

SKRIPSI

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

Disusun dan diajukan oleh:

**ARIF PUTERA WIJAYA
D121 19 1065**



**PROGRAM STUDI SARJANA TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
GOWA
2023**

LEMBAR PENGESAHAN SKRIPSI

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

Disusun dan diajukan oleh


**ARIF PUTERA WIJAYA
D121191065**

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian
Studi Program Sarjana Program Studi Teknik Informatika
Fakultas Teknik Universitas Hasanuddin
Pada tanggal 20 Desember 2023
dan dinyatakan telah memenuhi syarat kelulusan

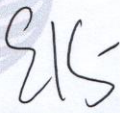
Menyetujui,

Pembimbing Utama,

Pembimbing Pendamping,

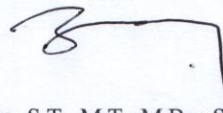

Prof. Dr. Ir. Indrabayu,

S.T., M.T., M.Bus.Sys., IPM, ASEAN. Eng.
NIP 197507162002121004


Elly Warni, S.T., M.T.

NIP 198202162008122001

Ketua Program Studi,


Prof. Dr. Ir. Indrabayu, S.T., M.T., M.Bus.Sys., IPM, ASEAN. Eng.
NIP 197507162002121004

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini;
Nama : Arif Putera Wijaya
NIM : D121191065
Program Studi : Teknik Informatika
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

Implementasi Algoritma *Knuth-morris-pratt* Dalam Sistem Pembuatan Korpus
bahasa Indonesia Dan Integrasi Kategorisasi Fonem Untuk Meningkatkan Akurasi
Text-to-speech

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 27 Desember 2023

Yang Menyatakan



Arif Putera Wijaya

ABSTRAK

Arif Putera Wijaya. *Implementasi Algoritma Knuth-morris-pratt Dalam Sistem Pembuatan Korpus bahasa Indonesia Dan Integrasi Kategorisasi Fonem Untuk Meningkatkan Akurasi Text-to-speech* (dibimbing oleh Indrabayu dan Elly Warni)

Perkembangan teknologi *text-to-speech* (TTS) menjadi salah satu fokus perkembangan kecerdasan buatan dalam bidang *natural language processing* (NLP). Diantara kendala yang masih menjadi problematik adalah ketidakadaan corpus fonem bahasa Indonesia yang mengikuti kaidah bahasa, hal ini menyebabkan hambatan dalam perkembangan TTS bahasa Indonesia.

Oleh karena itu, diperlukan sistem pembuatan korpus fonem bahasa Indonesia secara otomatis yang berkualitas dengan beracuan pada kaidah linguistik bahasa Indonesia.

Penelitian ini menggunakan metode algoritma *knuth-morris-pratt* yang kemudian dihubungkan dengan kategorisasi fonem untuk menghasilkan sistem pembuatan korpus fonem yang lebih baik.

Hasil pengujian menunjukkan tingkat keberhasilan sistem dalam mengkonversi teks menjadi suara memperoleh akurasi 100% pada keseluruhan skenario uji, hal ini menunjukkan keberhasilan sistem kategorisasi fonem dalam menambahkan korpus yang digunakan dalam konversi teks menjadi suara. Penelitian ini juga meninjau kualitas keluaran sistem menggunakan *mean opinion score* (MOS) dan membandingkannya dengan kualitas keluaran sistem yang menggunakan korpus *diphone concatenation*. Hasil perbandingan ini dilihat dari tingkat kejelasan (*intelligibility*), kelancaran (*fluidity*), maupun kealamian ucapan (*naturalness*) korpus yang terintegrasi kategorisasi fonem lebih unggul dengan nilai tingkat kejelasan 3,55, tingkat kelancaran 3,58 dan tingkat kealamian 3,42, sedangkan korpus *diphone concatenation* memiliki tingkat kejelasan 2,87, tingkat kelancaran 2,9 dan tingkat kealamian 2,82. Selain itu, hasil keluaran audio sistem TTS yang menggunakan korpus terintegrasi kategorisasi fonem memiliki ukuran file yang lebih ringan 48% jika dibandingkan dengan TTS yang menggunakan korpus *diphone concatenation*. Oleh karena itu, dapat diputuskan sistem TTS dengan algoritma *knuth-morris-pratt* dan korpus terintegrasi kategorisasi fonem memiliki tingkat keberhasilan konversi dengan sangat baik dan memiliki kualitas keluaran suara yang dapat dipahami dan perangkaian ucapannya cukup jelas, pengucapan yang dihasilkan lancar, transisi antar fonem nyaman, serta memiliki tingkat intonasi yang cukup baik, yang sesuai dengan pengucapan manusia pada umumnya.

Kata Kunci: *Text-To-Speech*, Korpus bahasa Indonesia, Algoritma *Knuth-Morris-Pratt*, *Natural Language Processing*.

ABSTRACT

Arif Putera Wijaya. *Implementation Of The Knuth-Morris-Pratt Algorithm In The Indonesian Language Corpus Creation System And Integration Of Phoneme Categorization To Increase Text-To-Speech Accuracy* (supervised by Indrabayu and Elly Warni)

The development of text-to-speech (TTS) technology is one of the focuses of artificial intelligence development in the field of natural language processing (NLP). Among the obstacles that are still problematic is the absence of a corpus of Indonesian phonemes that follow the rules of the language, this causes obstacles in the development of Indonesian TTS.

Therefore, a system of automatic generation of Indonesian phoneme corpus with quality is needed with reference to Indonesian linguistic rules.

This research uses the knuth-morris-pratt algorithm method which is then connected with phoneme categorization to produce a better phoneme corpus generation system.

The test results show the success rate of the system in converting text into sound obtaining 100% accuracy on the entire test scenario, this shows the success of the phoneme categorization system in adding the corpus used in converting text into sound. This study also reviewed the quality of system output using mean opinion score (MOS) and compared it with the quality of system output using diphone concatenation corpus. The results of this comparison are seen from the level of intelligibility, fluidity, and naturalness of speech. The corpus integrated with phoneme categorization is superior with a value of 3.55 intelligibility level, 3.58 fluency level and 3.42 naturalness level, while the diphone concatenation corpus has a 2.87 intelligibility level, 2.9 fluency level and 2.82 naturalness level. In addition, the audio output of the TTS system using the phoneme categorization integrated corpus has a file size that is 48% lighter than the TTS using the diphone concatenation corpus. Therefore, it can be decided that the TTS system with the knuth-morris-pratt algorithm and the integrated corpus of phoneme categorization has a very good conversion success rate and has a comprehensible sound output quality and the speech string is quite clear, the resulting pronunciation is smooth, the transition between phonemes is comfortable, and has a fairly good intonation level, which is in accordance with human pronunciation in general.

Keywords: Text-To-Speech, Indonesian Language Corpus, Knuth-Morris-Pratt Algorithm, Natural Language Processing.

DAFTAR ISI

LEMBAR PENGESAHAN SKRIPSI	i
PERNYATAAN KEASLIAN.....	ii
ABSTRAK.....	iii
ABSTRACT.....	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR	vi
DAFTAR TABEL.....	vii
DAFTAR LAMPIRAN	viii
KATA PENGANTAR	ix
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Manfaat Penelitian	3
1.5 Ruang Lingkup.....	3
BAB 2 TINJAUAN PUSTAKA	4
2.1 <i>Text-to-speech</i>	4
2.2 Korpus.....	5
2.3 Algoritma <i>knuth-morris-pratt</i>	7
2.4 Kaidah Struktur Linguistik.....	9
2.5 <i>Concatenation Synthesizer</i>	16
2.6 Metode Evaluasi <i>Black box</i> dan <i>Mean Opinion Score</i>	18
BAB 3 METODE PENELITIAN.....	19
3.1 Waktu dan Lokasi Penelitian	19
3.2 Instrumen Penelitian	19
3.3 Tahapan Penelitian.....	19
3.4 Perancangan Sistem	21
3.5 Teknik Pengambilan Data.....	29
3.6 Analisis Kinerja Sistem.....	30
BAB 4 HASIL dan PEMBAHASAN	32
4.1 Hasil Penelitian	32
4.2 Keluaran Sistem	32
4.3 Pengujian Sistem.....	37
BAB 5 KESIMPULAN dan SARAN	51
5.1 Kesimpulan	51
5.2 Saran.....	52
DAFTAR PUSTAKA	53

DAFTAR GAMBAR

Gambar 1. Sistem <i>text-to-speech</i>	5
Gambar 2. <i>Flowchart</i> algoritma <i>knuth-morris-pratt</i>	8
Gambar 3. Diftong menurun	13
Gambar 4. Diftong menaik.....	13
Gambar 5. Proses <i>concatenation text-to-speech</i>	17
Gambar 6. Tahapan penelitian	20
Gambar 7. Rancangan sistem.....	21
Gambar 8. <i>Flowchart</i> kategorisasi fonem.....	23
Gambar 9. <i>Flowchart</i> program fonem awal	26
Gambar 10. <i>Flowchart</i> program fonem tengah.....	27
Gambar 11. <i>Flowchart</i> program fonem akhir	28
Gambar 12. Daftar pengucapan fonem	29
Gambar 13. Ilustrasi pengambilan data suara	30
Gambar 14. Ilustrasi contoh pengambilan data suara untuk fonem “E”	30
Gambar 15. <i>Database</i> fonem	32
Gambar 16. <i>User interface system</i>	34
Gambar 17. Kata yang berhasil dikonversi	35
Gambar 18. Kata yang tidak berhasil dikonversi	35
Gambar 19. Notifikasi konfirmasi.....	36
Gambar 20. Kata berhasil ditambahkan	36
Gambar 21. Text berhasil dibersihkan	36

DAFTAR TABEL

Tabel 1. Huruf vokal	11
Tabel 2. Huruf konsonan.....	11
Tabel 3. Konsonan gabungan.....	13
Tabel 4. Kata berimbuhan	14
Tabel 5. Struktur suku kata bahasa indonesia	15
Tabel 6. Kriteria penilaian tingkat kejelasan (<i>Intelligibility</i>).....	31
Tabel 7. Kriteria penilaian tingkat kelancaran (<i>Fluidity</i>).....	31
Tabel 8. Kriteria penilaian tingkat kealamian (<i>Naturalness</i>).....	31
Tabel 9. Output pasca preprocessing	33
Tabel 10. Output pasca kategorisasi fonem	33
Tabel 11. Pengujian fungsional sistem.....	37
Tabel 12. Hasil pengujian fungsional sistem	38
Tabel 13. Hasil pengujian skenario 1 kaegorisasi fonem.....	40
Tabel 14. Rentang nilai skenario 1 kategorisasi fonem	41
Tabel 15. Hasil pengujian skenario 2 kaegorisasi fonem.....	41
Tabel 16. Rentang nilai skenario 2 kategorisasi fonem	42
Tabel 17. Hasil pengujian skenario 3 kaegorisasi fonem.....	43
Tabel 18. Rentang nilai skenario 3 kategorisasi fonem	43
Tabel 19. Hasil pengujian skenario 4 kaegorisasi fonem.....	44
Tabel 20. Rentang nilai skenario 4 kategorisasi fonem	45
Tabel 21. Perbandingan hasil pengujian kategorisasi fonem dan <i>diphone</i> <i>concatenation</i>	46
Tabel 22. Ringkasan hasil penelitian	49

DAFTAR LAMPIRAN

Lampiran 1. <i>Flowchart</i> kategorisasi fonem lengkap.....	55
Lampiran 2. Daftar pengucapan fonem.....	55
Lampiran 3. Format kuesioner	56
Lampiran 4. Hasil kuisisioner	61
Lampiran 5. <i>Database</i> audio	63
Lampiran 6. <i>Source code</i>	63

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT, yang telah melimpahkan rahmat, hidayah dan karunia-Nya sehingga penulis dapat menyelesaikan penyusunan tugas akhir dengan judul "Implementasi Algoritma *Knuth-morris-pratt* Dalam Sistem Pembuatan Korpus bahasa Indonesia dan Integrasi Kategorisasi Fonem Untuk Meningkatkan Akurasi *Text-to-speech*". Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana (S1) pada Program Studi Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin

Penulis menyadari bahwa skripsi ini tidak mungkin terselesaikan tanpa adanya dukungan, bantuan, bimbingan dan nasehat dari berbagai pihak selama penyusunan skripsi ini. Pada kesempatan ini penulis menyampaikan terima kasih setulus-tulusnya kepada:

1. Allah SWT atas berkat dan Rahmat-Nya sehingga penulis dapat memiliki kehidupan yang baik dan menyelesaikan tugas akhir ini.
2. Kedua Orang tua penulis, Bapak Budi Wijaya, S.Ag., M.Pd. dan Ibu Masirah BSW yang selalu memberikan dukungan, doa dan semangat yang tiada hentinya.
3. Kakak penulis, Fachri Ananda nusantara yang selalu memberikan support dalam perjalanan perkuliahan penulis.
4. Keluarga penulis, Tante Tuti, Om Jaya, Om Herman, Tante Fiha, beserta semua sepupu penulis, telah memberikan tempat dan bantuan kepada penulis dalam beradaptasi di lingkungan baru di Sulawesi Selatan ini.
5. Bapak Prof. Dr. Ir. Indrabayu, S.T., M.T., M.Bus.Sys., IPM, ASEAN. Eng. selaku pembimbing I dan Ibu Elly Warni, S.T., M.T. selaku pembimbing II yang senantiasa meluangkan waktu, tenaga, perhatian untuk memberikan saran yang luar biasa dalam mengarahkan penulis sehingga dapat menyelesaikan tugas akhir ini.
6. Segenap staf, dosen dan civitas akademika Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin yang telah membantu kelancaran hingga tugas akhir dapat terselesaikan.
7. Dr. Ikhwan M. Said, M.Hum yang telah memberikan saran dan membuka wawasan terkait fonetik bahasa Indonesia sehingga memberikan langkah awal dalam pengerjaan tugas akhir ini.
8. Kak Dhinda Fitri Wiludjeng yang selalu dengan senang memberikan masukan dalam pengembangan dalam pembuatan sistem korpus ini.
9. A. Rusmiati, Dea Wahsa, Juan Jimmy, Pahrul, Sabda, Dita, Deby, Citra, Besse, Zid yang menjadi tempat bertanya penulis saat penulis mengalami kendala dan masalah terkait tugas akhir.
10. M. Reza, Giga, Agil, Wira, teman teman dan kakak kakak lab AI yang secara supportive selalu menyemangati dalam penyelesaian skripsi ini.
11. Leon, Rayyan, Putri, A. Diatri, Yusuf, Ilal, Fatur, Hedar, Farhan dan teman-teman Kelas C yang berbagi kenangan menyenangkan selama berkuliah di sini.
12. Sayyid, Afwan, Zaki dan Teman-teman S19NIFIER yang telah berbagi kenangan suka dan duka selama berkuliah di Teknik Informatika Universitas Hasanuddin.

13. Arlin, Dylan, Nadin, May, Ima, Ian, Anggi, Gaby, Nopal, Angel, aswin, Havidz, Vira dan teman teman UKM PSM Universitas Hasanuddin semuanya, yang telah berbagi kenangan berharga.
14. Teman-teman KKNT Gel.108 Posko 8 Desa Garessi kec. Tanete rilau kab. Barru (Tifa, Huusnul, Asifah, Dinar, Nanda, Nabila, wahyudi) yang memberikan pengalaman baru kepada penulis saat KKN.
15. Teman-teman dari Komunitas Kucing Unhas yang telah menjadi tempat menyegarkan diri serta memberikan pengalaman kepada penulis untuk peduli dan bersikap ramah terhadap hewan, khususnya kucing liar.
16. Dan kepada semua pihak lain yang dengan sadar atau tidak sadar telah membantu penulis, yang tidak dapat penulis sebutkan satu per satu, terima kasih atas telah membantu penulis dalam menyelesaikan tugas akhir ini.

Akhir kata, penulis menyadari bahwa skripsi ini masih jauh dari sempurna dan masukan serta kritik membangun sangat penulis harapkan untuk perbaikan di masa mendatang. Semoga skripsi ini dapat bermanfaat dan memberikan kontribusi positif dalam pengembangan ilmu pengetahuan di bidang pembuatan korpus fonem bahasa Indonesia. Aminn

Gowa, 27 Desember 2023

Penulis,
Arif Putera Wijaya

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dalam era digital yang terus bergerak maju, pembacaan suara otomatis atau *Text-to-speech* (TTS) menjadi sesuatu yang menarik untuk dikembangkan. *Text-to-speech* diaplikasikan di berbagai hal seperti dalam aksesibilitas bagi individu dengan disabilitas, sistem informasi multimedia dan sebagainya. Namun, meskipun *Text-to-speech* telah berkembang pesat, masih ada beberapa kendala dalam akurasi pembacaan. Salah satu masalah tersebut adalah ketimpangan pembacaan yang disebabkan oleh kurang baiknya model suara dalam memahami bahasa.

Setiap bahasa memiliki karakteristik unik yang membedakannya dengan bahasa lain atau bisa disebut dengan kaidah fonem bahasa. Perbedaan ini dapat dilihat dalam fonologi, morfologi, atau sintaksis. Fonologi adalah bagian dari tata bahasa atau bidang linguistik yang menganalisis bunyi bahasa secara umum (Teunomvira & Gustianingsih, 2022). sedangkan fonem adalah unit bunyi terkecil yang tidak dapat diamati secara jelas atau tidak diartikulasikan. Fonem terdiri atas fonem vokal, konsonan dan semi vokal (Darwin dkk., 2021) yang mana fonem berbeda memiliki tata cara baca yang berbeda.

Penelitian yang dilakukan oleh Dhinda Fitri Wiludjeng pada tahun 2022 dengan judul "Pembuatan Korpus Berbahasa Indonesia menggunakan Metode Diphone *Concatenation* Untuk Sistem *Text-to-speech*". Penelitian tersebut memperoleh hasil pembuatan Sistem *Text-to-speech* yang mengkonversi teks menjadi suara dengan data suara sendiri dan menggunakan korpus bahasa Indonesia dengan metode Diphone *Concatenation* (Wiludjeng, 2022). Namun hasil penelitian ini masih kurang dikarenakan penambahan korpus dilakukan secara manual serta bentuk fonem yang digunakan pada korpus masih belum sesuai kaidah bahasa Indonesia, sehingga hasil suara yang dikeluarkan kurang baik. Selain itu terdapat penelitian yang dilakukan oleh Mohammad Ilham pada tahun 2020 dengan judul "Penerapan Algoritma *Knuth-morris-pratt* Dalam Fitur Pencarian Pengarsipan Dokumen Pada SMA Plus Negeri 17 Palembang". Penelitian tersebut memperoleh hasil bahwa Algoritma *Knuth-morris-pratt* berhasil diterapkan pada pencarian

pengarsipan dokumen di SMA Plus Negeri 17 Palembang, penelitian tersebut mencatat rata-rata performa algoritma *Knuth-morris-pratt* (KMP) pada form pencarian arsip dokumen sebesar 0.0017 detik dengan dokumen sebanyak 100 data dokumen (Ilham & Mirza, 2020).

Penelitian ini akan menggunakan algoritma KMP dan integrasi kategorisasi fonem untuk membaca dan menambah pada *database* korpus bahasa Indonesia. Algoritma KMP akan membantu dalam proses pencarian kata-kata dalam korpus dan memastikan bahwa kata-kata yang ditemukan sesuai dengan yang diinginkan. Integrasi kategori fonem sesuai kaidah fonem bahasa Indonesia akan membantu dalam meningkatkan akurasi *Text-to-speech*. Dengan demikian, penelitian ini diharap dapat meningkatkan akurasi sistem *Text-to-speech* melalui sistem pembuatan korpus bahasa Indonesia dengan mengkategorisasi fonem yang ada. Hal ini diharapkan dapat memperbaiki dan mengurangi ketimpangan pembacaan kata yang terjadi pada sistem *Text-to-speech*.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut Masalah dalam penelitian ini dirumuskan menjadi beberapa point berikut:

1. Bagaimana implementasi algoritma *Knuth-morris-pratt* dalam sistem pembuatan korpus bahasa Indonesia?
2. Bagaimana integrasi kategorisasi fonem dalam sistem pembuatan korpus bahasa Indonesia untuk *Text-to-speech*?
3. Bagaimana pengaruh integrasi fonem terkategori dalam pembuatan korpus bahasa Indonesia terhadap akurasi pembacaan *Text-to-speech*?

1.3 Tujuan Penelitian

Adapun tujuan penelitian ini adalah untuk :

1. Mengimplementasikan algoritma *Knuth-morris-pratt* untuk mempercepat proses pembuatan korpus bahasa Indonesia.
2. Menganalisis bagaimana kategorisasi fonem diterapkan dalam sistem pembuatan korpus bahasa Indonesia untuk *Text-to-speech*.

3. Menganalisis pengaruh integrasi fonem terkategoriisasi dalam korpus bahasa Indonesia terhadap akurasi pembacaan *Text-to-speech*.

1.4 Manfaat Penelitian

Adapun manfaat yang diperoleh adalah memberikan sistem pembuatan korpus bahasa Indonesia yang lebih cepat, serta memberikan informasi dan wawasan terkait kategoriisasi fonem terhadap akurasi *text-to-speech*.

1.5 Ruang Lingkup

Batasan masalah dalam penelitian ini adalah:

1. Bahasa yang digunakan adalah bahasa Indonesia.
2. Pola struktur kata yang digunakan adalah V, VK, KV, KVK, KKV dan VKK.
3. Tidak ada pembeda dalam pembacaan antara ê dan e.
4. Fonem yang terbentuk dibagi berdasarkan jumlah huruf, yaitu fonem 1 huruf, 2 huruf dan 3 huruf.
5. Input kata baru dalam kategoriisasi fonem minimal 3 huruf.
6. Suara yang dihasilkan belum memperhatikan aspek intonasi dan emosi.

BAB 2 TINJAUAN PUSTAKA

2.1 *Text-to-speech*

Sistem *text-to-speech* juga sering di singkat TTS merupakan suatu sistem yang dapat melakukan konversi *inputan* berupa teks menjadi *output* berupa suara. Sistem ini berkembang sejak abad ke-18 tepatnya pada tahun 1759, Professor Ch. G. Kratzenstein menciptakan rongga resonan (*resonant cavities*) yang *diexcite* (diaktifkan) dengan suatu vibrator sehingga dapat menghasilkan suara dari 5 vokal (huruf hidup) yaitu a, i, u, e dan o (Wundari, 2009). Hingga saat ini, TTS telah menghasilkan *output* suara dari *inputan* teks secara otomatis dengan cara *fonetisasi* yang cara kerjanya adalah dengan menyusun fonem-fonem untuk membentuk ucapan. Seiring dengan perkembangan tersebut, TTS saat ini telah diaplikasikan dalam berbagai bidang diantaranya (Widyawicara, 2021):

1. Pendidikan :

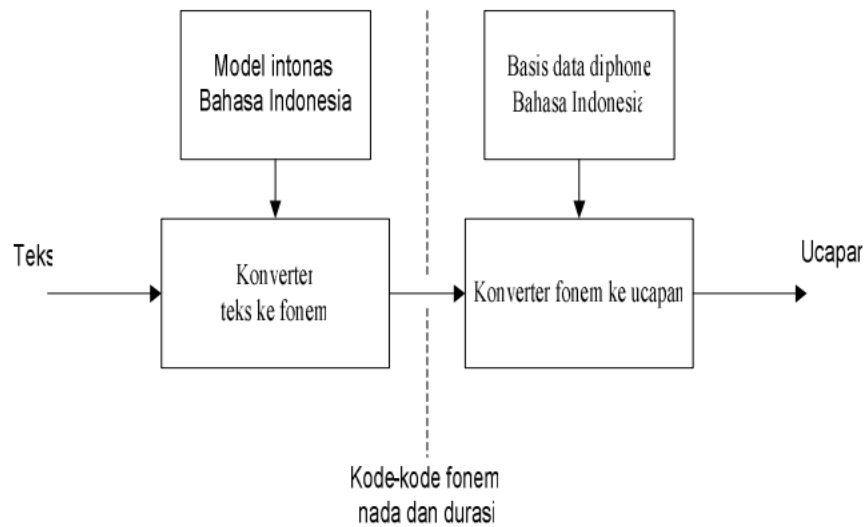
Teknologi TTS memiliki peran yang sangat penting dalam bidang pendidikan. Salah satu manfaatnya adalah membantu siswa dalam mempelajari kata-kata dengan cara mendengarkan pengucapan kata-kata tersebut melalui audiobook. Selain itu, penggunaan audiobook juga dapat meningkatkan kemampuan literasi siswa.

2. Bisnis :

Dalam dunia bisnis, teknologi TTS juga memiliki peranan yang signifikan, terutama dalam membantu perusahaan merespons pertanyaan-pertanyaan umum dari pelanggan melalui *chatbot customer service*. Dengan bantuan TTS ini, pelanggan dapat menerima jawaban yang akurat dan cepat, bahkan di luar jam kerja.

3. Fasilitas Publik :

Teknologi TTS juga dapat dimanfaatkan di fasilitas publik, misalnya saat memberikan pengumuman atau himbauan secara berulang di area publik. Selain itu, penggunaan TTS juga bisa menjadi alat bantu untuk membacakan tulisan-tulisan yang ada di mesin ATM bagi mereka yang memiliki gangguan penglihatan.



Gambar 1. Sistem *text-to-speech*

Sumber: (Hapsari, 2014)

Selain dari apa yang sudah disebutkan sebelumnya, ada banyak penerapan lain dari TTS. Hal ini dikarenakan upaya yang dilakukan dalam pengembangan sistem TTS. Pada umumnya, sistem TTS terdiri dari dua bagian yang dapat dilihat pada Gambar 1 adapun penjelasan kedua bagian tersebut adalah sebagai berikut:

1. Konversi Teks menjadi Fonem :

Bagian ini memiliki peran yang penting dalam mengolah teks yang diberikan dalam bahasa tertentu menjadi serangkaian kode suara yang mewakili bunyi, termasuk durasi dan intonasinya. Kode bunyi ini adalah representasi dari unit-unit suara yang akan diucapkan.

2. Konversi Fonem menjadi Ucapan :

Bagian ini menerima *input* berupa kode-kode bunyi beserta informasi tentang nada dan durasi yang dihasilkan oleh proses konversi sebelumnya. Berdasarkan informasi tersebut, bagian pengubahan bunyi menjadi ucapan akan menghasilkan suara atau sinyal ucapan yang sesuai dengan kalimat yang akan diucapkan.

2.2 Korpus

Korpus merupakan data bahasa yang telah dikumpulkan, diatur dan disimpan dalam format yang dapat digunakan baik itu tertulis maupun digital dan bertujuan untuk menganalisis linguistik, mengembangkan model bahasa, melakukan

penelitian sastra, ataupun berbagai keperluan lainnya. Secara definisi, Menurut Kamus Besar bahasa Indonesia (KBBI), korpus mengacu pada kumpulan ujaran yang tertulis atau lisan yang digunakan untuk menyokong atau menguji hipotesis tentang struktur bahasa. Demikian pula, menurut beberapa sumber korpus didefinisikan sebagai berikut:

1. Adi Budiwiyanto, seorang Koordinator KKLP Perkamusan dan Peristilahan di Badan Pengembangan dan Pembinaan bahasa menerangkan bahwa korpus adalah kumpulan teks alami, baik bahasa lisan maupun bahasa tulis, yang disusun secara sistematis (Budiwiyanto, 2022).
2. Nur Hizbullah, berpendapat bahwa korpus adalah sekumpulan data baik data biasa ataupun data digital, yang mengandung berbagai jenis informasi seperti kata-kata, struktur, makna dan konteks. Korpus digunakan untuk keperluan penelitian (Hizbullah dkk., 2016).
3. Millatul Islamiyah, berpendapat bahwa korpus adalah kumpulan teks autentik, baik tulis maupun transkrip percakapan dalam jumlah besar yang disimpan secara elektronik (Islamiyah & Fajri, 2019).

Perkembangan teknologi menciptakan pembaharuan informasi dan pengetahuan, hal ini juga membuat berkembangnya data kebahasaan yang awalnya berbentuk manuskrip atau tulisan biasa di atas kertas atau objek lain menjadi seperti sekarang dimana data kebahasaan yang berbentuk teks dan bahasa lisan di ubah menjadi bentuk digital sehingga lebih mudah disimpan dan diolah menjadi berbagai produk seperti kamus ataupun di implementasikan dalam aplikasi pengolah suara. Untuk keperluan itu, format korpus bukan hanya kumpulan data bahasa namun mewakili beberapa bagian bahasa sehingga pembuatan korpus bergantung pada tujuan yang ingin dicapai. Beberapa jenis korpus diantaranya (Wiludjeng, 2022):

1. *General Corpus* atau korpus umum, berisi teks–teks umum yang terdiri bukan satu jenis teks atau subjek. Contohnya adalah *Brown Corpus* terdiri dari 1 juta kata LOB Korpus terdiri dari 1 juta kata.
2. *Specialized Corpus* atau korpus khusus adalah korpus yang berisi teks yang dibuat dengan tujuan yang spesifik, seperti untuk memberikan sebuah topik. Contohnya *Michigan Corpus of Academic Spoken English (MICASE)* (*spoken registers in a US academic setting*) terdiri dari 5 juta kata.

3. *Regional Corpora* atau korpus regional, korpus yang dibangun untuk merepresentasikan keberagaman atau variasi bahasa dari suatu daerah, seperti dialek.
4. *Monolingual Corpus* atau korpus satu bahasa. Seperti namanya korpus ini dibuat hanya terdiri dari satu bahasa saja, biasanya dilengkapi dengan anotasi.
5. *Parallel corpus* atau korpus paralel, korpus berisi kata dari dua korpus *monolingual* yang saling terhubung / berpasangan.
6. *Multilingual corpus* atau korpus multibahasa, sepertihalnya korpus paralel korpus ini terdiri atas beberapa korpus *monolingual* yang saling berpasangan.
7. *Comparable Corpus* atau korpus sebanding adalah korpus pasangan dari dua atau lebih korpus *monolingual* yang mana teksnya berhubungan dengan topik yang sama. Contohnya, *International Corpus of English (ICE)* yang terdiri dari 1 juta kata.
8. *Learner Corpus* atau korpus pelajar adalah korpus teks yang dibuat oleh peneliti bahasa. Contohnya *International Korpus of Learner English (ICLE)* terdiri dari 20 ribu kata.
9. *Diachronic Corpus* adalah Korpus yang berisi teks yang dibuat dari periode yang berbeda dan digunakan untuk pengembangan. Contohnya, *Helsinki Corpus – 700 to 1700 texts*, terdiri dari 1,5 juta kata.
10. *Linguistik Corpus* adalah korpus yang terdiri atas sejumlah basis data bahasa yang terakumulasi secara sistematis dari berbagai bidang penggunaan bahasa yang mengikuti beberapa metode statistik dan teknik pengambilan data sampel.

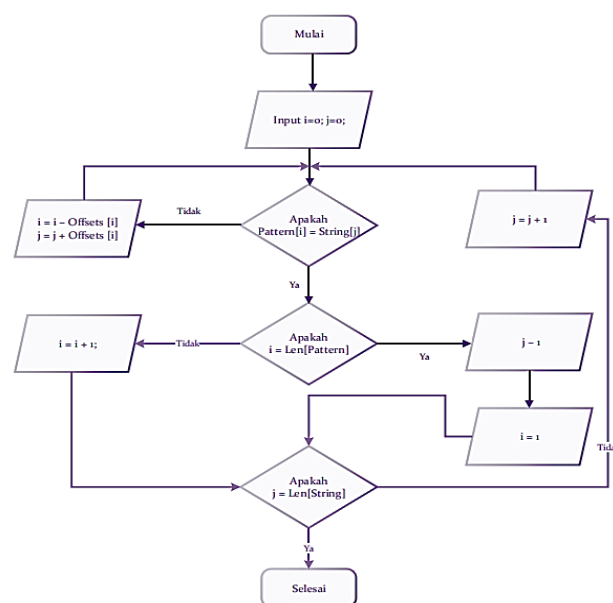
2.3 Algoritma *knuth-morris-pratt*

Algoritma *knuth-morris-pratt* atau yang biasa disingkat KMP, merupakan algoritma yang dikembangkan oleh Donald Knuth, Vaughan Pratt dan James H. Morris pada tahun 1977. Algoritma ini merupakan sebuah algoritma pencocokan pola atau *string matching* yang digunakan untuk mencari kemunculan suatu pola (*pattern*) dalam sebuah teks atau *string (text)* (Ilham & Mirza, 2020).

Algoritma ini dikembangkan sebagai solusi terhadap permasalahan pencocokan *string* dengan waktu eksekusi yang lebih efisien dibandingkan metode

pencocokan *string naif* atau *Boyer-Moore*. Metode pencocokan *string Boyer-Moore* akan mencoba semua kemungkinan posisi awal pencocokan pola dalam teks, yang menyebabkan waktu eksekusi yang tidak efisien terutama untuk pola panjang di dalam teks yang panjang. Sedangkan cara kerja algoritma KMP adalah dengan melewati kemungkinan – kemungkinan yang tidak diperlukan untuk menghindari banyaknya jumlah kemungkinan yang ada, dengan demikian waktu yang diperlukan untuk mencari *string* menjadi lebih efisien (Kumara, 2010). *Flowchart* algoritma ini dapat dilihat pada Gambar 2 Berikut adalah saat pencocokan *string*, adapun tahapan yang ada dalam algoritma KMP adalah sebagai berikut:

1. Algoritma KMP melakukan pencocokan *pattern* pada awal teks
2. Pencocokan akan dimulai dari kiri ke kanan teks, algoritma akan mencocokkan karakter per karakter *pattern*, dengan karakter di teks hingga kondisi terpenuhi. Adapun kondisi sebagai berikut :
 - a. Karakter di *pattern* dan teks tidak ada yang cocok (*mismatch*)
 - b. Semua karakter di *pattern* cocok, algoritma akan memberikan posisi teks yang cocok dengan *pattern*.
3. Algoritma menggeser *pattern* berdasarkan Tabel next, lalu menghitung langkah 2 sampai *pattern* berada di ujung teks



Gambar 2. *Flowchart* algoritma *knuth-morris-pratt*

Sumber: (Khairan & Ahmadian, 2019)

2.4 Kaidah Struktur Linguistik

Struktur linguistik memiliki bagian-bagian dasar diantaranya adalah fonologi, morfologi, sintaksis, semantik, leksikologi dan sebagainya. Setiap bagian memiliki konsentrasinya masing masing, seperti fonologi yang berkonsentrasi pada persoalan bunyi, morfologi pada persoalan struktur internal kata, sintaksis pada susunan kata dalam sebuah kalimat semantik pada persoalan makna kata dan leksikologi pada persoalan pembendaharaan kata. Adapun kedudukan dalam cabang linguistik sebagai berikut (Muslich, 2018):

1. Fonologi dipelajari dengan melihat 2 sudut pandang yaitu:
 - a. Fonetik, bunyi-bunyi ujar dipandang sebagai media bahasa semata atau bisa dibayangkan sebagai benda atau zat. Hal ini menjadikan bunyi dipandang sebagai bahan mentah seperti batu, pasir, semen, atau bahan lainnya yang digunakan dalam membangun rumah.
 - b. Fonemik, bunyi-bunyi ujar dipandang sebagai bagian dari sistem bahasa. Bunyi digambarkan sebagai unsur-unsur bahasa terkecil yang merupakan bagian dari struktur kata dan sekaligus berfungsi membedakan makna.
2. Morfologi mempelajari struktur internal kata (perilaku kata, proses pembentukan kata, sampai dengan nois yang timbul akibat pembentukan kata), sehingga morfologi diperlukan untuk menjelaskan berbagai kasus penulisan struktur kata sebagai berikut:
 - a. Variasi, kata {pukul} memiliki variasi pembacaan antara [pukUl] dan [pUkUl]
 - b. Sufiks, kata {pukul} menjadi [pukulan] setelah ditambah sufiks {-an}
 - c. Prefiks, kata {pukul} menjadi [memukul] setelah ditambah prefix {meN-}
3. Sintaksis mempelajari mengenai makna tataran kalimat, hal ini dapat diketahui dengan memanfaatkan intonasi, jeda dan tekanan dalam kalimat. Contoh perbedaan makna tataran kalimat adalah sebagai berikut :
 - a. Kalimat perintah, pada kalimat [Kamu di sini!]
 - b. Kalimat tanya, pada kalimat [kamu di sini?]
4. Semantik mempelajari mengenai makna kata, dimana menempatkan kapan kata dapat divariasikan ucapannya dan kapan tidak. Contoh penempatan kata semantik adalah sebagai berikut:

- a. Tidak membedakan makna, seperti kata { duduk } jika dibaca [dudUk] atau [dUdUk].
- b. Membedakan makna, seperti kata { tahu } jika dibaca [tahU] dan [tAhU].
5. Leksikologi dan leksokografi mempelajari persoalan pembendaharaan kata suatu bahasa baik dalam rangka penyusunan kamus maupun bukan.
6. Dialektologi mempelajari pemetaan wilayah pemakaian dialek atau variasi bahasa tertentu. Terutama variasi-variasi ucapan pemakaian bahasa, baik secara sosial maupun geografis.
7. Linguistik terapan mempelajari mengenai pengajaran bahasa kedua dan bahasa asing, yang bertujuan mempelajari keterampilan berbahasa lisan bahasa asing.
8. Psikolinguistik mempelajari perkembangan penguasaan bunyi, hal ini untuk mengetahui alasan mengapa bunyi bilabial, lateral, dan vocal rendah-depan lebih dahulu dikuasai dibandingkan bunyi labiodental, tril dan vocal tinggi-belakang.

Sejarah perkembangan kajian fonetik dimulai dari terbentuknya *International Phonetic Association* (IPA) pada tahun 1886, meskipun buku-buku yang membahas tentang bunyi sudah ada sejak tahun 1569. IPA mulai melakukan pengkajian fonetik secara intensif pada tahun 1907 setelah Universitas London mengakui usaha daniel Jones, seorang pakar fonetik dari Inggris. Perkembangan fonetik mengalami kemajuan yang signifikan pada tahun 1930-an, terutama dengan dimulainya kongres-kongres yang mengumpulkan para ahli fonetik di seluruh dunia. Beberapa kongres yang penting antara lain:

1. Kongres Ilmu Fonetik Internasional pertama diadakan pada tahun 1932 di Amsterdam.
2. Kongres Ilmu Fonetik Internasional kedua diadakan pada tahun 1935 di London, di mana dihadiri oleh 262 ahli dari 29 negara.
3. Kongres Ilmu Fonetik Internasional ketiga diadakan pada tahun 1938 di Ghent, Belgia, dengan jumlah peserta sebanyak 273 ahli, 18 lembaga dan persatuan, 38 universitas, serta diwakili oleh 32 negara.

Hingga saat ini, terjadi berbagai perkembangan di bidang fonetik, mulai dari studi tentang pengucapan hingga pembentukan berbagai korpus. Sebagai contoh,

menurut J.D. O'Connor, fonetik dapat didefinisikan sebagai ilmu yang terkait dengan suara-suara ujar yang dihasilkan oleh ucapan manusia, yang kemudian dapat diformulasikan menjadi rangkaian simbol yang memiliki fungsi untuk menyampaikan pesan tertentu. Sebagai contoh, kata "buku" dapat direkonstruksi oleh pendengar menjadi rangkaian bunyi [b][u][k][u].

Adapun kaidah dasar struktur bahasa yang dijelaskan dalam buku pedoman ejaan bahasa Indonesia dan buku fonologi bahasa Indonesia diantaranya sebagai berikut :

1. Huruf Vokal dan Huruf Konsonan

Huruf abjad terdiri atas 26 huruf yang terbagi atas 5 huruf vokal dan 21 huruf konsonan. Huruf vokal dan konsonan memiliki bentuk pembacaan seperti yang ditunjukkan pada Tabel 1. dan Tabel 2. (Atmanegara, 2021).

Tabel 1. Huruf vokal

Huruf Vokal	Contoh Pemakaian Dalam Kata		
	Posisi Awal	Posisi Tengah	Posisi Akhir
a	api	padi	lusa
e*	enak	petak	sore
	ember	pendek	-
	emas	kena	tipe
i	itu	simpan	murni
o	oleh	kota	radio
u	ulang	bumi	ibu

Sumber: (Atmanegara, 2021)

Tabel 2. Huruf konsonan

Konsonan	Contoh Pemakaian Dalam Kata		
	Posisi Awal	Posisi Tengah	Posisi Akhir
b	bahasa	sebut	adab
c	cakap	kaca	-
d	dua	ada	abad
f	fakir	kafan	maaf
g	guna	tiga	gudeg

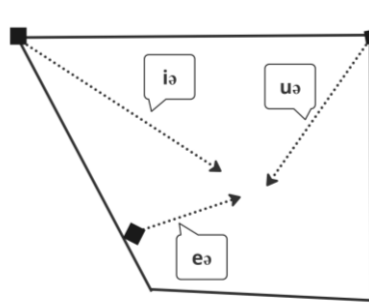
Konsonan	Contoh Pemakaian Dalam Kata		
	Posisi Awal	Posisi Tengah	Posisi Akhir
h	hari	saham	tuah
j	jalan	manja	mikraj
k	kami	paksa	politik
l	lekas	alas	akal
m	maka	kami	diam
n	nama	tanah	daun
p	pasang	apa	siap
q*	qariah	iqra	-
r	raih	bara	putar
s	sampai	asli	tangkas
t	tali	mata	rapat
v	variasi	lava	molotov
w	wanita	hawa	takraw
x*	xenon	-	-
y	yakin	Payung	-
z	zeni	lazim	juz

Sumber: (Atmanegara, 2021)

2. Huruf Diftong

Diftong atau perangkapan bunyi vokoid merupakan proses ketika dua deret vokoid diucapkan dalam satu hembusan suara. Ada 2 macam bunyi diftong yaitu (Muslich, 2018) :

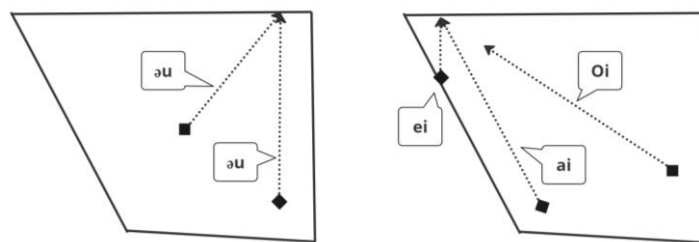
- a. Diftong Menurun (*falling diphtong*), yaitu ketika vokoid pertama bersonoritas sedangkan vokoid kedua kurang bersonoritas bahkan mengarah ke nonvokoid. Kemungkinan terjadinya diftong menurun dapat dilihat pada Gambar 3 (Cth : [a_ura] : “aura”).



Gambar 3. Diftong menurun

Sumber: (muslich, 2018)

- b. Diftong menaik (*rising diphthong*), yaitu ketika vokoid pertama kurang bersonoritas atau mengarah ke nonvokoid sedangkan vokoid kedua bersonoritas. Pada Gambar 4 terlihat bahwa kondisi menaik adalah huruf vokal. (Cth : [koI] : “koi”).



Gambar 4. Diftong menaik

Sumber: (muslich, 2018)

3. Konsonan Gabungan

Merupakan pengucapan huruf konsonan yang dibaca dalam satu bunyi, dalam bahasa Indonesia ada beberapa gabungan yang sering digunakan dapat dilihat pada Tabel 3. (Atmanegara, 2021).

Tabel 3. Konsonan gabungan

Konsonan Gabungan	Contoh Pemakaian Dalam Kata		
	Posisi Awal	Posisi Tengah	Posisi Akhir
kh	khusus	akhir	tarikh
ng	ngarai	bangun	senang
ny	nyata	banyak	-
sy	syarat	musyawarah	arasy

Sumber: (Atmanegara, 2021)

- b. Teori Prominans, menitikberatkan pada gabungan sonoritas dan ciri-ciri suprasegmental, terutama jeda. Ketika rangkaian bunyi diucapkan selain satuan kenyaringan bunyi juga terasa adanya jeda diantaranya, yaitu kesenyapan sebelum dan setelah puncak kenyaringan. Batasan diantara bunyi dilambangkan dengan palang tunggal [+]. Sehingga [mendaki] ditranskrip menjadi [men+da+ki] sehingga memiliki 3 suku kata.

Berdasarkan kedua teori di atas diketahui bahwa sebagian besar suku kata terdiri atas satu bunyi sonor yang berupa vokoid (V) baik tidak didahului dan diikuti kontoid (K), didahului dan diikuti kontoid, didahului kontoid saja, atau diikuti kontoid saja. Pernyataan itu ditulis dalam persamaan (1) berikut :

$$(K) V (K) \quad (1)$$

Adapun struktur suku kata, dalam bahasa Indonesia terdapat 11 pola struktur kata yang terdiri atas vokal (V) dan konsonan (K), seperti yang dijelaskan dalam Tabel 5 berikut (Muslich, 2018):

Tabel 5. Struktur suku kata bahasa indonesia

Struktur Suku Kata	Contoh
V	[a] pada [a+ku] [i] pada [i+bu]
KV	[si] pada [si+ku] [pa] pada [pa+da]
VK	[em] pada [em+ber] [ah] pada [ah+li]
KVK	[tam] pada [tam+pan] [ber] pada [ber+sih]
KKV	[pro] pada [pro+tes]
KKVK	[prak] pada [prak+tis]
KKVKK	[pleks] pada [kom+pleks]
VKK	[eks] pada [eks+por]
KVKK	[seks] pada [seks]
KKKV	[stra] pada [stra+te+gi]
KKKVK	[struk] pada [struk+tur]

Sumber: (Muslich, 2018)

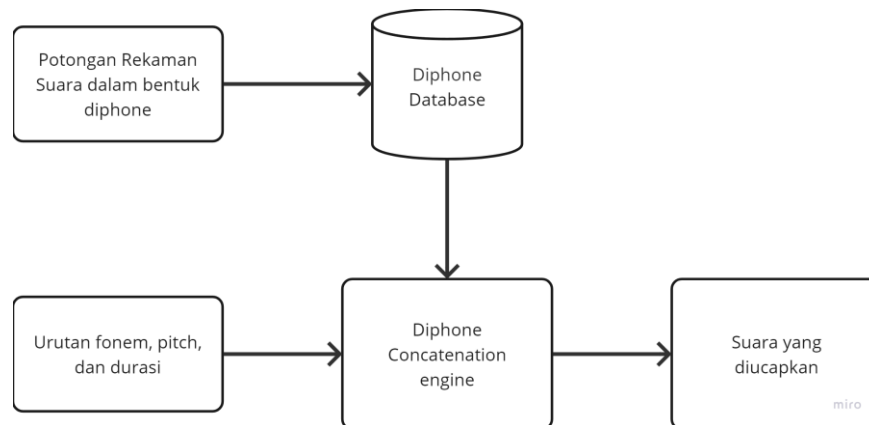
Meskipun terdapat banyak pola namun hanya pola V, VK, KV, KVK yang merupakan pola suku kata asli bahasa Indonesia, sedangkan lainnya merupakan pola suku kata dari bahasa asing dan bahasa daerah yang sudah diadopsi dalam bahasa Indonesia (Suprihatin, 2005).

Mengingat enam dasar aturan di atas, beserta batasan masalah yang ada, yaitu mengenai penggunaan fonem yang terbatas hanya pada fonem yang terdiri dari satu, dua dan tiga huruf, berikut adalah pedoman yang dirancang dalam sistem

1. Apabila ditemukan pola dua vokal berurutan selain diftong, maka memenggalan dilakukan diantara kedua vokal.
2. Apabila ditemukan pola dua konsonan berurutan selain konsonan gabungan, maka dilakukan pemenggalan diantara konsonan tersebut.
3. Apabila ditemukan suku kata konsonan tunggal, maka diberikan tanda “~” yang menandakan pembacaan hanya dibaca ½.
4. Spasi ditandai dengan simbol “#”.
5. Pola yang digunakan adalah pola bahasa Indonesia asli dan imbuhan, yaitu V, VK, KV, KVK, KKV dan VKK.
6. Pola tiga huruf diambil dari kata imbuhan dan juga gabungan antara fonem, contoh : [pergi]:[per+gi], [misal] : [mi+sa+l~], [tampar]:[ta+m~+pan].

2.5 Concatenation Synthesizer

Concatenation synthesizer merupakan proses yang mampu menghasilkan sinyal ucapan otomatis menggunakan transkrip grafemke- fonem (Atmaja, 2011), sehingga dengan proses ini memungkinkan menghasilkan *output* suara dari kata atau kalimat dengan cara menggabungkan (*Concatenation*) potongan fonem atau unit suara yang telah direkam sebelumnya, unit suara tersebut disimpan dalam *database* dalam bentuk gelombang (*uncoded*) atau dalam bentuk *encoded* yang disesuaikan dengan pengkodean ucapan (*speed coding method*) yang dipakai.



Gambar 5. Proses *concatenation text-to-speech*

Sumber: (Melangi, 2018)

Gambar 5 di atas mengilustrasikan cara kerja sintesis *concatenation* berdasarkan rangkaian bagian ucapan yang telah direkam. Dasar dari perangkaian sintesis adalah dengan menggabungkan segmen-segmen dari gelombang ucapan alami yang tersimpan dalam *database*. Segmen-segmen tersebut dapat berupa kata-kata (*words*), unit subkata (*subword unit*) seperti fonem (*phonemes*), diphone (*diphones*) dan suku kata (*syllables*). Sintesis ucapan perangkaian secara luas digunakan, bekerja dengan prinsip pemilihan unit (*unit selection*). Beberapa sistem populer sintesis perangkaian yang menggunakan prinsip pemilihan unit antara lain unisyn, clunits dan multisyn. (Melangi, 2018).

Concatenation synthesizer memberikan keunggulan berupa *database* yang lebih ringan dan pengolahan *database* korpus yang lebih fleksibel dibandingkan metode lainnya. Namun beberapa permasalahan masih belum terpecahkan dengan tuntas yaitu harapan akan tercapainya *Concatenation Synthesizer* yang memenuhi kualitas terdengar jelas (*intelligibility*) dan kealamian (*naturalness*) (Wiludjeng, 2022).

Oleh karena itu, dalam dunia teknologi audio, khususnya pada pengembangan suara sintesis, suara sintesis secara umum harus memenuhi tiga karakteristik utama, yakni tingkat pemahaman (*intelligibility*) untuk memastikan bahwa teks yang dihasilkan dapat dimengerti oleh pendengar, kelancaran pengucapan (*fluidity*) untuk menjamin lancarnya transisi antara suku kata dan kealamian pengucapan

(*naturalness*) untuk memastikan bahwa teks yang dihasilkan terdengar serupa dengan pengucapan manusia pada umumnya (Melangi, 2018).

2.6 Metode Evaluasi *Black box* dan *Mean Opinion Score*

Dalam melakukan evaluasi untuk menguji kinerja sistem, terdapat berbagai metode yang dapat diterapkan. Diantaranya adalah metode *black box* dan *Mean Opinion Score* (MOS). Penjelasan mendalam tentang kedua metode ini diberikan sebagai berikut.

1. Evaluasi *Black box*

Metode ini dikenal juga sebagai metode fungsional, adalah sebuah teknik pengujian di mana fokus utama adalah pada *input* dan *output* sistem tanpa mempertimbangkan bagaimana sistem tersebut bekerja di dalamnya. Dalam kata lain, tester/responden tidak memerlukan pengetahuan tentang kode internal, struktur, atau cara kerja dari perangkat lunak yang diuji (Wiludjeng, 2022). Metode ini sangat penting karena memungkinkan pengujian sistem secara holistik, dengan memastikan bahwa fungsionalitas keseluruhan sistem sesuai dengan harapan, tanpa memerlukan pemahaman teknis yang mendalam tentang internal sistem.

2. Evaluasi *Mean Opinion Score* (MOS)

Metode ini merupakan metode pengujian yang dilakukan secara subjektif dengan cara melakukan pembagian kuesioner pada penguji. MOS memberikan pemahaman yang lebih mendalam tentang bagaimana suara diinterpretasikan dan diterima oleh pengguna, dengan mempertimbangkan faktor subjektif yang tidak dapat diukur secara langsung melalui metode analisis teknis atau matematis. Adapun perhitungan yang digunakan ditunjukkan pada persamaan (2) berikut (Wiludjeng, 2022):

$$MOS = \frac{\text{jumlah nilai yang diberi responden}}{\text{jumlah responden}} \quad (2)$$

BAB 3

METODE PENELITIAN

3.1 Waktu dan Lokasi Penelitian

Penelitian ini dilaksanakan sejak turunnya surat penugasan penelitian pada tanggal 16 Februari 2023. Penelitian ini dilakukan di Laboratorium Kecerdasan Buatan Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

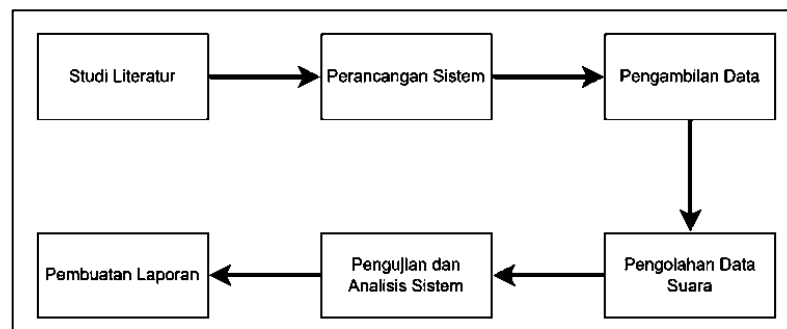
3.2 Instrumen Penelitian

Penelitian ini menggunakan dua komponen, yaitu *software* dan *hardware*. Adapun instrumen penelitian yang digunakan adalah sebagai berikut:

1. *Hardware*:
 - a. Laptop Asus Vivobook 14 X412DA (AMD Ryzen 3 3200U with Radeon Vega Mobile Gfx, 2,60 GHz)
 - b. Mic Audio Technical at 2040
 - c. Samson Pop Filter
 - d. Hercules Stand Mic
 - e. Kabel Jack XLR
 - f. Alctron PF36B
 - g. Ruangan Studio
2. *Software*:
 - a. Windows 11 x64
 - b. PyCharm
 - c. Audacity

3.3 Tahapan Penelitian

Tahap-tahap dalam penelitian ini dapat dilihat pada Gambar 6.



Gambar 6. Tahapan penelitian

1. Studi Literatur

Studi literatur merupakan tahap awal dari penelitian, pada tahap ini penulis menganalisis masalah yang ada dan mencari penelitian yang berkaitan dengan permasalahan yang ingin diselesaikan, yaitu terkait algoritma *Knuth-morris-pratt* yang akan diterapkan, kaidah fonem bahasa Indonesia, korpus fonem *diphone Concatenation* dan Sistem *text-to-speech*.

2. Perancangan Sistem

Perancangan sistem merupakan tahap awal dalam penelitian ini, kesatuan sistem dibangun atas beberapa program yang harus dipersiapkan yaitu program pencaharian korpus, program kategorisasi fonem, program *text-to-speech* dan program simpan audio.

3. Pengambilan Data

Pengambilan data dilakukan dengan melakukan rekaman suara di studio milik Laboratorium Kecerdasan Buatan Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin, rekaman ini nantinya akan diolah sehingga menjadi *database* audio yang akan dipergunakan dalam merangkai suara.

4. Pengolahan Data Suara

Pengolahan data suara merupakan proses pemotongan sinyal suara rekaman awal menjadi bentuk audio fonem yang telah ditentukan.

5. Pengujian dan Analisis Sistem

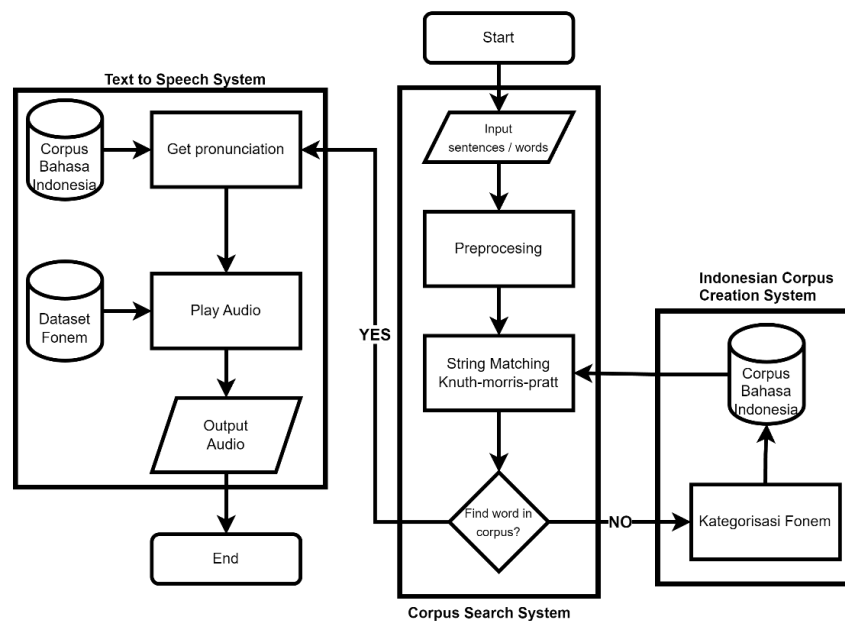
Setelah keseluruhan sistem dan *database* audio selesai, akan dilakukan uji coba untuk melihat apakah hasil dari penggunaan algoritma *Knuth-morris-pratt* dapat berjalan dengan baik yang dapat dilihat dari hasil uji *black box* serta

apakah penerapan kategorisasi fonem dapat meningkatkan akurasi *text-to-speech* melihat nilai dari *Mean opinion score* (MOS).

6. Pembuatan Laporan

Setelah dilakukan pengujian dan evaluasi sistem, selanjutnya masuk pada tahap akhir yaitu, penulisan laporan penelitian dalam bentuk skripsi.

3.4 Perancangan Sistem



Gambar 7. Rancangan sistem

Gambar 7 menampilkan rancangan sistem penelitian. Penelitian ini bertujuan untuk membangun sebuah sistem yang terbagi menjadi tiga bagian utama. Pertama, terdapat *corpus search system* (program pencarian korpus) yang menggunakan algoritma *Knuth-Morris-Pratt* untuk melakukan pencarian kata *input* dalam korpus. Kedua, terdapat *indonesian corpus creation system* (program pembuatan korpus) yang bertujuan untuk menambahkan kata-kata baru ke dalam korpus dengan menggunakan kategorisasi fonem yang merupakan sistem untuk mengkategorikan fonem dalam penelitian ini fonem dikategorikan dalam fonem 1 huruf, fonem 2 huruf, dan fonem 3 huruf. Ketiga, terdapat *text-to-speech system* (program konversi teks menjadi suara) yang bertujuan untuk mengonversi fonem menjadi *output* suara

sesuai dengan *input* teks, semua *source code* dari rancangan sistem ini dapat dilihat pada Lampiran 6. Berikut adalah penjelasan dari masing-masing program:

1. Program pencarian korpus

- a. *Input* teks

Pada tahap ini, program akan menerima *input* berupa teks yang nantinya akan dikenali oleh sistem.

- b. *Preprocessing* teks

Pada tahap ini, teks yang di *input* sebelumnya akan diseragamkan ke dalam huruf kecil, dihapus tanda bacanya dan dipecah menjadi kata per kata. Misalnya teks yang menjadi masukan adalah “Siapa yang pergi ke pasar?” maka hasil akhirnya setelah diseragamkan, dihapus tanda baca dan dipecah kata per kata adalah sebagai berikut “[siapa], [yang], [pergi], [ke], [pasar]”.

- c. *String matching word and corpus*

Pada tahap ini, setiap kata akan dicari kedalam korpus menggunakan algoritma *knuth-morris-pratt*, yaitu dengan cara membandingkan *substring* (teks) dalam *string* (*database* korpus). Apabila kata tidak ditemukan dalam korpus maka kata tersebut akan masuk dalam program kategorisasi fonem, sebaliknya apabila ditemukan, akan lanjut ke kata selanjutnya hingga semua kata habis dan dapat lanjut dalam program *text-to-speech*.

2. Program *text-to-speech*

- a. Get Pronunciation

Pada tahap ini, teks akan di konversi menjadi fonem, fonem diambil dari *database* korpus yang sebelumnya dihasilkan dari program kategorisasi fonem, sehingga ini memastikan 100% *text* dapat melakukan konversi fonem. Hasil konversi inilah yang akan digunakan untuk menghasilkan *output* suara yang sesuai dengan *input* teks.

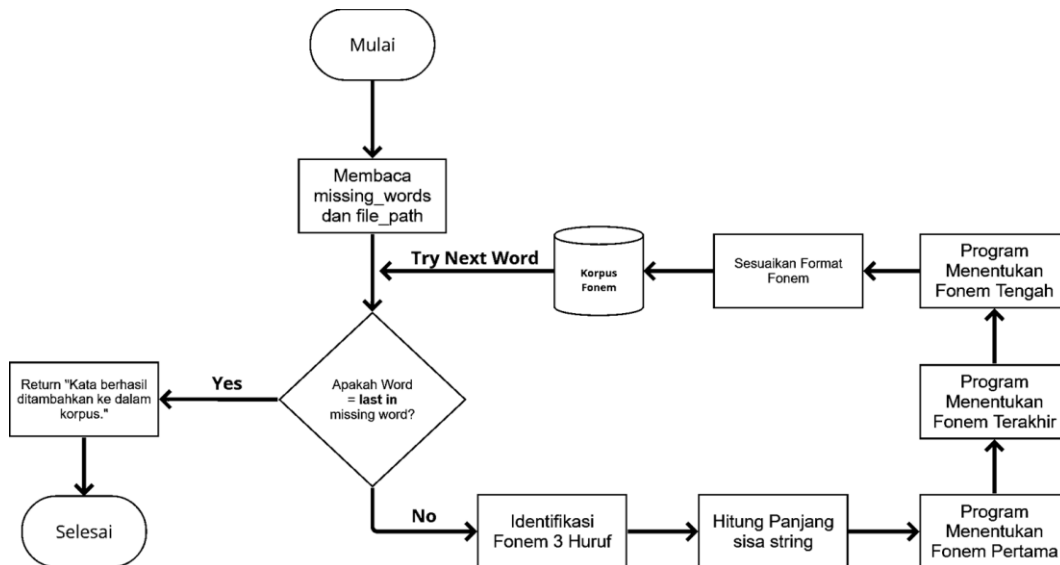
- b. Playing Audio

Pada tahap ini, dilakukan konversi fonem menjadi audio, proses ini dilakukan dengan cara mencocokkan fonem dengan *database* audio yang

telah dibuat. Setelah dicocokkan maka audio akan diambil dan diputar berurutan sesuai dengan label suara fonem.

3. Program kategorisasi fonem

Program kategorisasi fonem dapat dilihat dari alur sistem yang tertera pada Gambar 8, dalam gambar tersebut terlihat *flowchart* kategorisasi fonem yang telah disederhanakan. *Flowchart* lengkapnya ada pada Lampiran 1



Gambar 8. *Flowchart* kategorisasi fonem

Pada Gambar 8, terlihat beberapa tahapan yang harus dilalui untuk menentukan kategori fonem pembentuk kata, mulai dari penentuan fonem awal, fonem akhir dan fonem tengah, hingga penyimpanan fonem dalam korpus. Dengan mengikuti pedoman kaidah struktur linguistik, dibuatlah program untuk menentukan fonem awal, fonem akhir dan fonem tengah. Adapun *Flowchart* dari tiga program ini dapat dilihat pada Gambar 9, Gambar 10 dan Gambar 11.

a. Identifikasi Fonem 3 Huruf

Pada tahap ini, program akan mengidentifikasi apakah di dalam kata terdapat fonem 3 huruf (kan, nga, nya, sya, nyi, nyo, tya, syu, ter, ber, per, pem, pri, tur, tes, pan, vei, sur, men, lah, kah). Setelah itu, dilakukan penghitungan panjang *string* yang dikurangi jumlah fonem 3 huruf (jika ada).

b. Program Menentukan Fonem Awal

Pada tahap ini, program akan menentukan apakah fonem awal terbentuk dari 1, 2, atau 3 huruf. Seperti yang dilihat pada Gambar 9 untuk mencapai hal tersebut, kata akan melewati beberapa tahapan, yaitu:

- 1) Melakukan perhitungan panjang sisa *string*. Jika panjang *string* kosong, proses akan langsung beralih ke tahap 4, yaitu penentuan bentuk fonem awal.
- 2) Menentukan apakah panjang sisa *string* merupakan bilangan ganjil atau genap untuk menentukan pola kategorisasi lebih lanjut.
- 3) Memeriksa pola huruf vokal dan konsonan dalam sisa *string* untuk memastikan bahwa tidak ada pelanggaran aturan pedoman linguistik, misalnya, menghindari VV yang bukan merupakan diftong atau KK yang bukan konsonan gabungan.
- 4) Menentukan bentuk fonem awal
- 5) Menghitung panjang *string* fonem awal

c. Program Menentukan Fonem Terakhir

Seperti yang dilihat pada Gambar 11 pada tahap ini program akan menentukan apakah fonem terakhir terbentuk dari 1, 2, atau 3 huruf. Namun, perlu diperhatikan bahwa mungkin tidak ada fonem akhir jika panjang *string* yang tersisa sudah habis. Langkah-langkahnya meliputi:

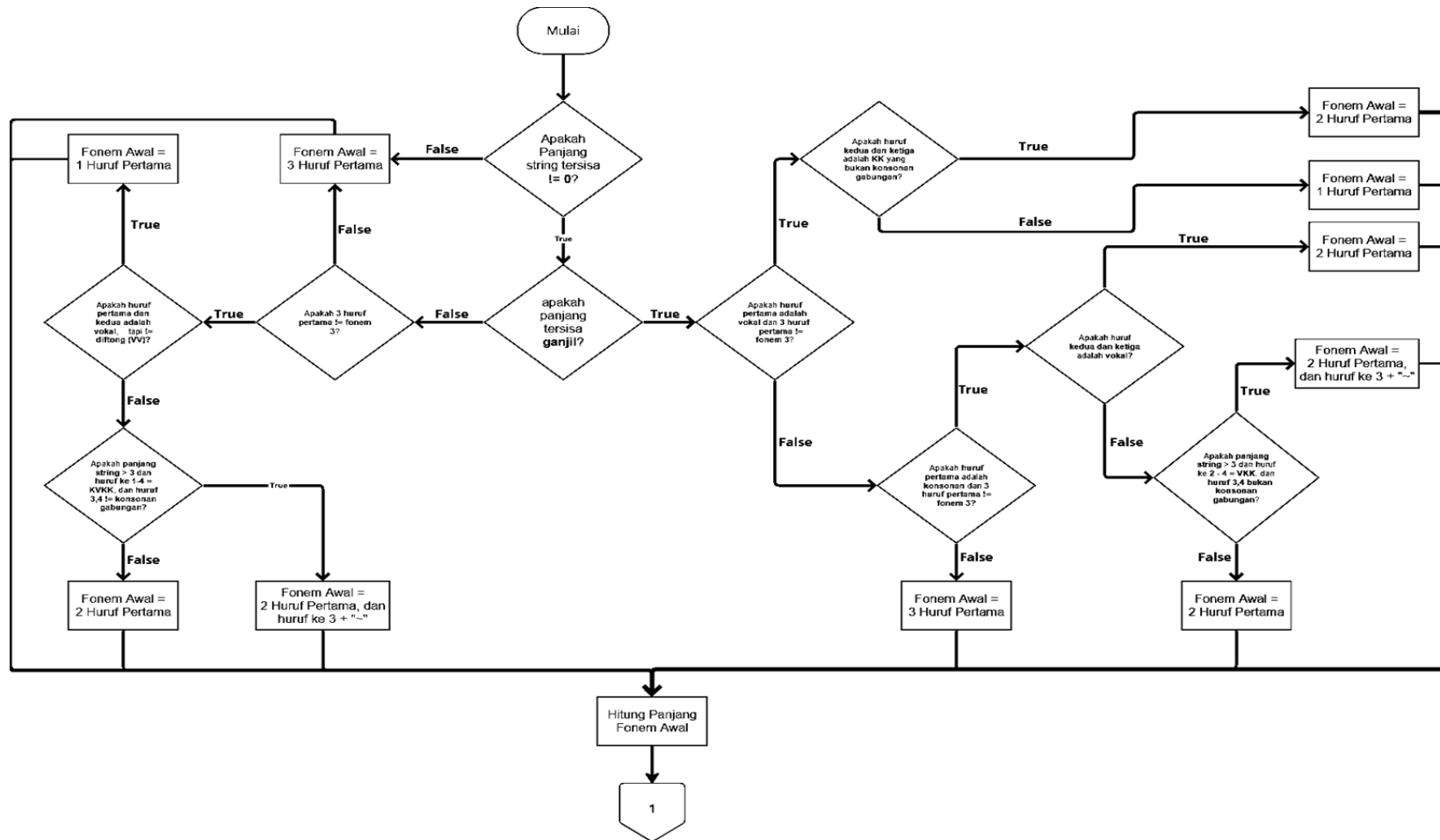
- 1) Melakukan perhitungan panjang sisa *string*. Jika panjang *string* kosong, proses akan langsung beralih ke tahap 5, yaitu Menghitung panjang fonem terakhir.
- 2) Menentukan apakah panjang sisa *string* merupakan bilangan ganjil atau genap untuk menentukan pola kategorisasi lebih lanjut.
- 3) Memeriksa pola huruf vokal dan konsonan dalam sisa *string* untuk memastikan bahwa tidak ada pelanggaran aturan pedoman linguistik, misalnya, menghindari VV yang bukan merupakan diftong atau KK yang bukan konsonan gabungan.
- 4) Menentukan bentuk fonem terakhir.
- 5) Menghitung panjang fonem terakhir.

d. Program Menentukan Fonem Tengah

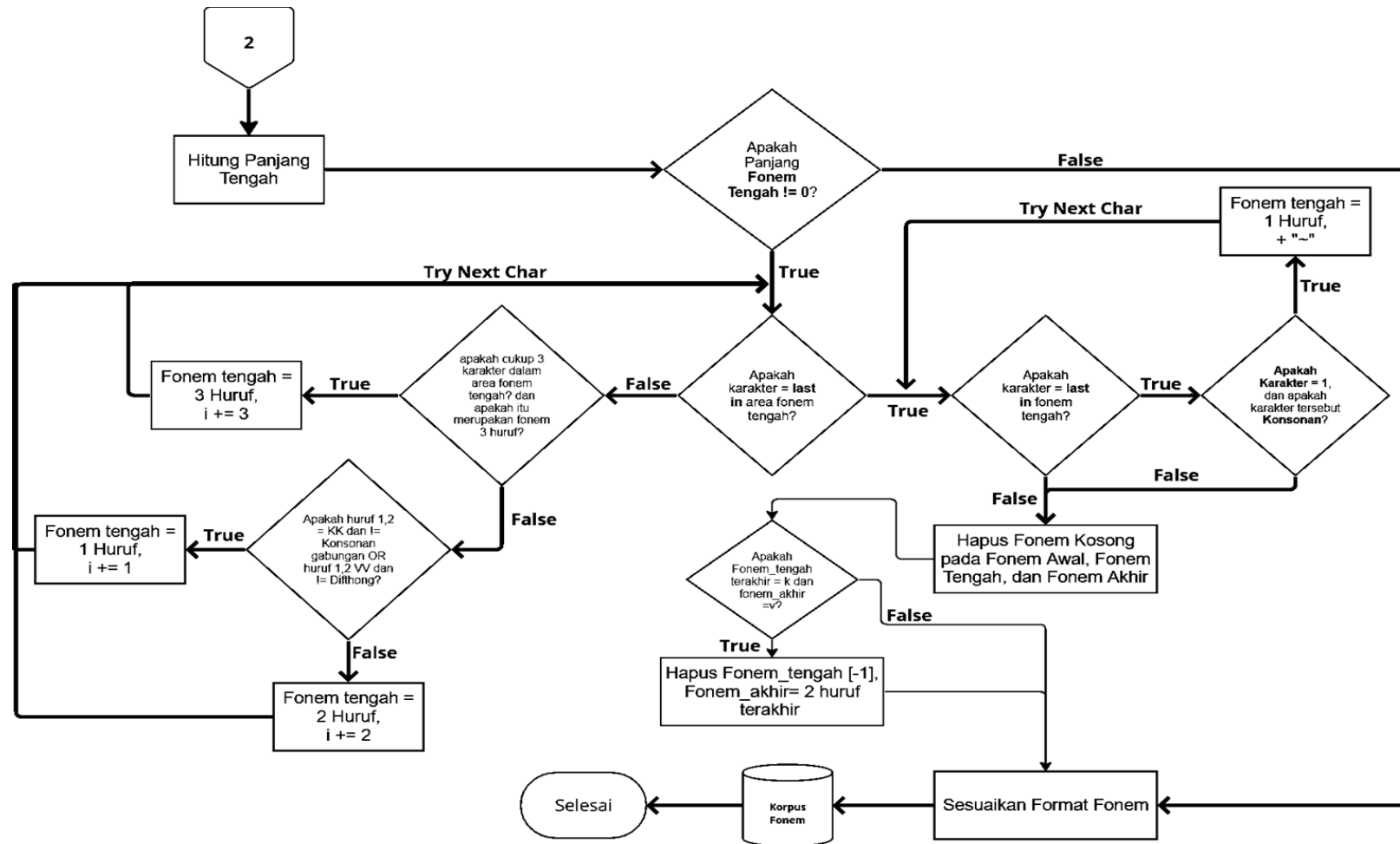
Seperti yang dilihat pada Gambar 10 pada tahap ini, serupa dengan penentuan fonem terakhir, mungkin tidak akan ada fonem tengah jika panjang *string* yang tersisa sudah habis. Alur program yang digunakan untuk menentukan fonem tengah cenderung lebih sederhana karena umumnya fonem tengah terdiri dari 2 huruf. Langkah-langkahnya meliputi:

- 1) Melakukan perhitungan panjang sisa *string*. Jika panjang *string* kosong, proses akan langsung beralih ke tahap 5, yaitu Menghapus fonem kosong.
- 2) Memeriksa setiap 3 *string* pertama pada fonem tengah. Jika membentuk fonem 3 huruf, maka polanya adalah fonem 3 huruf. Jika tidak, masuk ke tahap iii untuk memastikan tidak ada pelanggaran aturan pedoman.
- 3) Memeriksa pola huruf pertama dan kedua untuk memastikan tidak ada pelanggaran aturan pedoman linguistik. Hal ini termasuk menghindari VV yang bukan merupakan diftong atau KK yang bukan konsonan gabungan. Jika pelanggaran ditemukan, pola dihitung sebagai 1 huruf; jika tidak, pola dihitung sebagai 2 huruf.
- 4) Memeriksa setiap *string* dari fonem tengah yang terbentuk sebelumnya. Jika terdapat fonem tunggal, akan diperiksa apakah merupakan konsonan atau vokal. Jika konsonan, maka akan diberikan tanda “~”.
- 5) Menghapus fonem kosong dari variabel yang menyimpan nilai fonem awal, fonem akhir dan fonem tengah.
- 6) Menyesuaikan format korpus dan menyimpan dalam *database* korpus.

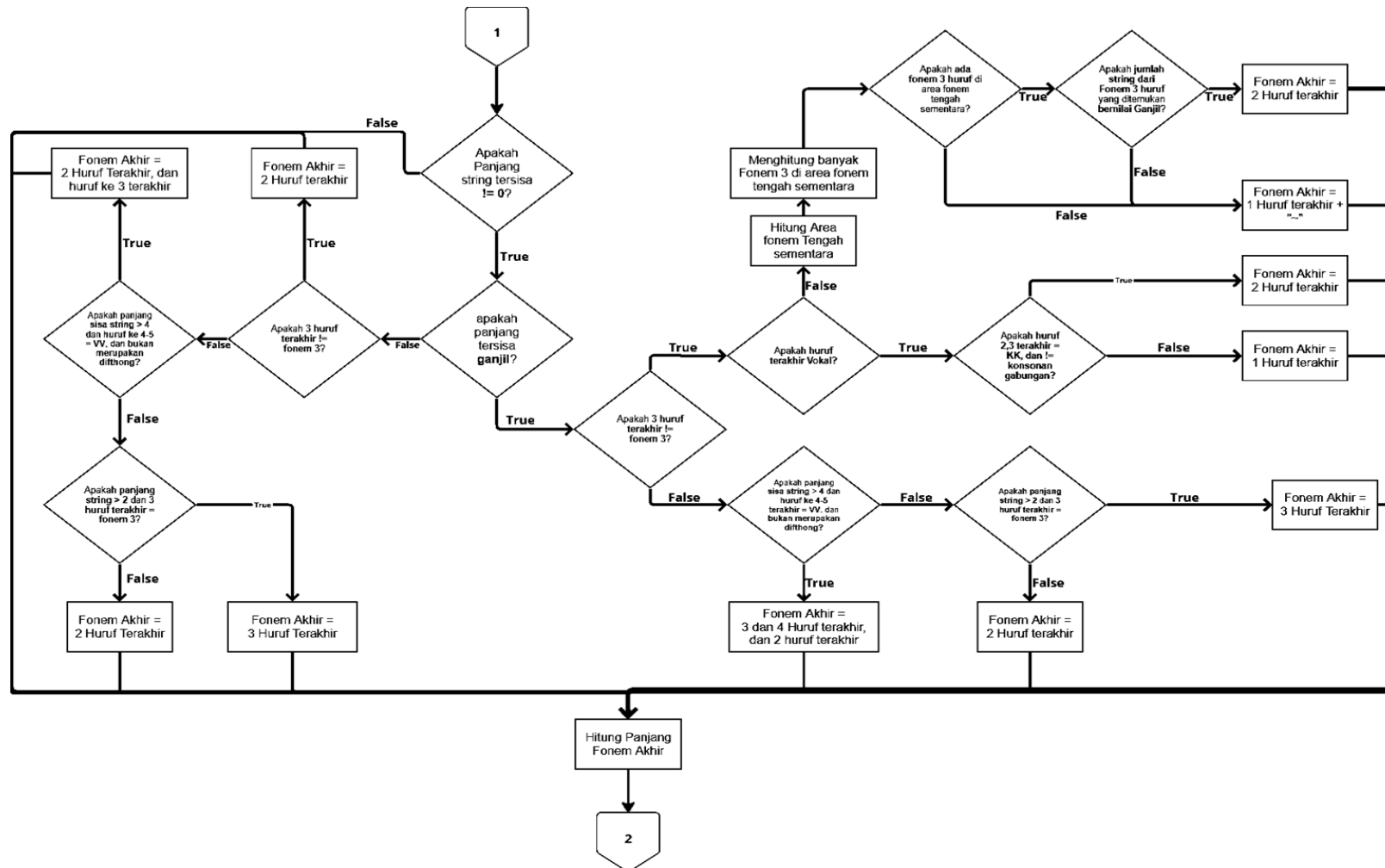
Selanjutnya, penelitian ini akan melakukan evaluasi terhadap aplikasi menggunakan uji *black box*. Selain itu, korpus yang telah dibuat juga akan melalui proses evaluasi melalui survei, yang kemudian akan dianalisis berdasarkan metode *Mean Opinion Score* (MOS). Setelah sistem pembuatan korpus ini divalidasi, diharapkan adanya korpus yang lebih berkualitas dapat membantu perkembangan *text-to-speech* bahasa Indonesia.



Gambar 9. Flowchart program fonem awal



Gambar 10. Flowchart program fonem tengah



Gambar 11. Flowchart program fonem akhir

3.5 Teknik Pengambilan Data

Pengambilan data suara dilakukan dengan cara merekam setiap fonem AA-ZZ, fonem 3 huruf dan fonem 1 huruf yang dibaca $\frac{1}{2}$ untuk memastikan tidak ada fonem yang hilang maka dibuat daftar pengucapan yang menyimpan fonem yang dibutuhkan ditunjukkan pada Gambar 12 dan lainnya pada berada pada Lampiran 2.

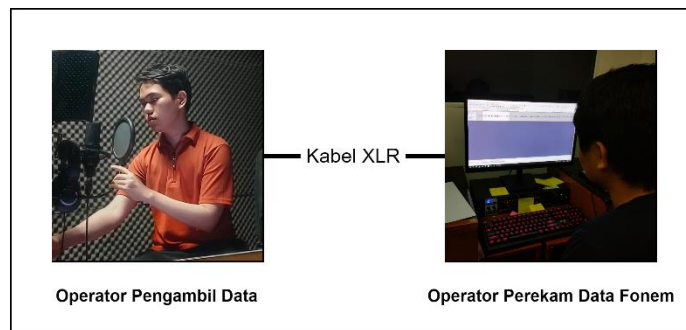
	A	B	C	D	E	F	G	H
A	A	Batu	Cacing	Dadu	ea	fatur	gaga	halik
B	Abu	B	Coba	Dbu	ebi	fbo	gbut	hbis
C	Acuan	Bcek	C	Dcak	eco	fca	gco	hco
D	Aduan	bduk	Cendol	D	edan	fda	gde	hdo
E	Ae	beli	Cemara	Deras	E	fe	gema	heran
F	Afrika	bf	Canggih	Dfa	Efa	F	gfa	hfa
G	Agen	Bg	Cegah	Dga	egar	fga	G	hgo
H	Ahli	bh	Chelsea	Dhan	eh	fham	ghi	H
I	Air	bilang	Cinta	Dirham	ei	fikri	gila	hitam
J	Ajar	bja	Cajuput	Dja	eja	fja	gjah	hjo
K	Akal	bkam	Cek	Dka	eka	fka	gko	hkim
L	Alam	bla	Caleg	Dli	eli	flor	gli	hlim

Gambar 12. Daftar pengucapan fonem

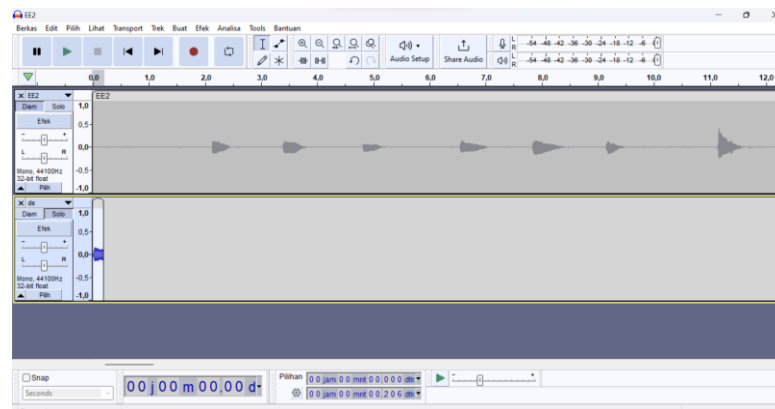
Setelah daftar pengucapan fonem dibuat, selanjutnya proses perekaman data audio, proses ini dilakukan dengan beberapa tahapan yaitu:

1. Lokasi pengambilan data dilakukan di dalam Laboratorium Kecerdasan Buatan Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.
2. Mengatur tata letak alat untuk mempermudah saat perekaman berlangsung.
3. Proses pengambilan data dilakukan oleh dua orang, satu orang bertugas untuk melakukan pengucapan fonem menggunakan *microphone* dan satu lagi bertugas sebagai operator untuk merekam.
4. Pengambilan data suara dilakukan dengan mengucapkan fonem satu persatu, setiap kata diulang sebanyak 3 kali.
5. Data suara yang telah direkam akan melalui proses pemotongan sinyal suara, sehingga sinyal yang tersisa hanyalah sinyal suara fonem yang dibutuhkan.
6. Data suara yang sudah bersih kemudian disimpan dalam format wav (*Waveform Audio File Format*).

Adapun ilustrasi proses pengambilan data dan contoh sample data suara yang telah diambil dapat dilihat pada Gambar 13 dan Gambar 14.



Gambar 13. Ilustrasi pengambilan data suara



Gambar 14. Ilustrasi contoh pengambilan data suara untuk fonem “E”

3.6 Analisis Kinerja Sistem

Analisis ini bertujuan untuk mengukur dan mengetahui kinerja sistem dari tingkat akurasi, berdasarkan dengan hasil perhitungan persentase jumlah kata yang dideteksi dengan benar. Perhitungan akurasi ditunjukkan pada persamaan (3) berikut:

$$\text{Akurasi} = \frac{\text{jumlah kata benar}}{\text{jumlah kata}} \times 100\% \quad (3)$$

Selain mengukur akurasi kinerja sistem, penggunaan metode evaluasi *Mean Opinion Score* (MOS) menjadi penting untuk mengukur kualitas suara yang dihasilkan oleh sistem. Sebagai yang dijabarkan dalam Bab 2, MOS dihitung menggunakan rumus persamaan (2). Dengan tingkat kriteria penilaian didefinisikan berdasarkan tiga karakteristik utama, yakni *Intelligibility* (Kemudahan Pemahaman), *Fluidity* (Kelancaran) dan *Naturalness* (Kekenyalan). Setiap karakteristik ini dinilai dalam rentang skor 1 hingga 5, kriteria penilaian yang dijelaskan dengan lebih detail pada Tabel 6 (*Intelligibility*), Tabel 7 (*Fluidity*) dan Tabel 8 (*Naturalness*). Dalam menentukan nilai akhir, setiap karakteristik dari

kriteria penilaian dihitung menggunakan metode MOS (*Mean Opinion Score*) yang dijabarkan melalui persamaan (2), adapun format kuesioner dapat ditunjukkan pada Lampiran 3

Tabel 6. Kriteria penilaian tingkat kejelasan (*Intelligibility*)

Nilai	Kriteria
1	Ucapan tidak dapat dipahami, perangkaian ucapan sangat tidak jelas
2	Ucapan tidak dapat dipahami, perangkaian ucapan kurang jelas
3	Ucapan dapat dipahami, perangkaian ucapan kurang jelas
4	Ucapan dapat dipahami, perangkaian ucapan jelas
5	Ucapan dapat dipahami, perangkaian ucapan sangat jelas

Tabel 7. Kriteria penilaian tingkat kelancaran (*Fluidity*)

Nilai	Kriteria
1	Pengucapan tidak lancar, transisi antar fonem sangat mengganggu
2	Pengucapan tidak lancar, transisi antar fonem mengganggu
3	Pengucapan cukup lancar, transisi antar fonem sedikit mengganggu
4	Pengucapan lancar, transisi antar fonem nyaman
5	Pengucapan lancar, transisi antar fonem sangat nyaman

Tabel 8. Kriteria penilaian tingkat kealamian (*Naturalness*)

Nilai	Kriteria
1	Pengucapan datar
2	Pengucapan sedikit berintonasi, tidak sesuai pengucapan manusia pada umumnya
3	Pengucapan sedikit berintonasi, sesuai pengucapan manusia pada umumnya
4	Pengucapan berintonasi baik, sesuai pengucapan manusia pada umumnya
5	Pengucapan identik dengan pengucapan manusia pada umumnya

BAB 4 HASIL DAN PEMBAHASAN

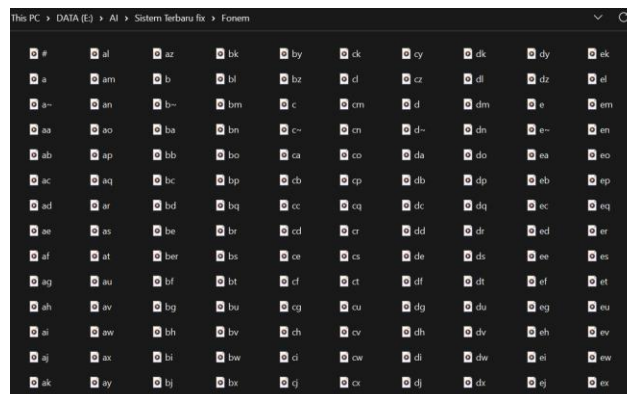
4.1 Hasil Penelitian

Penelitian ini dilakukan dengan cara membuat sistem dan menilai hasil keluaran sistem menggunakan 4 skenario berbeda, nilai akan dilihat berdasarkan 2 pengujian yaitu, Pengujian Fungsional dan Pengujian Kualitas Keluaran Sistem. Pengujian fungsional menggunakan metode *black box* sedangkan Pengujian Kualitas Keluaran Sistem dilakukan dengan melakukan survei untuk menilai tingkat kriteria kejelasan (*Intelligibility*), kelancaran (*Fluidity*), maupun kealamian ucapan (*Naturalness*) menggunakan perhitungan *Mean Opinion Score* (MOS). Adapun nilai ini akan dibandingkan dengan nilai keluaran sistem penelitian sebelumnya yaitu keluaran sistem yang menggunakan korpus *diphone contatenation*, yang di nilai dari pengujian yang sama.

4.2 Keluaran Sistem

1. Database Audio Fonem

Database audio fonem merupakan *database* audio yang menyimpan data suara setiap fonem yang dibutuhkan, data suara fonem yang telah dibuat disatukan dalam sebuah folder yang diberi nama "Fonem" dengan format wav, seperti yang terlihat pada Gambar 15. Jumlah total keseluruhan fonem yang ada berjumlah 749. Angka ini terdiri atas 26 fonem 1 huruf abjad, 26 fonem 1 huruf abjad yang dibaca ½, 20 fonem 3 huruf, 1 fonem spasi dan 676 fonem 2 huruf, adapun data audio dapat dilihat pada Lampiran 5.



Gambar 15. *Database* fonem

2. Preprocessing

Input berupa kalimat pada sistem akan melalui proses penyeragaman ke dalam huruf kecil, dihapus tanda bacanya proses ini menggunakan library regular expression python dan terakhir dipecah menjadi kata per kata. Contoh hasil proses ini dapat dilihat pada Tabel 9.

Tabel 9. Output pasca preprocessing

Sebelum	Sesudah
minum kopi	[minum],[kopi]
presiden jokowi	[presiden],[jokowi]
siapa yang pergi?	[siapa],[yang],[pergi]
Mau kemana?	[mau],[kemana]
lihat kesana!	[lihat],[kesana]

3. Kategorisasi Fonem

Kata kata yang sudah melewati preprocessing, selanjutnya akan melalui proses kategorisasi fonem untuk menentukan fonem awal, tengah dan akhir sesuai dengan aturan kaidah linguistik bahasa Indonesia yang telah disiapkan sebelumnya, contoh keluaran dapat dilihat pada Tabel 10.

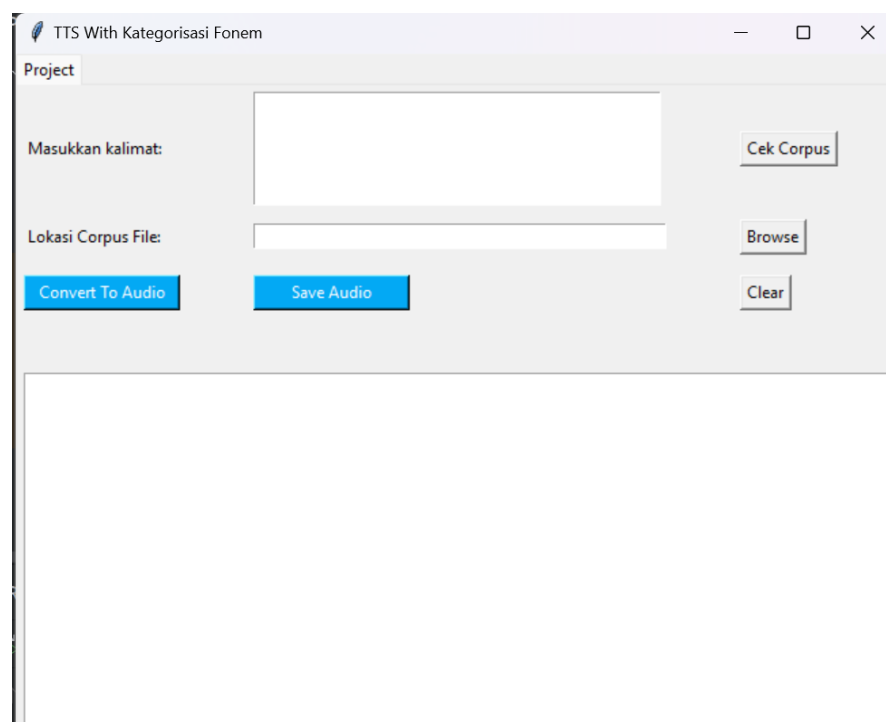
Tabel 10. Output pasca kategorisasi fonem

Kata	Fonem Awal	Fonem Tengah	Fonem Akhir	Korpus
yang	ya	-	ng	ya+ng
dibuat	di	bu	at	di+bu+at
gula	gu	-	la	gu+la
membersihkan	me+m~	ber+si+h~	kan	me+m~+ber+si+h~+kan
ibu	I	-	bu	i+bu
banjir	ba+n~	ji	r~	ba+n~+ji+r~
ayah	ay	-	ah	ay+ah
teras	ter	-	as	ter+as
tetangga	te	ta+ng	ga	te+ta+ng+ga
gurame	gu	ra	me	gu+ra+me
misal	mi	sa	l~	mi+sa+l~
ember	em	-	ber	em+ber

Kata	Fonem Awal	Fonem Tengah	Fonem Akhir	Korpus
tampan	ta	m~	pan	ta+m~+pan
pemilu	pem	il	u	pem+il+u
memenangkan	me	men+an+g~	kan	me+men+an+g~+kan
jokowi	jo	ko	wi	jo+ko+wi
merdeka	me+r~	de	ka	me+r~+de+ka

4. User Interface System

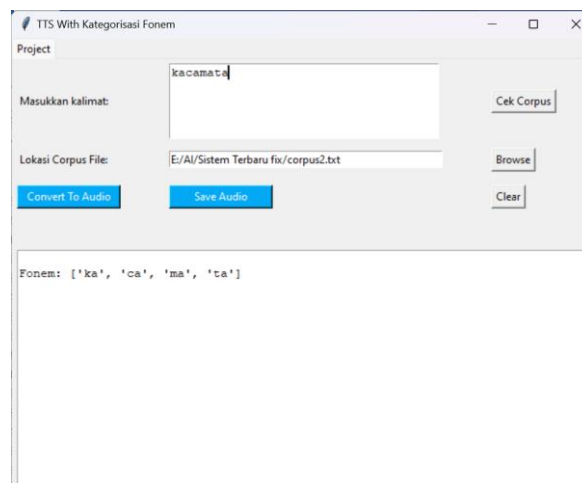
Ketika sistem ini dijalankan, akan ditampilkan antarmuka pengguna (*User Interface System*) yang berfungsi untuk menampilkan *form* yang digunakan sebagai tampilan awal program *text-to-speech*. Antarmuka pengguna sistem (*User Interface System*) ditunjukkan pada Gambar 16.



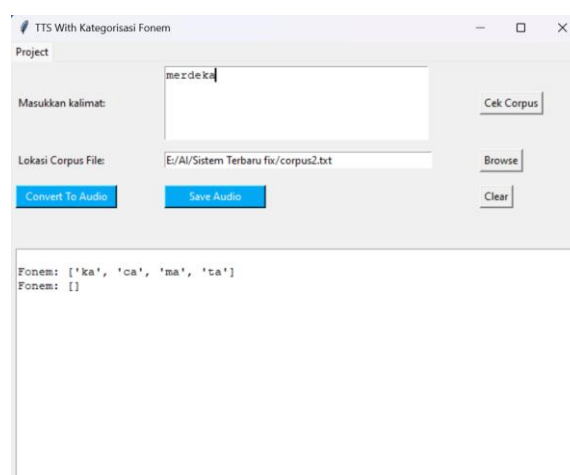
Gambar 16. *User interface system*

Pada *User Interface System* di atas, terdapat *text box* yang dapat digunakan untuk memasukan kata-kata yang ingin di dengar dan *text box* kedua yang menggunakan tombol '*browse*' untuk memilih korpus yang di inginkan. Ketika kata dimasukan dalam *text box* dan menekan tombol '*convert to audio*' maka kata akan langsung di konversi menjadi suara, Gambar 17 menunjukan tampilan pada sistem saat didengarkan. Sedangkan Gambar 18 , menampilkan

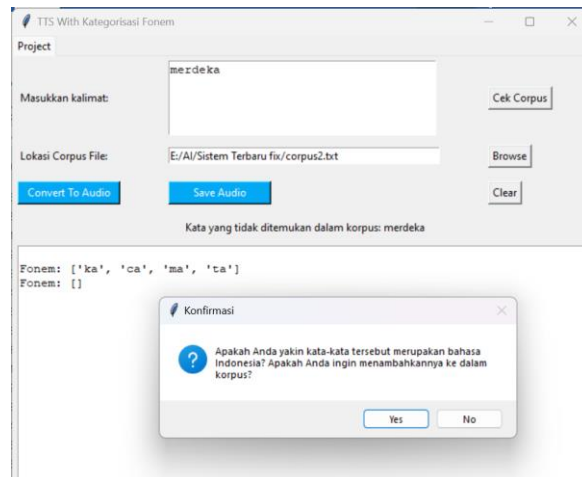
apabila kata tidak ada dalam *database* korpus sehingga tidak berhasil di konversi. Namun hal itu dapat di atasi dengan menekan tombol ‘cek korpus’ dimana akan dilakukan pengecekan setiap kata dalam *text box* menggunakan algoritma *Knuth-morris-pratt* (KMP). apabila ada kata yang tidak terdapat dalam korpus akan muncul notifikasi konfirmasi untuk memastikan kata tersebut benar bahasa Indonesia seperti pada Gambar 19 dan apabila menekan tombol ‘yes’ maka kata-kata tersebut akan masuk dalam program kategorisasi fonem untuk ditambahkan dalam korpus fonem, sehingga dapat melakukan konversi seperti yang ditunjukkan pada Gambar 20, selain itu terdapat tombol ‘clear’ yang digunakan untuk menghapus fonem dari suara yang dihasilkan yang ditunjukkan pada *text area* pada Gambar 21.



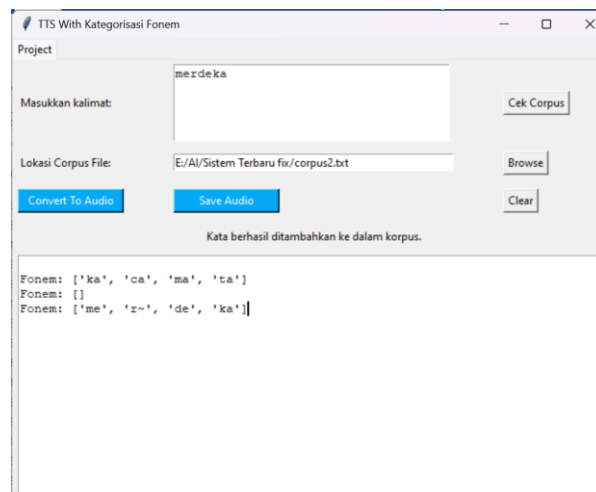
Gambar 17. Kata yang berhasil dikonversi



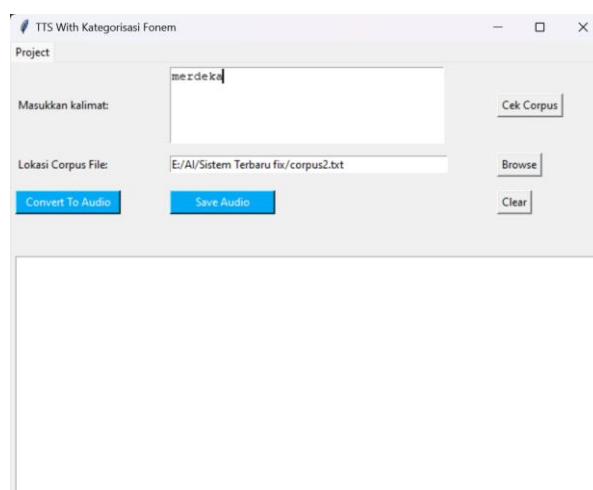
Gambar 18. Kata yang tidak berhasil dikonversi



Gambar 19. Notifikasi konfirmasi



Gambar 20. Kata berhasil ditambahkan



Gambar 21. Text berhasil dibersihkan

Adapun fitur lainnya adalah adanya tampilan keluaran berupa potongan-potongan fonem yang telah dimasukkan dan dapat diterima oleh pengguna saat sistem sedang berjalan. Selain itu, terdapat tombol '*save audio*' yang akan menyimpan audio dari kata yang dimasukkan pengguna dalam folder '*output*' dengan format .wav (*Waveform Audio File Format*). Setelah sistem dijalankan, dilakukan pengujian untuk mengevaluasi kinerja sistem, yang bertujuan untuk mengetahui apakah keluaran sistem telah berfungsi dengan baik atau tidak. Pengujian ini dilakukan dengan menilai keluaran suara berdasarkan beberapa kriteria dengan memberikan skala *likert* 1-5 untuk tingkat pemahaman, tingkat kelancaran dan kealamian ucapan.

4.3 Pengujian Sistem

Pengujian dilakukan bertujuan untuk menemukan kekurangan pada sistem yang telah dibuat. Pengujian dilakukan dengan dua tahap yaitu tahap pertama adalah pengujian fungsional dan selanjutnya pengujian kualitas keluaran suara.

1. Pengujian Fungsional

Pengujian fungsional merupakan metode uji sistem tanpa mengetahui struktur internal kode atau program. Dalam pengujian ini dilakukan untuk menguji setiap fitur yang ada pada sistem yang dibuat. Pengujian dilakukan secara *Black box* yang didasarkan pada detail seperti tampilan sistem, tombol-tombol dan fungsi-fungsi yang ada pada sistem. Pengujian secara fungsional yang dilakukan pada sistem dapat dilihat pada Tabel 11 berikut ini.

Tabel 11. Pengujian fungsional sistem

No.	Fitur yang di uji	Detail pengujian	Jenis uji
1	<i>User Interface</i>	Menampilkan semua tampilan yang ada pada sistem	<i>Black box</i>
2	<i>Convert Audio</i>	Mengubah teks yang telah dimasukkan ke dalam sistem menjadi suara	<i>Black box</i>
3	Kategorisasi Fonem	Menambahkan teks yang tidak ada dalam korpus, kedalam korpus	<i>Black box</i>
4	Korpus	Memilih korpus yang ada pada	<i>Black box</i>

No.	Fitur yang di uji	Detail pengujian	Jenis uji
5	<i>Save audio</i>	Menyimpan audio <i>text</i> yang di masukan kedalam sistem	<i>Black box</i>
6	<i>Clear</i>	Membersihkan <i>area text</i> yang menyimpan bentuk fonem dari suara yang dihasilkan	<i>Black box</i>

Berdasarkan pengujian fungsional sistem pada Tabel 9 maka dapat dilakukan pengujian pada sistem yang dibuat. Hasil dari pengujian fungsional pada sistem ditunjukkan pada Tabel 12 berikut.

Tabel 12. Hasil pengujian fungsional sistem

No.	Kasus dan Hasil Uji			
	Fitur yang di uji	Aksi / Data Masukan	Pengamatan	Kesimpulan
1	<i>User Interface</i>	Menampilkan semua tampilan yang ada pada sistem	Sistem menampilkan semua tampilan yang ada	[√] Diterima [] Ditolak
2	<i>Convert Audio</i>	Mengubah teks yang telah dimasukkan ke dalam sistem menjadi suara	Sistem dapat mengkonversi teks yang dimasukkan menjadi suara	[√] Diterima [] Ditolak
3	Kategorisasi Fonem	Menambahkan teks yang tidak ada dalam korpus, kedalam korpus	Sistem dapat menambahkan teks yang tidak ada dalam korpus	[√] Diterima [] Ditolak
4	Korpus	Memilih korpus yang berbeda	Sistem dapat memilih korpus yang berbeda	[√] Diterima [] Ditolak
5	<i>Save audio</i>	Menyimpan audio dalam format wav dari <i>text</i> yang di masukan kedalam sistem	Sistem dapat menyimpan audio dari teks yang dimasukan kedalam format WAV	[√] Diterima [] Ditolak
6	<i>Clear</i>	Membersihkan <i>area text</i> yang menyimpan bentuk fonem dari suara yang dihasilkan	Sistem dapat membersihkan area text yang menyimpan bentuk fonem dari suara yang dihasilkan	[√] Diterima [] Ditolak

Berdasarkan Hasil pengujian sistem pada Tabel 12, dapat disimpulkan bahwa secara fungsional seluruh proses pada sistem yang dibuat telah berjalan dengan baik sesuai dengan yang diharapkan.

2. Pengujian Kualitas Keluaran Sistem

Dalam pengujian ini, Sistem *text-to-speech* yang telah dibuat akan dinilai berdasarkan kinerjanya dengan cara menguji kinerjanya menggunakan beberapa data kata yang ada pada *database* korpus bahasa Indonesia yang telah disiapkan sebelumnya. Pengujian kualitas keluaran sistem ini terbagi atas dua bagian, yaitu menghitung akurasi sistem dalam melakukan konversi teks ke suara menggunakan persamaan (3) dan melakukan pengujian kinerja sistem berdasarkan kriteria penilaian pada tingkat kejelasan (*Intelligibility*), kelancaran (*Fluidity*), maupun kealamian ucapan (*Naturalness*) menggunakan perhitungan *Mean Opinion Score* (MOS) dengan persamaan (2). Pengujian ini melibatkan 101 responden dengan mendengarkan hasil rekaman konversi teks ke suara yang menjadi data uji, selanjutnya responden akan mengisi form online yang ditunjukkan pada Lampiran 3, adapun responden akan menilai dengan bobot 1-5 berdasarkan kriteria penilaian yang terdapat dalam Tabel 6, Tabel 7 dan Tabel 8. Sehingga dapat diketahui nilai keluaran suara berdasarkan tingkat *Intelligibility*, *Fluidity* dan *Naturalness* pada sistem yang telah di uji dimana nilai terendah 1 dan nilai tertinggi adalah 5. Selain itu dilakukan juga pengujian pada keluaran penelitian sebelumnya yang menggunakan korpus *diphone contatenation* dengan data uji yang sama demi mendapatkan perbandingan diantara korpus kategorisasi fonem dan juga korpus *diphone contatenation*.

Ketiga kriteria penilaian tersebut diuji dengan empat skenario yang berbeda. Skenario 1 terdiri dari satu kata dengan panjang 3, 4, 5 dan 6 huruf. Skenario 2 berisi kalimat yang terdiri dari 2 dan 3 kata. Skenario 3 berisi kata-kata dengan imbuhan, sementara skenario keempat berisi kalimat berfrasa dengan susunan 8 kata. Hasil pengujian ini menjadi dasar kesimpulan pada sistem yang telah dibuat, dengan membandingkan hasil dari masing-masing skenario yang telah dilakukan. Hal ini dilakukan untuk mengetahui pengaruh

panjang kata dan kata-kata yang digunakan pada setiap skenario serta hasil keluaran pada tingkat keberhasilan korpus terintegrasi kategorisasi fonem dalam menghasilkan keluaran suara dengan memperhatikan nilai *Intelligibility*, *Fluidity* dan *Naturalness*.

a. Skenario pertama

Pada pengujian ini, keluaran sistem berupa satu kata dengan panjang huruf yang berbeda yaitu 3, 4, 5 dan 6 huruf. Tiap huruf memiliki 5 kata yang berbeda. Tabel 13 menunjukan hasil keluaran dari skenario pertama.

Tabel 13. Hasil pengujian skenario 1 kaegorisasi 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	Iya	1	100	3,84	3,80	3,52	43
2	Ini	1	100	3,59	3,73	3,33	45
3	Dia	1	100	3,50	3,58	3,29	45
4	Abu	1	100	3,65	3,73	3,46	46
5	Air	1	100	3,52	3,55	3,18	45
6	Kamu	1	100	3,60	3,59	3,54	43
7	Saya	1	100	3,61	3,52	3,54	48
8	Ikan	1	100	3,81	3,79	3,70	52
9	Mama	1	100	3,65	3,53	3,37	41
10	Lima	1	100	3,68	3,50	3,45	42
11	Makan	1	100	3,94	4,02	3,78	53
12	Siapa	1	100	3,37	3,44	3,43	71
13	Kapan	1	100	3,69	3,69	3,64	52
14	Lapis	1	100	3,45	3,56	3,42	82
15	Rumah	1	100	3,62	3,64	3,50	76
16	Kucing	1	100	3,30	3,35	3,29	75
17	Anjing	1	100	3,22	3,28	3,29	79
18	Kancil	1	100	3,37	3,47	3,42	93
19	Sayang	1	100	3,25	3,36	3,23	76
20	Negeri	1	100	3,63	3,73	3,57	76
Total/Rata-Rata		20	100	3,57	3,59	3,45	1183

Berdasarkan Tabel 13, hasil uji akurasi mendapat nilai 100% yang menunjukkan semua kata berhasil di konversi menjadi suara. Adapun rentang nilai untuk setiap huruf pada skenario ini dapat dilihat pada Tabel 14.

Tabel 14. Rentang nilai skenario 1 kategorisasi fonem

Skenario 1		<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>
3 Huruf	MIN	3,50	3,55	3,18
	MAX	3,84	3,80	3,52
4 Huruf	MIN	3,60	3,50	3,37
	MAX	3,81	3,79	3,70
5 Huruf	MIN	3,37	3,44	3,42
	MAX	3,94	4,02	3,78
6 Huruf	MIN	3,22	3,28	3,23
	MAX	3,63	3,73	3,57

Dilihat pada Tabel 14, pada skenario ini terlihat nilai tertinggi pada setiap tingkatan diperoleh dari kata dengan 5 huruf, disusul kata dengan 4, 3, dan terakhir 6 huruf. Meskipun demikian hasil ini dianggap cukup baik jika dilihat hasil akhir nilai MOS skenario pertama, yaitu 3,57 untuk *Intelligibility*, 3,59 untuk *Fluidity* dan 3,45 untuk *Naturalness*. Disamping itu keseluruhan 20 data uji audio ini hanya membutuhkan file data penyimpanan sebesar 1183 Kb.

b. Skenario kedua

Pada pengujian ini, keluaran sistem berupa dua dan tiga kata, dengan masing masing 5 data uji. Tabel 15 menunjukan hasil keluaran skenario dua.

Tabel 15. Hasil pengujian skenario 2 kaegorisasi fonem

No	Data Uji	Jumlah Kata Terkonversi	Akuratsi (100%)	Kategorisasi Fonem				Size (KB)
				<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>		
1	Makan Malam	2	100	3,68	3,58	3,42		160
2	Jalan Sekarang	2	100	3,46	3,40	3,27		210

No	Data Uji	Jumlah Kata Terkonversi	Akuratsi (100%)	Kategorisasi Fonem			
				<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>	<i>Size (KB)</i>
3	Minum Kopi	2	100	3,74	3,61	3,56	162
4	Masuk Sekolah	2	100	3,62	3,58	3,38	188
5	Belok Kanan	2	100	3,47	3,48	3,29	148
6	Kopi Ini Manis	3	100	3,67	3,67	3,52	227
7	Kepala Saya Sakit	3	100	3,54	3,51	3,36	230
8	Saya Mau Makan	3	100	3,65	3,58	3,50	192
9	Saya Sedang Masak	3	100	3,32	3,42	3,18	241
10	Ini Kamar Saya	3	100	3,54	3,61	3,30	204
Total/Rata-Rata		25	100	3,57	3,55	3,38	1962

Berdasarkan Tabel 15, hasil uji akurasi mendapat nilai 100% yang menunjukkan semua kata berhasil di konversi menjadi suara. Adapun rentang nilai untuk setiap huruf pada skenario ini dapat dilihat pada Tabel 16.

Tabel 16. Rentang nilai skenario 2 kategorisasi fonem

Skenario 2		<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>
2 Kata	MIN	3,46	3,40	3,27
	MAX	3,74	3,61	3,56
3 Kata	MIN	3,32	3,42	3,18
	MAX	3,67	3,67	3,52

Dilihat pada Tabel 16, pada skenario ini terlihat nilai tertinggi pada setiap tingkatan diperoleh oleh kalimat dengan 2 kata dilanjutkan dengan 3 kata. Hasil ini juga dianggap cukup baik jika dilihat hasil akhir nilai MOS

skenario kedua, yaitu 3,57 untuk *Intelligibility*, 3,55 untuk *Fluidity* dan 3,38 untuk *Naturalness*. Disamping itu keseluruhan 10 data uji audio ini hanya membutuhkan file data penyimpanan sebesar 1962 Kb.

c. Skenario Ketiga

Pada pengujian ini, keluaran sistem berupa 5 kata berimbuhan, akan dilakukan pengujian sama seperti sebelumnya, hasil pengujian dapat dilihat pada Tabel 17

Tabel 17. Hasil pengujian 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	1	100	3,91	3,96	3,63	84
2	Menjaga	1	100	3,79	3,87	3,56	81
3	Menikah	1	100	3,45	3,50	3,34	81
4	Bersalaman	1	100	3,67	3,75	3,53	149
5	Berikan	1	100	3,55	3,47	3,40	91
Total/Rata-Rata		5	100	3,68	3,71	3,49	486

Berdasarkan Tabel 17, hasil uji akurasi mendapat nilai 100% yang menunjukkan semua kata berhasil di konversi menjadi suara. Adapun rentang nilai untuk setiap huruf pada skenario ini dapat dilihat pada Tabel 18.

Tabel 18. Rentang nilai skenario 3 kategorisasi fonem

Skenario 3		<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>
Kata Berimbuhan	MIN	3,45	3,47	3,34
	MAX	3,91	3,96	3,63

Dilihat pada Tabel 18, pada skenario ini terlihat nilai tertinggi untuk 3,91 *Intelligibility*, 3,96 *Fluidity*, dan 3,63 *Naturalness*, sedangkan nilai terendah secara berturut-turut adalah 3,45, 3,47, dan 3,34. Hasil ini

dianggap cukup baik jika dilihat hasil akhir nilai MOS skenario ketiga, yaitu 3,68 untuk *Intelligibility*, 3,71 untuk *Fluidity* dan 3,49 untuk *Naturalness*. Disamping itu keseluruhan 5 data uji audio ini hanya membutuhkan file data penyimpanan sebesar 486 Kb.

d. Skenario Keempat

Pada pengujian ini, keluaran sistem berupa 5 kalimat berfrasa, akan dilakukan pengujian sama seperti sebelumnya, hasil pengujian dapat dilihat pada Tabel 19.

Tabel 19. Hasil pengujian skenario 4 kaegorisasi 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	Saya pergi ke sekolah untuk belajar dan bermain	8	100	3,45	3,51	3,45	689
2	Hari ini sangat panas karena telah musim kemarau	8	100	3,42	3,55	3,30	711
3	Kopi yang dibuat sangat manis karena diberi gula banyak	9	100	3,20	3,50	3,27	802
4	ayah dan ibu membersihkan teras rumah karena banjir	8	100	3,38	3,42	3,28	770
5	ibu sedang makan ikan gurame yang diberi tetangga	8	100	3,35	3,37	3,27	699
Total/Rata-Rata		41	100	3,36	3,47	3,31	3671

Berdasarkan Tabel 19, hasil uji akurasi mendapat nilai 100% yang menunjukkan semua kata berhasil di konversi menjadi suara. Adapun rentang nilai untuk setiap huruf pada skenario ini dapat dilihat pada Tabel 20.

Tabel 20. Rentang nilai skenario 4 kategorisasi fonem

Skenario 4		<i>Intelligibility Fluidity Naturalness</i>		
Kalimat Berfrasa	MIN	3,20	3,37	3,27
	MAX	3,45	3,55	3,45

Dilihat pada Tabel 20, pada skenario ini terlihat nilai tertinggi untuk 3,45 *Intelligibility*, 3,55 *Fluidity*, dan 3,45 *Naturalness*, sedangkan nilai terendah secara berturut-turut adalah 3,20, 3,37, dan 3,27. Hasil ini dianggap cukup baik jika dilihat hasil akhir nilai MOS skenario ketiga, yaitu 3,36 untuk *Intelligibility*, 3,47 untuk *Fluidity* dan 3,31 untuk *Naturalness*. Disamping itu keseluruhan 5 data uji audio dengan 49 kata ini membutuhkan file data penyimpanan sebesar 3671 Kb.

Berdasarkan hasil penelitian yang ditunjukkan pada skenario 1 hingga skenario 4, tingkat keberhasilan sistem dalam mengkonversi teks menjadi suara berhasil mencapai akurasi 100%, hal ini menunjukkan bahwa sistem pembuatan korpus yang terintegrasikan kategorisasi fonem berjalan dengan baik sesuai dengan yang diharapkan.

Adapun hasil keseluruhan pengujian kategorisasi fonem di atas akan di bandingkan dengan keluaran sistem *text-to-speech* yang menggunakan korpus *Diphone Concatenation*. Format pengujian yang diberikan sama terkecuali pada bagian akurasi karena sistem ini sebelumnya tidak memiliki penambahan korpus secara otomatis sehingga format survei seperti yang terlampir pada Lampiran 3. Perbandingan ini dapat dilihat pada Tabel 21, disamping itu data setiap responden pada pengujian ini ditunjukkan pada Lampiran 4.

Tabel 21. Perbandingan hasil pengujian kategorisasi fonem dan *diphone concatenation*

No	Data Uji	<i>Diphone Concatenation</i>				Kategorisasi Fonem			
		<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>	<i>Size (KB)</i>	<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>	<i>Size (KB)</i>
1	Iya	3,22	3,28	3,04	102	3,84	3,80	3,52	43
2	Ini	3,02	3,09	2,97	104	3,59	3,73	3,33	45
3	Dia	2,98	3,13	2,87	101	3,50	3,58	3,29	45
4	Abu	3,18	3,10	2,94	100	3,65	3,73	3,46	46
5	Air	2,97	2,87	2,83	107	3,52	3,55	3,18	45
6	Kamu	2,94	3,09	2,95	125	3,60	3,59	3,54	43
7	Saya	2,93	2,99	2,93	134	3,61	3,52	3,54	48
8	Ikan	2,85	3,01	2,81	126	3,81	3,79	3,70	52
9	Mama	2,89	2,92	2,82	124	3,65	3,53	3,37	41
10	Lima	3,11	2,98	2,93	126	3,68	3,50	3,45	42
11	Makan	3,11	3,10	3,10	153	3,94	4,02	3,78	53
12	Siapa	2,85	2,86	2,81	155	3,37	3,44	3,43	71
13	Kapan	2,92	2,94	2,90	150	3,69	3,69	3,64	52
14	Lapis	2,96	2,90	2,88	161	3,45	3,56	3,42	82
15	Rumah	2,93	3,09	2,97	164	3,62	3,64	3,50	76

No	Data Uji	<i>Diphone Concatenation</i>				Kategorisasi Fonem			
		<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>	<i>Size (KB)</i>	<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>	<i>Size (KB)</i>
16	Kucing	2,61	2,68	2,63	188	3,30	3,35	3,29	75
17	Anjing	2,62	2,53	2,64	196	3,22	3,28	3,29	79
18	Kancil	2,68	2,69	2,63	194	3,37	3,47	3,42	93
19	Sayang	2,63	2,63	2,63	194	3,25	3,36	3,23	76
20	Negeri	2,75	2,77	2,77	187	3,63	3,73	3,57	76
21	Makan Malam	2,98	2,98	2,86	321	3,68	3,58	3,42	160
22	Jalan Sekarang	2,70	2,79	2,70	420	3,46	3,40	3,27	210
23	Minum Kopi	2,90	2,91	2,74	309	3,74	3,61	3,56	162
24	Masuk Sekolah	2,84	2,76	2,73	400	3,62	3,58	3,38	188
25	Belok Kanan	2,96	2,84	2,86	331	3,47	3,48	3,29	148
26	Kopi Ini Manis	2,99	3,00	2,91	359	3,67	3,67	3,52	227
27	Kepala Saya Sakit	2,68	2,77	2,84	492	3,54	3,51	3,36	230
28	Saya Mau Makan	2,83	2,89	2,77	409	3,65	3,58	3,50	192
29	Saya Sedang Masak	2,67	2,67	2,76	508	3,32	3,42	3,18	241
30	Ini Kamar Saya	2,79	2,87	2,76	359	3,54	3,61	3,30	204
31	Bermain	3,06	3,12	3,05	218	3,91	3,96	3,63	84
32	Menjaga	2,96	3,03	2,89	212	3,79	3,87	3,56	81

No	Data Uji	<i>Diphone Concatenation</i>				Kategorisasi Fonem			
		<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>	<i>Size (KB)</i>	<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>	<i>Size (KB)</i>
33	Menikah	2,88	2,95	2,83	210	3,45	3,50	3,34	81
34	Bersalaman	2,92	3,11	2,88	304	3,67	3,75	3,53	149
35	Berikan	2,95	3,08	2,85	216	3,55	3,47	3,40	91
36	Saya pergi ke sekolah untuk belajar dan bermain	2,84	2,75	2,73	1361	3,45	3,51	3,45	689
37	Hari ini sangat panas karena telah musim kemarau	2,69	2,68	2,57	1243	3,42	3,55	3,30	711
38	Kopi yang dibuat sangat manis karena diberi gula banyak	2,71	2,62	2,63	1605	3,20	3,50	3,27	802
39	ayah dan ibu membersihkan teras rumah karena banjir	2,76	2,64	2,63	1486	3,38	3,42	3,28	770
40	ibu sedang makan ikan gurame yang diberi tetangga	2,64	2,65	2,54	1413	3,35	3,37	3,27	699
Total/Rata-rata		2,87	2,90	2,82	15067	3,55	3,58	3,42	7302

3. Pembahasan

Tabel 22. Ringkasan hasil penelitian

Skenario	Bagian	<i>Diphone Concatenation</i>				Kategorisasi Fonem			
		<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>	<i>Size (Kb)</i>	<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>	<i>Size (Kb)</i>
1	Bagian A	3,07	3,09	2,93	514	3,62	3,68	3,35	224
	Bagian B	2,94	3,00	2,89	635	3,67	3,59	3,52	226
	Bagian C	2,95	2,98	2,93	783	3,61	3,67	3,55	334
	Bagian D	2,66	2,66	2,66	936	3,35	3,44	3,36	399
2	Bagian A	2,88	2,86	2,78	1781	3,59	3,53	3,38	868
	Bagian B	2,79	2,84	2,81	2127	3,55	3,56	3,37	1094
3	Bagian A	2,95	3,06	2,90	1160	3,68	3,71	3,49	486
4	Bagian A	2,73	2,67	2,62	7108	3,36	3,47	3,31	3671
Total	Keseluruhan	2,87	2,90	2,82	15044	3,55	3,58	3,42	7302

Apabila ditinjau berdasarkan kriteria penilaian *Mean Opinion Score*, Tabel 22 yang berisi ringkasan hasil penelitian dari Tabel 21 menunjukkan nilai-nilai dari tingkat *Intelligibility*, *Fluidity* dan *Naturalness* bahwa korpus yang terintegrasi kategorisasi fonem memiliki nilai yang lebih tinggi pada ketiga tingkatan jika dibandingkan dengan korpus *Diphone Concatenation* dengan perbedaan nilai yang cukup jauh pada ke empat skenario. Hal ini dapat dilihat dari hasil penilaian yang dilakukan oleh 101 responden, korpus yang terintegrasi kategorisasi fonem memiliki tingkat *Intelligibility* pada skenario 1 hingga skenario 4 rentang nilai yang diberikan responden adalah 3,35-3,68 dengan nilai akhir sebesar 3,55, Sehingga dapat dikatakan bahwa hasil konversi teks ke suara yang dihasilkan dapat dipahami dan perangkaian ucapannya cukup jelas. selanjutnya tingkat *Fluidity* memiliki penilaian untuk skenario 1 hingga skenario 4 dengan rentang 3,44-3,71 dengan nilai akhir 3,58 menunjukkan bahwa hasil konversi teks ke suara yang dihasilkan lancar dan transisi antar fonem nyaman, terakhir tingkat *Naturalness* memiliki nilai yang sedikit lebih rendah dari tingkat lainnya yaitu nilai dengan rentang 3,31-3,55 dengan nilai akhir 3,42, hal ini menunjukkan bahwa pengucapan memiliki tingkat intonasi yang cukup baik, yang sesuai dengan pengucapan manusia pada umumnya.

Sejalan dengan itu, hasil penelitian berhasil menunjukkan adanya peningkatan yang cukup signifikan dibandingkan dengan penelitian sebelumnya. Dengan metode pengujian yang sama, penelitian sebelumnya sistem *text-to-speech* yang menggunakan korpus *Diphone Concatenation* memperoleh nilai akhir 2,87 untuk tingkat *Intelligibility*, 2,9 untuk *Fluidity* dan 2,82 untuk *Naturalness*. Dengan demikian, peningkatan nilai pada ketiga aspek tersebut menunjukkan kemajuan yang positif dalam pengembangan sistem, menegaskan keberhasilan implementasi kategorisasi fonem pada korpus dalam peningkatan kualitas suara sintesis. Disamping itu, Korpus terintegrasi kategorisasi fonem menunjukkan perbedaan ukuran file yang cukup besar, di mana Kategorisasi fonem memiliki ukuran file yang lebih kecil berdasarkan skenario 1 hingga skenario 4 Kategorisasi Fonem memiliki total ukuran file sebesar 7302Kb sedangkan *Diphone Concatenation* memiliki total ukuran sebesar 15044 Kb, sehingga menunjukan Kategorisasi Fonem lebih ringan 48% dibandingkan dengan *Diphone Concatenation*.

BAB 5 KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut:

1. Kategorisasi fonem merupakan sistem untuk mengkategorikan fonem, dalam penelitian ini fonem dibagi menjadi 3 kategori, yaitu fonem 1 huruf, fonem 2 huruf, dan fonem 3 huruf.
2. Sistem dengan algoritma *Knuth-morris-pratt* serta korpus terintegrasi kategorisasi fonem yang dibuat mampu melakukan konversi kata-kata baru yang tidak terdapat dalam korpus menjadi fonem dan tersimpan dalam *database* fonem, sehingga kedepannya sistem dapat menyediakan semua keperluan korpus fonem bahasa Indonesia yang digunakan pada sistem *text-to-speech* dengan lebih cepat.
3. Korpus terintegrasi kategorisasi fonem yang dibuat ditinjau dalam 2 bagian yaitu, dengan mengukur akurasi keberhasilan sistem konversi teks menjadi suara dan menguji kinerja sistem menggunakan *mean opinion score* (MOS) berdasarkan kriteria penilaian pada tingkat kejelasan (*Intelligibility*), kelancaran (*Fluidity*), maupun kealamian ucapan (*Naturalness*). Hal ini ditinjau menggunakan 4 skenario, dimana keseluruhan skenario memiliki tingkat akurasi 100%, selain itu hasil uji MOS memiliki tingkat penilaian responden yang tidak terlalu berbeda pada ke 4 skenario, berdasarkan tingkat kejelasan menunjukkan nilai akhir dari skenario 1 hingga skenario 4 sebesar 3,55, tingkat kelancaran memiliki nilai akhir 3,58 dan tingkat kealamian memperoleh nilai akhir 3,42. sehingga sistem *text-to-speech* dengan menggunakan korpus terintegrasi kategorisasi fonem yang dibuat hasil konversi suaranya dapat dipahami dan perangkaian ucapannya cukup jelas, pengucapan yang dihasilkan lancar, transisi antar fonem nyaman, serta memiliki tingkat intonasi yang cukup baik, yang sesuai dengan pengucapan manusia pada umumnya.
4. Keluaran sistem *text-to-speech* yang menggunakan korpus terintegrasi kategorisasi fonem dibandingkan dengan keluaran sistem *text-to-speech* yang

menggunakan korpus *Diphone Concatenation* menghasilkan perbandingan yang cukup signifikan, berdasarkan tingkat kejelasan (*Intelligibility*), kelancaran (*Fluidity*), maupun kealamian ucapan (*Naturalness*) dan ukuran (*Size*). menggunakan MOS korpus terintegrasi kategorisasi fonem memiliki tingkat kejelasan 3,55, tingkat kelancaran 3,58 dan tingkat kealamian 3,42, sedangkan korpus *Diphone Concatenation* memiliki tingkat kejelasan 2,87, tingkat kelancaran 2,9 dan tingkat kealamian 2,82. Hal ini menunjukkan keluaran sistem *text-to-speech* yang menggunakan korpus terintegrasi kategorisasi fonem memiliki keluaran yang lebih baik dibandingkan dengan sistem *text-to-speech* yang menggunakan korpus *Diphone Concatenation*. Disamping itu keluaran korpus terintegrasi kategorisasi fonem memiliki ukuran file yang lebih ringan 48% dibandingkan dengan keluaran sistem *text-to-speech* yang menggunakan korpus *Diphone Concatenation*, hal ini berdasarkan *size* dari 40 data uji yang sama dari ke 4 skenario.

5.2 Saran

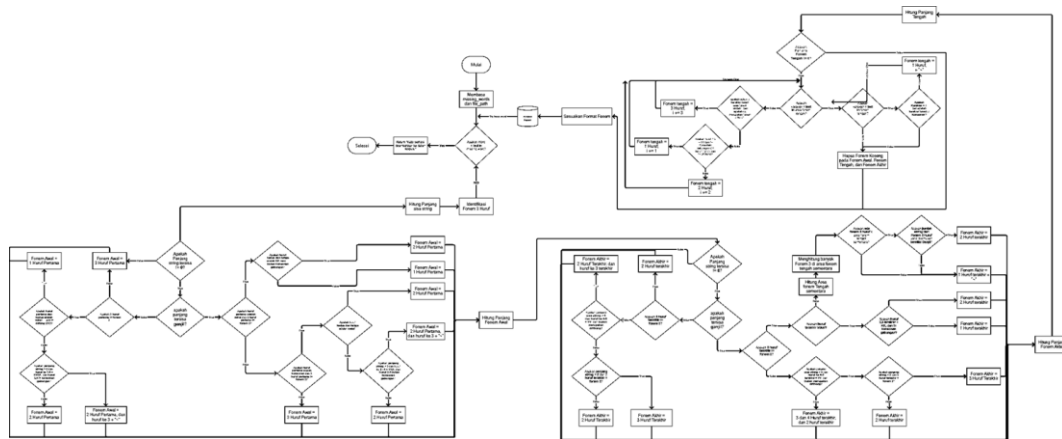
1. Menambahkan fungsi pada sistem yang dapat mengubah angka dan simbol ke dalam pelafalan.
2. Menambahkan metode yang dapat membedakan pembacaan huruf ê dan e.
3. Menambahkan kategori fonem lebih jauh seperti pola kata serapan dari bahasa asing atau bahasa daerah, sehingga mampu melakukan proses konversi teks menjadi suara lebih baik lagi.
4. Menambahkan metode yang dapat menyempurnakan dalam pemisahan fonem pada kata berimbuhan.
5. Menambahkan metode pengubah pitch suara untuk membaca kalimat Tanya, perintah, dan kalimat berita.
6. Menemukan metode pengubah pitch suara untuk mengubah intonasi audio sesuai dengan emosi.
7. Menemukan metode yang lebih baik dalam melakukan *voice over* pada perekaman dataset audio fonem.
8. Menemukan metode untuk pemisahan fonem berdasarkan pola struktur kata bahasa Indonesia lainnya .

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/>
- 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

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	zbba
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

Lampiran 3. Format kuesioner

1. Kuesioner Skenario 1 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	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			
				<i>Intelligibility</i>	<i>Fluidity</i>	<i>Naturalness</i>	<i>Size (Kb)</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 Terkonversi	Akurasi (100%)	Kategorisasi Fonem			
				<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						

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>
	Makan				
1	Malam				
	Jalan				
2	Sekarang				
3	Minum Kopi				
	Masuk				
4	Sekolah				
5	Belok Kanan				
	Kopi Ini				
6	Manis				
	Kepala Saya				
7	Sakit				
	Saya Mau				
8	Makan				
	Saya Sedang				
9	Masak				
	Ini Kamar				
10	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>
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>
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 kuisioner

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. Fadhlu 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

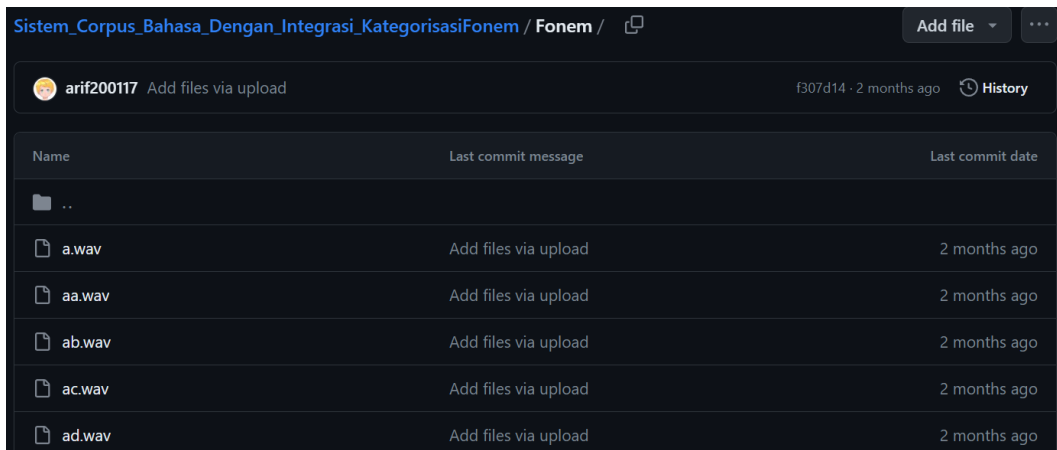
Link Hasil Kuesioner Sesi Kategorisasi Fonem : https://bit.ly/Survei_Kategorisasi_Fonem

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. Fadhlu 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
Total Akurasi / Nilai	3,07			2,62	

Link Hasil Kuesioner Sesi *Diphone Concatenation*: https://bit.ly/Survei_Diphone_Concatenation

Lampiran 5. *Database audio*



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

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()

    except:
        pass

    return wf

# 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, columnspan=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, columnspan=4, padx=5, pady=5)

tab1_display = tk.Text(tab1)
tab1_display.grid(row=7, column=0, columnspan=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

LEMBAR PERBAIKAN SKRIPSI

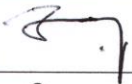

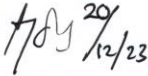
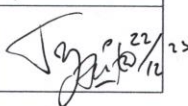
“IMPLEMENTASI ALGORITMA *KNUTH-MORRIS-PRATT* DALAM SISTEM PEMBUATAN KORPUS BAHASA INDONESIA DAN INTEGRASI KATEGORISASI FONEM UNTUK MENINGKATKAN AKURASI *TEXT-TO-SPEECH*”

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.	