.....	4	.....	3	.....	4	.....
Kyky	5	.....	5	.....	5	.....	5	.....	5	.....	4	.....
.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....
Agunawan Ali Nur	4	.....	4	.....	4	.....	4	.....	4	.....	4	.....
<b>Total Akurasi / Nilai</b>	<b>3,07</b>		<b>.....</b>		<b>.....</b>		<b>.....</b>		<b>.....</b>		<b>2,62</b>	

Link Hasil Kuesioner Sesi *Diphone Concatenation*: [https://bit.ly/Survei\\_Diphone\\_Concatenation](https://bit.ly/Survei_Diphone_Concatenation)

Lampiran 5. *Database audio*

The screenshot shows a GitHub repository interface for 'Sistem\_Corpus\_Bahasa\_Dengan\_Integrasi\_KategorisasiFonem / Fonem'. The user 'arif200117' is shown with the option to 'Add files via upload'. The repository was last committed by 'f307d14' 2 months ago. The file tree shows a directory structure with a folder '..' and several audio files: 'a.wav', 'aa.wav', 'ab.wav', 'ac.wav', and 'ad.wav'. Each file has a commit message 'Add files via upload' and a commit date of '2 months ago'.

Name	Last commit message	Last commit date
..		
a.wav	Add files via upload	2 months ago
aa.wav	Add files via upload	2 months ago
ab.wav	Add files via upload	2 months ago
ac.wav	Add files via upload	2 months ago
ad.wav	Add files via upload	2 months ago

Link database Audio: [https://bit.ly/Database\\_Fonem](https://bit.ly/Database_Fonem)

Lampiran 6. *Source code*

```
# Library
import os
import tkinter as tk
from tkinter import filedialog, messagebox
import re
from tkinter import ttk
import wave
import pyaudio
import time
import soundfile as sf

# Variabel global untuk menyimpan lokasi file korpus
WORDS_PRON_DICT = ''
# Ukuran buffer atau blok data adalah 1024 byte
CHUNK = 1024

def remove_punctuation(text):
    text_with_space = text.replace('-', ' ')
    text_without_punctuation = re.sub(r'^[\w\s]', '', text_with_space)
    return text_without_punctuation

def compute_lps(pattern):
    lps = [0] * len(pattern)
    length = 0
    i = 1
    while i < len(pattern):
        if pattern[i] == pattern[length]:
            length += 1
            lps[i] = length
            i += 1
        else:
            if length != 0:
                length = lps[length - 1]
            else:
                lps[i] = 0
            i += 1
    return lps

def kmp_search(text, pattern, start=0, end=None):
    if end is None:
        end = len(text)

    M = len(pattern)
    N = end
```

```

lps = compute_lps(pattern)
i = start
j = 0

while i < N:
    if pattern[j] == text[i]:
        i += 1
        j += 1

        if j == M:
            return True
    else:
        if j != 0:
            j = lps[j - 1]
        else:
            i += 1
return False

# Fungsi untuk mencari pola dalam sebuah file
def search_in_file(pattern, file_path):
    with open(file_path, 'r') as file:
        text = file.read().replace('\n', '')
        return pattern in text

# Fungsi mencari file

# Fungsi Kategorisasi Fonem
def kategorisasi_fonem(missing_words, file_path):
    # Definisi Vokal, Konsonan, difthong, konsonan gabungan, dan fonem 3 huruf

    v = ['a', 'i', 'u', 'e', 'o']
    k = ['b', 'c', 'd', 'f', 'g', 'h', 'j', 'k', 'l', 'm', 'n', 'p', 'q', 'r', 's',
't', 'v', 'w', 'x', 'y', 'z']
    v2 = ['ai', 'au', 'oi']
    k2 = ['kh', 'ng', 'ny', 'sy']
    f3 = ['kan', 'nga', 'nya', 'sya', 'nyi', 'nyo', 'tya', 'syu', 'ter', 'ber',
'per', 'pem', 'pri', 'tur',
'tes', 'pan', 'vei', 'sur', 'men', 'lah']

    #memisahkan kata per kata
    for word in missing_words:
        fonem_awal = []
        fonem_akhir = []
        fonem_tengah = []
        fonem_3 = []

        #mengecek apakah ada fonem 3 huruf dalam kata
        for fonem in f3:
            if kmp_search(word, fonem):
                fonem_3.append(fonem)

        #menghitung panjang string kata tersisa setelah dikurangi panjang fonem 3
        huruf(Jika ada)
        remaining_letters = len(word) - (len(fonem_3) * 3)

        #Menentukan Fonem pertama
        if remaining_letters != 0:
            # kondisi string tersisa = Ganjil
            if remaining_letters % 2 == 1:
                # kondisi huruf pertama = vokal, 3 huruf pertama bukan fonem 3 huruf
                if word[0] in v and word[0:3] not in f3:
                    #melakukan pengecekan pada huruf ke 2 dan ke 3 merupakan KK
                    (konsonan konsonan) yang bukan konsonan gabungan
                    #input "indonesia" output : ["in"]
                    if word[1] in k and word[2] in k and word[1:3] not in k2:
                        fonem_awal.append(word[0:2])
                    if word[1] in v and word[0:2] in v2:
                        fonem_awal.append(word[0:2])
                    # input "angsa" output : ["a"], karena walaupun 'n','g'
                    merupakan konsonan namun ia masuk dalam k2

```

```

else :
    fonem_awal.append(word[0])

# kondisi huruf pertama = konsonan, 3 huruf pertama bukan fonem 3
huruf
elif word[0] in k and word[0:3] not in f3:
    #melakukan pengecekan huruf ke 2 dan ke 3 merupakan VV (vokal
vokal)
    if word[1] in v and word[2] in v:
        fonem_awal.append(word[0:2])
    # kondisi panjang string >3 : pengecekan huruf ke 2,3, dan 4
merupakan VKK (vokal konsonan konsonan)
    # input "contoh" output : ["co", "n~"]
    elif len(word) > 3 and word[1] in v and word[2] in k and word[3]
in k and word[2:4] not in k2:
        fonem_awal.append(word[0:2])
        fonem_awal.append(word[2]+"~")
    #mengatasi jika pola = KVKV / KK+
    else:
        fonem_awal.append(word[0:2])

# kondisi 3 huruf pertama merupakan fonem 3 huruf
else:
    fonem_awal.append(word[0:3])

# kondisi string tersisa = Genap
else:
    #memastikan 3 huruf pertama bukan fonem 3 huruf
    if word[0:3] not in f3:
        #memastikan tidak terjadi pola VV (vokal vokal) yang bukan
difthong
        if word[0] in v and word[1] in v and word[0:2] not in v2:
            fonem_awal.append(word[0])
        #memastikan tidak terjadi pola KK pada karakter ke 3 dan 4
        elif len(word)>3 and word[0] in k and word[1] in v and word[2]
in k and word[3] in k and word[2:4] not in k2:
            fonem_awal.append(word[0:2])
            fonem_awal.append(word[2]+"~")
        # mengatasi selain pola VV, seperti VK KV
        else:
            fonem_awal.append(word[0:2])
    #kondisi 3 huruf pertama merupakan fonem 3 huruf
    else:
        fonem_awal.append(word[0:3])

#kondisi 3 huruf pertama merupakan fonem 3 huruf
else:
    fonem_awal.append(word[0:3])

#menghitung sisa string tersisa setelah dikurang jumlah string fonem pertama
first_fonem = 0
for fonem in fonem_awal:
    if "~" in fonem:
        first_fonem += len(fonem) - 1
    else:
        first_fonem += len(fonem)

remaining_letters2 = len(word) - first_fonem

#kondisi huruf tersisa belum habis
if remaining_letters2 != 0:
    #kondisi string tersisa = ganjil
    if remaining_letters2 % 2 == 1:
        #kondisi 3 huruf terakhir bukan merupakan fonem 3 huruf
        if word[-3:] not in f3:
            #kondisi huruf terakhir = vokal
            if word[-1] in v:
                #kondisi 2 dan 3 huruf terakhir memenuhi pola KK yang bukan
merupakan konsonan gabungan
                if word[-2] in k and word[-3] in k and word[-3:-1] not in
k2:

```

```

        fonem_akhir.append(word[-2:])
    else:
        fonem_akhir.append(word[-1])

    #kondisi huruf terakhir = konsonan
    else:
        #menciptakan ruang fonem tengah sementara
        area_ft = word[first_fonem:len(word) - 1]
        i = 0
        ft3 = []

        #melakukan pengecekan karakter di ruang fonem tengah
        for char in area_ft:
            #melakukan pengecekan apakah ada fonem 3huruf di ruang
fonem tengah
                if i + 3 <= len(area_ft) and area_ft[i:i + 3] in f3:
                    ft3.append(area_ft[i:i + 3])
                    i += 3
                #kondisi ditemukan fonem 3huruf pada ruang fonem tengah
            if ft3:
                #kondisi jumlah string fonem 3 huruf = ganjil
                if len(ft3) % 2 == 1:
                    fonem_akhir.append(word[-2:])
                #kondisi jumlah string fonem 3 huruf = genap
                else:
                    fonem_akhir.append(word[-1:] + "~")
                #kondisi tidak ditemukan fonem 3 huruf pada ruang fonem
tengah
            else:
                fonem_akhir.append(word[-1:] + "~")
        #kondisi 3 huruf terakhir merupakan fonem 3 huruf
    else:
        #memastikan tidak terjadi kondisi vv pada 4 dan 5 karakter
        terakhir yang bukan difthong
        if remaining_letters2 > 4 and word[-4] in v and word[-5] in v
and word[-5:-3] not in v2:
            fonem_akhir.append(word[-4:-2])
            fonem_akhir.append(word[-2:])
        elif remaining_letters2 > 2 and word[-3:] in f3:
            fonem_akhir.append(word[-3:])
        #kondisi apabila string tersisa tidak sampai <3
        else:
            fonem_akhir.append(word[-2:])

    #kondisi string tersisa = genap
    else:
        # kondisi 3 huruf terakhir bukan merupakan fonem 3 huruf
        if word[-3:] not in f3:
            if remaining_letters2 > 4 and word[-3] in v and word[-4] in v
and word[-4:-2] not in v2 or remaining_letters2 > 4 and word[-3] in k and word[-4]
in k and word[-4:-2] not in k2:
                fonem_akhir.append(word[-3:-1])
                if word[-1:] in k:
                    fonem_akhir.append(word[-1:]+ "~")
                else:
                    fonem_akhir.append(word[-1:])
            else:
                fonem_akhir.append(word[-2:])
        #kondisi 3 huruf terakhir merupakan fonem 3 huruf
    else:
        # memastikan tidak terjadi kondisi vv pada 4 dan 5 karakter
        terakhir yang bukan difthong
        if remaining_letters2 > 4 and word[-4] in v and word[-5] in v
and word[-5:-3] not in v2:
            fonem_akhir.append(word[-4:-2])
            fonem_akhir.append(word[-2:])
        elif remaining_letters2 > 2 and word[-3:] in f3:
            fonem_akhir.append(word[-3:])
        # kondisi apabila string tersisa tidak sampai <3
        else:
            fonem_akhir.append(word[-2:])

```

```

#menghitung panjang string fonem akhir
last_fonem = 0
for fonem in fonem_akhir:
    if "~" in fonem:
        last_fonem += len(fonem) - 1
    else:
        last_fonem += len(fonem)

#menghitung ruang fonem tengah
area_ft = word[first_fonem:len(word) - last_fonem]
if area_ft:

    i = 0

    #fungsi semua karakter di ruang fonem tengah
    for char in area_ft:
        #mengecek setiap 3 karakter dalam area fonem tengah, dan mengecek
        apakah itu merupakan fonem 3?
        if i + 3 <= len(area_ft) and area_ft[i:i + 3] in f3:
            fonem_tengah.append(area_ft[i:i + 3])
            i += 3
        else:
            if i + 3 <= len(area_ft):
                if (area_ft[i] in v and area_ft[i+1] in v and area_ft[i:i +
2] not in v2) or (area_ft[i] in k and area_ft[i+1] in k and area_ft[i:i + 2] not in
k2):
                    fonem_tengah.append(area_ft[i])
                    i += 1
                else:
                    fonem_tengah.append(area_ft[i:i + 2])
                    i += 2
            else:
                fonem_tengah.append(area_ft[i:i + 2])
                i += 2

    for i, fonem in enumerate(fonem_tengah):
        if len(fonem) == 1 and fonem in k:
            fonem_tengah[i] = fonem + "~"

#memastikan tidak ada fonem kosong
fonem_awal = [item for item in fonem_awal if item != ""]
fonem_tengah = [item for item in fonem_tengah if item != ""]
fonem_akhir = [item for item in fonem_akhir if item != ""]

if fonem_tengah and fonem_akhir:
    if "~" in fonem_tengah[-1] and fonem_akhir[0] in v:
        fonem_tengah.pop()
        fonem_akhir.pop()
        fonem_akhir.append(word[-2:])

def clean_and_combine(fonem_list):
    #Bersihkan list dari item kosong dan gabungkan dengan tanda +
    return "+".join([item for item in fonem_list if item != ""])

# Gabungkan semua fonem yang ada
combined_fonem = []
for fonem in [fonem_awal, fonem_tengah, fonem_akhir]:
    if fonem:
        combined_fonem.append(clean_and_combine(fonem))

content = word + " " + "+".join(combined_fonem)

with open(file_path, 'a') as file:
    file.write(content + "\n")

return "Kata berhasil ditambahkan ke dalam korpus."

def run_search():
    loaded_words = load_words(WORDS_PRON_DICT)
    # kalimat = kalimat_entry.get()

```

```

kalimat = kalimat_entry.get('1.0', 'end-1c')
corpus_file_path = corpus_file_entry.get()

kalimat = kalimat.lower()
kalimat_without_punctuation = remove_punctuation(kalimat)
kata_kata = kalimat_without_punctuation.split()

file_path = corpus_file_path
found_words = []
missing_words = []

with open(file_path, 'r') as file:
    lines = file.read().split("\n\n")
    for line in lines:
        words = line.split()
        for word in words:
            found_indices = kmp_search(kalimat_without_punctuation,
word.lower())
            if found_indices:
                found_words.append(word)
            else:
                missing_words.append(word)
missing_words = list(set(kata_kata) - set(found_words))

if missing_words:
    result_label.config(text="Kata yang tidak ditemukan dalam korpus: " + ",
".join(missing_words))
    response = messagebox.askquestion("Konfirmasi",
"Apakah Anda yakin kata-kata tersebut
merupakan bahasa Indonesia? Apakah Anda ingin menambahkannya ke dalam korpus?")
    if response == 'yes':
        result_message = kategorisasi_fonem(missing_words, file_path)
        result_label.config(text=result_message)
    else:
        print("Kata tidak ditemukan dalam korpus, pastikan kata merupakan bahasa
Indonesia.")
    else:
        result_label.config(text="Semua kata ditemukan dalam korpus.")

# Fungsi untuk memuat kata-kata dari file korpus
def load_words(words_pron_dict):
    corpus = {}
    with open(words_pron_dict, 'r') as file:
        for line in file:
            if not (line.startswith(';;;') or line.strip() == ''):
                # Lakukan sesuatu dengan baris yang tidak merupakan komentar atau
baris kosong
                key, val = line.split(' ', 2)
                corpus[key] = re.findall(r"[\w~#]+", val)
    return corpus

def get_pronunciation(corpus, str_input):
    list_pron = []
    list_corpus = []
    # print(str_input) #terbaca
    x = re.sub(r"\s+", ' # ', str_input)
    str_input = x

    for word in re.findall(r"[\w~#]+", str_input.lower()): # ambil tiap kata
        if word in corpus: # cek kata ada dalam corpus? tidak terdeteksi
            list_pron.extend(corpus[word])
        else:
            print("word tidak terbaca di corpus")

    delay = 0.145
    result = '\nFonem: {}'.format(list_pron)
    tab1_display.insert(tk.END, result)

    for pron in list_pron:
        sound_fonem = play_audio(pron, delay)

```



```

# Fungsi untuk memainkan audio berdasarkan fonem
def play_audio(sound, delay):
    try:
        time.sleep(delay)
        wf = wave.open("Fonem/" + sound + ".wav", 'rb')
        p = pyaudio.PyAudio()
        stream = p.open(format=p.get_format_from_width(wf.getsampwidth()),
                        channels=wf.getnchannels(),
                        rate=wf.getframerate(),
                        output=True)

        data = wf.readframes(CHUNK)

        while data:
            stream.write(data)
            data = wf.readframes(CHUNK)

        stream.stop_stream()
        stream.close()

        p.terminate()

    return wf
    except:
        pass

# Fungsi untuk mengonversi teks menjadi audio
def convert_audio1():
    loaded_words = load_words(WORDS_PRON_DICT)
    # text_info = text.get()
    text_info = kalimat_entry.get('1.0', 'end-1c')
    get_pronunciation(loaded_words, text_info)

# Fungsi untuk menyimpan audio
def save_audio(corpus, str_input):
    list_pron = []
    dir_fon = "E:\\AI\\Sistem Terbaru fix\\Fonem"
    dir_out = "E:\\AI\\Sistem Terbaru fix\\Output"
    #print(corpus) #terbaca
    x = re.sub(r"\s+", ' # ', str_input)
    str_input = x
    for word in re.findall(r"[\w#]+", str_input.lower()): # ambil tiap kata
        if word in corpus: # cek kata ada dalam corpus? tidak terdeteksi
            list_pron.extend(corpus[word])
        else:
            print("word tidak terbaca di corpus")

    output_audio = []

    for file in list_pron:
        audio_data, _ = sf.read(os.path.join(dir_fon, f'{file}.wav'))
        output_audio.append(audio_data)

    combined_audio = output_audio[0]
    delay = 0.145

    for audio_data in output_audio[1:]:

        silent_samples = int(delay * 44100) # Assuming a sample rate of 44100 Hz
        silence = [0.0] * silent_samples
        combined_audio = list(combined_audio) + silence + list(audio_data)
        # Menambahkan penundaan untuk fonem berikutnya

    output_filename = f'{str_input}.wav' # Tambahkan ekstensi .wav
    output_file = os.path.join(dir_out, output_filename)
    sf.write(output_file, combined_audio, 44100) # Writing the combined audio data

# Fungsi mengeksekusi penyimpanan audio
def saved():
    loaded_words = load_words(WORDS_PRON_DICT)
    # text_info = text.get()
    text_info = kalimat_entry.get('1.0', 'end-1c')

```

```

        save_audio(loaded_words, text_info)

def clear_text():
    tab1_display.delete(1.0, tk.END)
def browse_corpus_file():
    global WORDS_PRON_DICT
    file_path = filedialog.askopenfilename(title="Select Corpus File",
filetypes=[("Text Files", "*.txt")])
    WORDS_PRON_DICT = file_path
    corpus_file_entry.delete(0, tk.END)
    corpus_file_entry.insert(tk.END, file_path)

def retrieve_input():
    global text
    text = kalimat_entry.get('1.0', 'end-1c')

def mulai():
    retrieve_input()
    browse_corpus_file()

# Membuat jendela utama aplikasi
root = tk.Tk()
root.title("TTS With Kategorisasi Fonem")
root.geometry("650x500")

# Membuat tab menggunakan ttk.Notebook
tab_control = ttk.Notebook(root)
tab1 = ttk.Frame(tab_control)
tab_control.add(tab1, text="Project")
tab_control.pack(expand=1, fill='both')

# Membuat elemen-elemen GUI
kalimat_label = tk.Label(tab1, text="Masukkan kalimat:")
kalimat_label.grid(row=0, column=0, padx=5, pady=5, sticky=tk.W)
kalimat_entry = tk.Text(tab1, height=5, width=37)
kalimat_entry.grid(row=0, column=1, colspan=3, padx=5, pady=5, sticky=tk.W)
save_var = tk.Button(tab1, text='Save to Variable "text"', command =
retrieve_input)

corpus_file_label = tk.Label(tab1, text="Lokasi Corpus File:")
corpus_file_entry = tk.Entry(tab1, width=50)
corpus_file_entry.insert(tk.END, "Pilih Corpus Dulu ^ *")
corpus_file_label.grid(row=1, column=0, padx=5, pady=5, sticky=tk.W)
corpus_file_entry.grid(row=1, column=1, padx=5, pady=5, sticky=tk.W)

corpus_file_button = tk.Button(tab1, text="Browse", command=mulai)
search_button = tk.Button(tab1, text="Cek Corpus", command=run_search)
corpus_file_button.grid(row=1, column=2, padx=5, pady=5, sticky=tk.W)
search_button.grid(row=0, column=2, padx=5, pady=5, sticky=tk.W)

result_label = tk.Label(tab1, text="", wraplength=400)
result_label.grid(row=3, column=0, colspan=4, padx=5, pady=5)

tab1_display = tk.Text(tab1)
tab1_display.grid(row=7, column=0, colspan=3, padx=5, pady=5)
button = tk.Button(tab1, text="Convert To Audio", command=convert_audio1,
width="15", bg="#03A9F4", fg="#FFF")
button.grid(row=2, column=0, padx=5, pady=10, sticky=tk.W)
button = tk.Button(tab1, text="Save Audio", command=saved, width="15", bg="#03A9F4",
fg="#FFF")
button.grid(row=2, column=1, padx=5, pady=10, sticky=tk.W)
clear_button = tk.Button(tab1, text="Clear", command=clear_text)
clear_button.grid(row=2, column=2, padx=5, pady=0, sticky=tk.W)
# Menjalankan loop utama GUI
tab1.mainloop()

```

Link Source Code : [https://bit.ly/Source\\_Code\\_Kategorisasi\\_Fonem](https://bit.ly/Source_Code_Kategorisasi_Fonem)

**LEMBAR PERBAIKAN SKRIPSI**

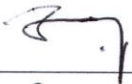

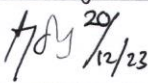
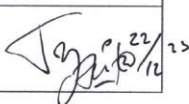
**“IMPLEMENTASI ALGORITMA *KNUTH-MORRIS-PRATT* DALAM SISTEM PEMBUATAN KORPUS BAHASA INDONESIA DAN INTEGRASI KATEGORISASI FONEM UNTUK MENINGKATKAN AKURASI *TEXT-TO-SPEECH*”**

**OLEH:**


**ARIF PUTERA WIJAYA  
D121191065**

Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 20 Desember 2023.  
Telah dilakukan perbaikan penulisan dan isi skripsi berdasarkan usulan dari penguji dan pembimbing skripsi.

Persetujuan perbaikan oleh tim penguji:

	Nama	Tanda Tangan
Ketua	Prof. Dr. Ir. Indrabayu, S.T., M.T., M.Bus.Sys., IPM, ASEAN. Eng.	
Sekretaris	Elly Warni, S.T., M.T.	
Anggota	Dr. A. Ejah Umraeni Salam, ST., MT.	 20/12/23
	Tyanita Puti M W., ST.M.Inf	 22/12/23

Persetujuan Perbaikan oleh pembimbing:

Pembimbing	Nama	Tanda Tangan
I	Prof. Dr. Ir. Indrabayu, S.T., M.T., M.Bus.Sys., IPM, ASEAN. Eng.	
II	Elly Warni, S.T., M.T.	