



**PERANCANGAN APLIKASI TEXT READING DAN VOICE COMMAND
BAHASA INGGRIS MENGGUNAKAN METODE DIPHONE
CONCATENATION**

Disusun dan Diajukan untuk Memenuhi Persyaratan Ujian Akhir Memperoleh
Gelar Sarjana Komputer pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH

**NAMA : HERU ISNAN
NPM : 1214370382
PROGRAM STUDI : SISTEM KOMPUTER**

**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2019**

PERANCANGAN APLIKASI TEXT READING DAN VOICE COMMAND BAHASA INGGRIS MENGGUNAKAN METODE DIPHONE CONCATENATION

Ditulis dan Dijukan Sebagai Salah Satu Syarat Untuk Menempuh Skripsi
Memperoleh Gelar Sarjana Komputer Pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

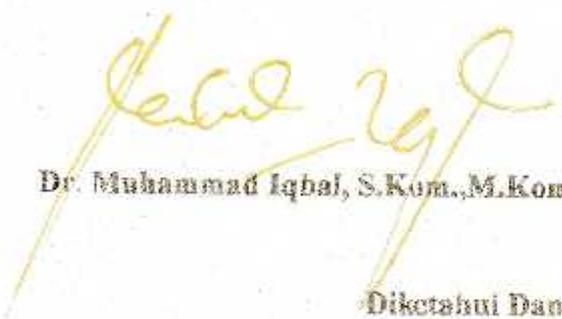
OLEH

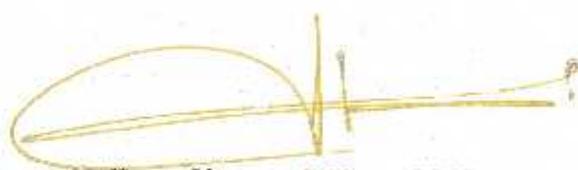
NAMA : HERU ISNAN
N.P.M : 1214370382
PROGRAM STUDI : SISTEM KOMPUTER

Diketahui Dan Disetujui Oleh

Dosen Pembimbing I

Dosen Pembimbing II


Dr. Muhammad Iqbal, S.Kom., M.Kom

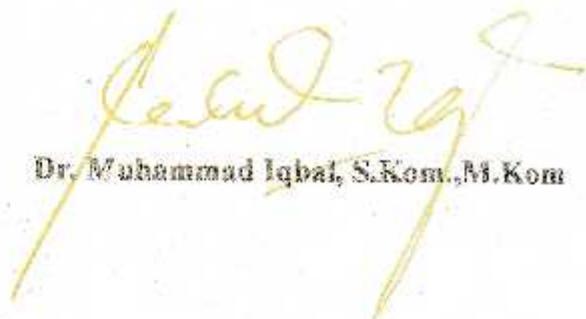

Zulham Sitorus, S.Kom., M.Kom

Diketahui Dan Disahkan Oleh

Dekan Fakultas Sains Dan Teknologi

Ketua Program Studi


Sri Shindi Yandira, ST., M.Sc


Dr. Muhammad Iqbal, S.Kom., M.Kom

SURAT PERNYATAAN

Saya yang bertandatangan di bawah ini :

Nama : Heru Isnain
NPM : 1214370382
Prodi : Sistem Komputer
Konsentrasi : SK (Sistem Komputer)
Judul Skripsi : Perancangan Aplikasi Text Reading Dan Voice
Command Menggunakan Diphone Concatenation

Dengan ini menyatakan bahwa :

1. Tugas Akhir/Skripsi saya bukan hasil Plagiat
2. Saya tidak akan menuntut perbaikan nilai indeks Prestasi Kumulatif (IPK) setelah ujian Sidang Meja Hijau
3. Skripsi saya dapat dipublikasikan oleh pihak lembaga, dan saya tidak akan menuntut akibat publikasi tersebut

Demikian pernyataan ini saya perbuat dengan sebenar-benarnya, terimakasih

Medan, 30 Agustus 2019

Yang membuat pernyataan



1214370382

Hal : Permohonan Meja Hijau



FM-BPAA-2012-041

Medan, 01 Juli 2019
 Kepada Yth : Bapak/Ibu Dekan
 Fakultas SAINS & TEKNOLOGI
 UNPAB Medan
 Di
 Tempat



Dengan hormat, saya yang bertanda tangan di bawah ini :

Nama : HERU ISNAN
 Tempat/Tgl. Lahir : Aceh Tenggara / 17 Mei 1993
 Nama Orang Tua : Syahrudin
 N. P. M : 1214370382
 Fakultas : SAINS & TEKNOLOGI
 Program Studi : Sistem Komputer
 No. HP : 085276595793
 Alamat : Desa Pehidinen Kec. Lawe Sigal-gala

Datang bermohon kepada Bapak/Ibu untuk dapat diterima mengikuti Ujian Meja Hijau dengan judul perancangan aplikasi text reading dan voice command bahasa Inggris dengan menggunakan metode concatenation, Selanjutnya saya menyatakan :

1. Melampirkan KKM yang telah disahkan oleh Ka. Prodi dan Dekan
2. Tidak akan menuntut ujian perbaikan nilai mata kuliah untuk perbaikan indek prestasi (IP), dan mohon diterbitkan ijazahnya setelah lulus ujian meja hijau.
3. Telah tercap keterangan bebas pustaka
4. Tertampir surat keterangan bebas laboratorium
5. Tertampir pas photo untuk ijazah ukuran 4x6 = 5 lembar dan 3x4 = 5 lembar Hitam Putih
6. Tertampir foto copy STTB SLTA dilegalisir 1 (satu) lembar dari bagi mahasiswa yang lanjutan D3 ke S1 lampirkan ijazah dan transkripnya sebanyak 1 lembar.
7. Tertampir pelunasan kwintasi pembayaran uang kuliah berjalan dan wisuda sebanyak 1 lembar
8. Skripsi sudah dijilid lux 2 examplar (1 untuk perpustakaan, 1 untuk mahasiswa) dan jilid kertas jeruk 5 examplar untuk penguji (bentuk dan warna penjiplakan diserahkan berdasarkan ketentuan fakultas yang berlaku) dan lembar persetujuan sudah di tandatangani dosen pembimbing, prodi dan dekan
9. Soft Copy Skripsi disimpan di CD sebanyak 2 disc (Sesuai dengan Judul Skripsinya)
10. Tertampir surat keterangan BKKOL (pada saat pengambilan ijazah)
11. Setelah menyelesaikan persyaratan point-point diatas berkas di masukan kedalam MAP
12. Bersedia melunaskan biaya-biaya uang dibebankan untuk memproses pelaksanaan ujian dimaksud, dengan rincian sbb :

1. [102] Ujian Meja Hijau	: Rp.	2.000.000
2. [170] Administrasi Wisuda	: Rp.	1.500.000
3. [202] Bebas Pustaka	: Rp.	100.000
4. [221] Bebas LAB	: Rp.	5.000
Total Biaya	: Rp.	1.505.000
		J. 855.000

Ukuran Toga : L

Hormat saya


 HERU ISNAN
 1214370382



Catatan :

- 1. Surat permohonan ini sah dan berlaku bila ;
 - a. Telah dicap Bukti Pelunasan dari UPT Perpustakaan UNPAB Medan.
 - b. Melampirkan Bukti Pembayaran Uang Kuliah aktif semester berjalan
- 2. Dibuat Rangkap 3 (tiga), untuk - Fakultas - untuk BPAA (asli) - Mhs.ybs.



Plagiarism Detector v. 977 - Originality Report:

Analyzed document: 25/08/2017 14:26:00

"Heru Isnan_ 1214370382_ Ilmu Komputer.docx"

Licensed to: Universitas Pembangunan Panca Budi



Relation chart:



Distribution graph:

Comparison Preset: Rewrite. Detected language: English

Top sources of plagiarism:

Percentage	Words	Source
% 25	wrds: 2082	https://www.researchgate.net/profile/Samsudin_Syahr/publication/305492511_PERANCANGAN_API_IKASI_TEXT...
% 24	wrds: 1996	https://ppm.ingunadharma.ac.id/public/modul/hp/PENERAPAN%20METODOE%20PONE%20PADA%20PERANCANGAN%20
% 18	wrds: 1667	https://pt.scribd.com/doc/549342507/105040018-Pengubah-Teks-Ka-Suara

Show other Sources:}

Processed resources details:

79 - Ok / 16 - Failed

Show other Sources:}

Important notes:

Wikipedia:	Google Books:	Ghostwriting services:	Anti-cheating:
[not detected]	[not detected]	[not detected]	[not detected]

Excluded Urls:

Included Urls:



UNIVERSITAS PEMBANGUNAN PANCA BUDI

FAKULTAS ILMU KOMPUTER

Jl. Jend. Gatot Subroto Km 4,5 Medan Fax. 061-8458077 PG.BOX : 1099 MEDAN

PROGRAM STUDI SISTEM KOMPUTER
PROGRAM STUDI TEKNIK KOMPUTER

(TERAKREDITASI)
(TERAKREDITASI)

PERMOHONAN MENGAJUKAN JUDUL SKRIPSI

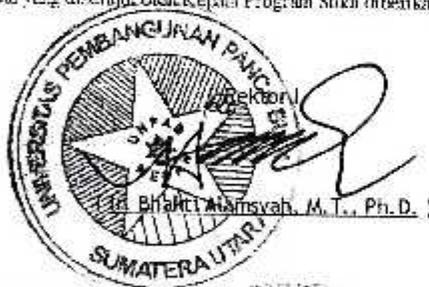
Saya yang bertanda tangan di bawah ini :

Nama Lengkap : HERU ISNAN
 Tempat/Tgl. Lahir : / 17 Mei 1993
 Nomor Pokok Mahasiswa : 1214370382
 Program Studi : Sistem Komputer
 Konsentrasi : Sistem Kendali Komputer
 Jumlah Kredit yang telah dicapai : 121 SKS, IPK 2.32

Dengan ini mengajukan judul skripsi sesuai dengan bidang ilmu, dengan judul:

No.	Judul Skripsi	Persetujuan
1.	sistem pendukung keputusan untuk menentukan mahasiswa terbaik di perguruan tinggi	<input type="checkbox"/>
2.	sistem pakar diagnosa gejala kecanduan game online dengan menggunakan metode certainty factory	<input type="checkbox"/>
3.	perancangan aplikasi text reading dan voice command bahasa Inggris dengan menggunakan metode concatenation	<input checked="" type="checkbox"/> 28/8/17

Judul yang disetujui oleh Kepala Program Studi dibenarkan tanda



Medan, 18 April 2017

Pembimbing I,

(HERU ISNAN)

Nomor :
Tanggal : 18 April 2017

Disahkan oleh :

Melisa Sari Pangajaitan, S.Ts, M.Kom
(~~Heru Isnand~~)

Tanggal :

Disetujui oleh :
Dosen Pembimbing I :

(Heru Isnand)

Tanggal :

Disetujui oleh :
Ka. Prodi Sistem Komputer

(Eko Harlyanto, S.Kom, M.Kom)

Tanggal :

Disetujui oleh :
Dosen Pembimbing II :

(Heru Isnand)

No. Dokumen: FM-LPPM-08-01

Revisi: 02

Tgl. Eff: 20 Des 2015



UNIVERSITAS PEMBANGUNAN PANCA BUDI
FAKULTAS ILMU KOMPUTER

Jl. Jend. Gatot Subroto Km. 4,5 Telp (061)-
Medan - Indonesia

FM-BPAA-2012-038

Universitas
Fakultas
Dosen Pembimbing I
Dosen Pembimbing II
Nama Mahasiswa
Jurusan/Program Studi
Nomor Pokok Mahasiswa
Jenjang Pendidikan
Judul Tugas Akhir /Skripsi

Universitas Pembangunan Panca Budi
Fakultas Ilmu Komputer
Dosen Pembimbing I: Karel SARIYATA, S.S.Pd, M.Ts
Dosen Pembimbing II: Zulwan Gatotus, S.Kom, M.Kom
Nama Mahasiswa: H. Pu. Satrio
Jurusan/Program Studi: Sistem Komputer
Nomor Pokok Mahasiswa: 1241370382
Jenjang Pendidikan: Strata Satu (S-1) / Diploma Tiga (D-III) *)
Judul Tugas Akhir /Skripsi: Pengembangan Aplikasi Test Reading Komparatif Command Bahasa Inggris pada Windows Antiface
DIPLOMA CONGRATULATION

Tanggal	Pembahasan Materi	Paraf	Keterangan
20/7/17	Revisi penulisan Bab I, II, III.		
29/7/17	Latihan belajar di bagian akhir di revisi		
2/8/17	masalah di bagian akhir dan akhir		
13/8/17	Acc. I, II, III lanjut IV-V		
20/8/17	Analisis pada keplemetri di sisi lain dan bab IV.		
24/8/17	Kejelasan ket. lain.		
25/8/17	Acc. Revisi		
25/9/19	Acc. Selang meja lisa		

2/9/19.

Medan,
Diketahui/Disetujui
oleh :
Dekan



*) Coret yang tidak perlu



YAYASAN PROF. DR. IT. KADIRIN YAHYA
UNIVERSITAS PEMBANGUNAN PANCA BUDI
LABORATORIUM KOMPUTER
Jl. Jend. Gatot Subroto Km 4,5 Sei Sikambing Telp. 061-8455571
Medan - 20122

KARTU BEBAS PRAKTIKUM

Yang bertanda tangan dibawah ini Ka. Laboratorium Komputer dengan ini menerangkan bahwa :

Nama : HERU ISNAN
N.P.M : 1214370382
Tingkat/Semester : Akhir
Fakultas : SAINS & TEKNOLOGI
Jurusan/Prodi : Sistem Komputer

Benar dan telah menyelesaikan urusan administrasi di Laboratorium Komputer Universitas Pembangunan Panca Budi Medan.

Medan, 02 Juli 2019

Ka. Laboratorium



ABSTRAK

HERU ISNAN

Perancangan Aplikasi Text Reading Dan Voice Command Bahasa Inggris Menggunakan Metode Diphone Concatenation

Salah satu pengenalan ucapan yang terkenal adalah yang dilakukan oleh Microsoft Corporation yang dikembangkan untuk sistem operasi Windows. Microsoft mengembangkan standar untuk mesin pengenalan ucapan, yaitu SAPI (Speech Application Programming Interface). SAPI memberikan kemampuan workstation untuk mengenali ucapan manusia sebagai masukan, dan membuat audio keluaran seperti manusia dari text tertulis. Kemampuan ini menambahkan dimensi baru interaksi manusia dan komputer. Layanan pengenalan ucapan dapat digunakan untuk memperluas penggunaan komputer bagi mereka yang menemui bawaha mengetik terlalu sulit karena keterbatasan fisik. Penelitian ini bertujuan untuk mengembangkan sebuah Aplikasi Text To Speech dan Voice Command. Model pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah model RAD. Penjelasan rinci tentang algoritma pengenalan ucapan berada di luar cakupan skripsi ini.

Kata kunci : Speech Application Programming Interface, Text To Speech, Voice Command

DAFTAR ISI

LEMBAR PENGESAHAN

SURAT PERNYATAAN

LEMBAR PENGESAHAN

Halaman

ABSTRAK ..	i
KATA PENGANTAR ..	ii
DAFTAR ISI ..	iv
DAFTAR GAMBAR ..	vi
DAFTAR TABEL.....	vii

BAB I. PENDAHULUAN 1

1.1 Latar Belakang Masalah.	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan dan manfaat penelitian	3
1.1 Tujuan penelitian.....	4
1.2 Manfaat Penelitian.	4
1.5 Sistematika Penulisan.....	5

BAB II. LANDASAN TEORI 6

2.1 Pengertian Teks.....	6
2.2 Suara.....	6
2.3 Dhipone.....	7
2.4 Diphone Concatenation.....	7
2.5 Text To Speech (TTS).....	8
2.6 Pengubahan Teks Menjadi Ucapan.....	11
2.7 Normalisasi Teks.....	14
2.8 Konversi Huruf Ke Fonem Dan Pengecualian.....	15
2.9 Pembangkitan Intonas.....	17
2.10 Analisis Fonetik.....	18
2.10.1. Speech Application Programming Interface	18
2.10.2 Pemrosesan Bahasa Alami	19
2.10.3 Voice Command.....	20
2.10.4 Text Reading	20

2.11	Pengklasifikasi (Classifier)	21
2.12	Use Case Diagram	23
2.13	Activity Diagram	26
2.14	Squence Diagram	27
2.15	Borland Delphi 7	28
2.16	Integrated Development Environment	29
2.17	Main menu, Speed Bar dan Palette Bar	29
2.18	Object Editor	30
2.19	Form	31
2.20	Code Editor	31
BAB III. ANALISIS DAN PERANCANGAN		32
3.1	Analisis	32
3.2	Perancangan Konversi Teks	33
3.3	Analisa Fonem Voice Command	36
3.4	Analisa String Matching Dalam Proses Konversi Suara	38
3.5	Proses Teks Menjadi Suara	42
3.6	Proses Diphone Bekerja	42
3.7	Use Case Dan Activity Diagram Serta Squence Diagram	44
3.8	Perancangan Aplikasi	47
BAB IV. IMPLEMENTASI PERANGKAT LUNAK		50
4.1	Kebutuhan Perangkat Lunak	50
4.2	Implementasi	50
BAB V. PENUTUP		57
5.1	Kesimpulan	57
5.2	Saran	57

DAFTAR PUSTAKA

BIOGRAFI PENULIS

LAMPIRAN - LAMPIRAN

BAB I

PENDAHULUAN

1. Latar Belakang Masalah

Ilmu dan teknologi terutama teknologi informasi berkembang sangat pesat. Pesatnya perkembangan teknologi ini berdampak pada berbagai perubahan sosial budaya, dan bahasa merupakan bagian terkena efek positif dari perkembangan teknologi komputer saat ini, banyak *software-software* yang beredar di pasaran yang berinteraksi dengan bahasa, seperti kamus elektronik, *google translate*, *voice command* dan *Text To Speech*.

Bahasa *Inggris* merupakan bahasa yang paling banyak digunakan dalam berkomunikasi setiap hari terutama dalam berkomunikasi dengan orang asing, dalam belajar bahasa *Inggris* banyak buku bahasa *Inggris* tersedia, tapi sangat sedikit *software* yang tersedia untuk membaca ataupun mengenali bahasa *Inggris*, dengan *software* tersebut user yang belajar bahasa *Inggris* bisa secara langsung mengetahui bagaimana pengucapan bahasa *Inggris* yang benar karena dilakukan dengan mendengar ucapan bahasa *Inggris* secara langsung.

Penguasaan bahasa *Inggris* sebagai bahasa *Internasional* merupakan salah satu akses untuk meraih kesuksesan di segala bidang. Didalam bahasa *Inggris*, pengucapan dari suatu kata berbeda dengan penulisannya. Ditambah lagi terdapat perbedaan pengucapan dan penulisan beberapa kata antara bahasa *Inggris* versi *British* dengan bahasa *Inggris* versi *American*. Bahasa *Inggris British* cenderung mempertahankan ejaan banyak kata yang asalnya dari *Perancis*, sedangkan *Inggris American* mencoba untuk mengeja kata lebih mendekati cara mereka melafalkannya

dan mereka menghilangkan huruf-huruf yang tidak diperlukan.

Untuk mengatasi permasalahan tersebut, penulis ingin membuat sebuah aplikasi *Text To Speech*, yaitu suatu aplikasi yang digunakan untuk mengkonversikan tulisan/teks kedalam bentuk ucapan dengan menggunakan pemodelan bahasa natural manusia. Dengan menggunakan aplikasi *Text To Speech* ini diharapkan pengguna bisa mempelajari cara pengucapan bahasa *Inggris* dengan tepat, baik versi *British* ataupun versi *American*.

Bagian konverter *phonem* ke ucapan akan menerima masukan berupa kode-kode *phonem* serta *pitch* dan durasi yang dihasilkan oleh bagian sebelumnya. Berdasarkan kode-kode tersebut, bagian konverter *phonem* ke ucapan akan menghasilkan bunyi atau sinyal ucapan yang sesuai dengan kalimat yang ingin diucapkan. Konversi dari teks ke *phonem* sangat dipengaruhi oleh aturan-aturan yang berlaku dalam suatu bahasa. Pada prinsipnya proses ini melakukan konversi dari simbol-simbol tekstual menjadi simbol-simbol *phonetik* yang merepresentasikan unit bunyi terkecil dalam suatu bahasa.

Diphone Concatenation merupakan suatu cara yang biasa digunakan dalam memeriksa kata atau kalimat yang akan dikonversi ke suara, proses *Diphone Concatenation* akan memarsing setiap karakter dalam suatu kata, kemudian kata yang mengandung karakter tersebut akan di kenali dan di konversi ke suara, dari penjelasan di atas saya berinisiatif mengambil judul ” **Perancangan Aplikasi *Text Reading* dan *Voice Command* Bahasa Inggris Menggunakan Metode *Diphone Concatenation*”.**

2. Rumusan Masalah

Berdasarkan latar belakang di atas diperoleh rumusan masalah dalam penelitian ini yaitu sebagai berikut :

- a. Bagaimana menerapkan metode *Diphone Concatenation* untuk mengkonversi dari text ke suara serta memiliki kemampuan mengucapkan per kata maupun per kalimat.
- b. Bagaimana merancang aplikasi yang bisa mengkonversi dari text ke suara serta memiliki kemampuan mengucapkan per kata maupun per kalimat metode *Diphone Concatenation*.

3. Batasan Masalah

Sesuai dengan latar belakang pemilihan judul di atas, maka yang menjadi permasalahan di dalam skripsi ini adalah sebagai berikut:

- a. *Diphone Concetanation* yang digunakan adalah *Microsoft Speech Application Programming Interface*.
- b. *Text* yang dikenali adalah text dalam bahasa Inggris.
- c. *Text* yang dikonversi menjadi suara didapat berdasarkan input pengguna atau diambil dari file text *.txt.
- d. Kata atau kalimat yang tidak terdeteksi dalam fasa bahasa Inggris akan dibaca dengan dialek Inggris jika kata tersebut bisa dibaca, sehingga pelafalannya dapat terbaca.
- e. Aplikasi yang dirancang menggunakan bahasa pemrograman Delphi 7.

4. Tujuan dan Manfaat Penelitian

1. Tujuan Penelitian

Tujuan yang ingin diperoleh penulis adalah:

- a. Membuat sebuah program aplikasi yang bisa membaca kalimat bahasa Inggris secara baik dan benar.
- b. Mengetahui *Speech Application Programming Interface* dalam proses konversi text ke suara.

2. Manfaat Penelitian

Sedangkan manfaat yang ingin dicapai adalah:

- a. Memahami secara lebih dalam bagaimana proses *Diphone Concatenation* bekerja.
- b. Membuat suatu sistem program yang mudah digunakan dalam memeriksa kata atau kalimat dalam bahasa Inggris.

5. Sistematika Penulisan

Sistematika penyusunan Skripsi ini dibagi menjadi lima bab, sesuai dengan sistematika/ketentuan dalam pembuatan Skripsi, adapun pembagian bab-bab tersebut adalah:

BAB I : PENDAHULUAN

Bab ini menerangkan tentang latar belakang masalah, perumusan masalah, ruang lingkup masalah, tujuan, manfaat, metode penelitian dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Dalam bab ini akan diuraikan mengenai konsep serta sekilas tentang *software* yang digunakan.

BAB III : ANALISA DAN PERANCANGAN

Disini penulis membahas mengenai analisa dan perancangan berdasarkan judul serta dasar teori yang telah dibuat.

BAB IV : IMPLEMENTASI DAN HASIL

Dalam bab ini akan dibahas tentang implementasi sistem yang telah dirancang.

BAB V : KESIMPULAN DAN SARAN

Bab terakhir ini berisi tentang kesimpulan dan saran dari Skripsi yang telah dibuat.

BAB II

LANDASAN TEORI

1. Pengertian Teks

Setiap data dalam berbentuk karakter yang dikenali oleh komputer akan diterjemahkan kedalam kumpulan jajaran bit. Untuk menterjemahkan karakter-karakter tersebut digunakan ASCII (*American Standart Code for Information Interchange*), merupakan kode yang digunakan secara umum pada saat ini. ASCII merupakan kombinasi kode 8 bit, yang terdiri atas 7 bit data dan 1 bit parity, sehingga mempunyai 2^7 atau 128 kode karakter yang berbeda dan unik yang terdiri dari bit 0 dan bit 1. Kode ini digunakan dalam Personal Computer (PC). Piranti yang menggunakan kode ini perlu menterjemahkan 1 bit didepan sebagai parity. Bit parity berfungsi sebagai tanda kesalahan dalam pengiriman data selama komunikasi, yang terdiri atas parity genap (bit 1 apabila jumlah bit 1 dalam 7 deretan bit data berjumlah genap) dan parity ganjil (bit 1 apabila jumlah bit 1 dalam 7 deretan bit data berjumlah ganjil).

2. Suara

Suara adalah fenomena fisik yang dihasilkan oleh getaran benda, getaran suatu benda yang berupa sinyal analog dengan amplitudo yang berubah secara kontinyu terhadap waktu. Konsep dasar suara dihasilkan oleh getaran suatu benda. Selama bergetar perbedaan tekanan terjadi di udara sekitarnya ini yang disebut dengan Gelombang. Gelombang mempunyai pola sama yang berulang pada interval tertentu, yang disebut sebagai "Periode" (Bambang Ela Purnama, 2013).

Suara yang berada pada batas pendengaran manusia sebagai “*audio*”, dan gelombangnya sebagai “*acoustic signals*”. Suara di luar batas pendengaran manusia dapat dikatakan sebagai “*noise*” (getaran yang tidak teratur dan tidak berurutan dalam berbagai frekuensi, tidak dapat di dengar manusia). Format audio yang ada saat ini antara lain : AAC (*Advanced Audio Coding*) [.m4a], *WAVEFORM AUDIO* [.WAV], *Audio Interchange File Format* [.AIF], *Audio CD* [.cda], *Mpeg Audio Layer 3* [.mp3], MIDI (*Music Instrument Digital Interface*) dan sebagainya (Bambang Ela Purnama, 2013).

3. *Diphone*

Diphone merupakan sebuah database yang digunakan oleh *MBrola engine* untuk melakukan pemeriksaan *string* yang akan dirubah ke dalam bentuk suara, teks yang akan dirubah dalam bentuk suara harus melewati penyaringan kata yang sesuai dengan diphone terlebih dahulu, tanpa adanya diphone maka proses pengubahan teks menjadi suara tidak akan bisa dilakukan (Angga Narullita, 2012).

4. *Diphone Concatenation*

Diphone Concatenation berfungsi untuk mengubah kalimat masukan dalam bahasa tertentu dalam bentuk teks ke bentuk suara, yang biasanya diwakili oleh fonem kode, durasi dan pitch.

Fonem ke fungsi *Speech Converter* digunakan untuk menghasilkan sinyal suara berdasarkan kode fonem dihasilkan dari proses sebelumnya. Bagian ini sebagian besar melibatkan *diphone concatenation*, yang harus didukung oleh *database diphone* yang terdiri dicatat segmen suara seperti kombinasi dari dua atau lebih bentuk fonem suara. Ucapan bahasa adalah dibentuk oleh satu set suara yang berbeda dari bahasa lain. Ini adalah alasan utama mengapa setiap bahasa harus

memiliki database diphone sendiri dan hingga saat penelitian ini dibuat belum ada bentuk baku untuk fonem bahasa indonesia selain dari bentuk kata dan juga cara pengucapan bisa berbeda dikarenakan beberapa dialek bahasa indonesia itu sendiri, berbeda dengan bahasa inggris dan bahasa lainnya (Sutarman, 2015).

5. *Text To Speech (TTS)*

Text to Speech adalah suatu sistem yang dapat mengubah deretan kata-kata sebagai masukan menjadi ucapan sebagai keluaran. Sistem TTS juga biasa disebut dengan Sistem pensintesis ucapan. Sistem ini dapat digunakan untuk berbagai aplikasi, misalnya sistem informasi tagihan telepon atau sistem informasi lainnya yang diucapkan secara lisan. TTS juga dapat digunakan untuk mengubah teks pesan SMS ke ucapan sehingga pesan SMS dapat didengar. Metode sintesis ucapan memungkinkan mesin dapat melewatkan perintah atau informasi kepada pengguna lewat “ucapan”. Proses ini melibatkan pemecahan kata menjadi fonem, menganalisis untuk penanganan khusus dari teks seperti angka, jumlah mata uang, perubahan nada suara, dan juga pemberian tanda-tanda baca (Iwan Iwut Tritoasmoro, 2006).

Pesan atau informasi yang dikirimkan lewat ucapan memiliki kelebihan antara lain :

- a. Pengguna dapat dengan mudah memahami pesan atau informasi tanpa perlu intensitas konsentrasi tinggi.
- b. Pesan atau informasi dapat diterima saat pengguna sedang terlibat dengan aktivitas lain, misalnya saat berjalan, menangani atau sedang melihat objek lain.
- c. Pesan atau informasi bisa dibuat dalam banyak bahasa.

Sistem *Text to Speech* pada prinsipnya terdiri dari dua sub sistem, yaitu bagian Konverter Teks ke Fonem (*Text to Phoneme*) dan bagian Konverter Fonem ke Ucapan (*Phoneme to Speech*) (Pande Made Mahendri Pramadewi, 2012:220).

Bagian konverter teks ke fonem berfungsi untuk mengubah kalimat masukan dalam suatu bahasa tertentu yang berbentuk teks menjadi rangkaian kode-kode bunyi yang biasanya direpresentasikan dengan kode fonem, durasi serta pitch-nya. Untuk suatu bahasa baru, bagian ini harus dikembangkan secara lengkap khusus untuk bahasa tersebut (Pande Made Mahendri Pramadewi, 2012:220).

Fonem merupakan unit bunyi terkecil yang dapat dibedakan oleh manusia, dan suatu ucapan kata atau kalimat pada prinsipnya dapat dilihat sebagai urutan fonem. Durasi adalah lama waktu pengucapan untuk setiap fonem dan pitch atau nada merupakan nilai frekuensi dasar pada pengucapan fonem untuk menghasilkan intonasi yang diinginkan (Pande Made Mahendri Pramadewi, 2012:220). Proses konversi dari teks ke fonem terdiri dari beberapa tahap, yaitu normalisasi teks, konversi setiap fonem menjadi kode fonem dan penetapan durasi dan pitch untuk setiap fonem. Normalisasi teks merupakan suatu proses yang merepresentasikan teks tertulis menjadi teks yang sesuai dengan pengucapan oleh manusia (Pande Made Mahendri Pramadewi, 2012:220).

Konverter fonem ke ucapan berfungsi untuk membangkitkan sinyal ucapan berdasarkan kode-kode fonem yang dihasilkan dari proses sebelumnya (Pande Made Mahendri Pramadewi, 2012:220). Pembentukan ucapan pada pensintesa ucapan menggunakan metode diphone concatenation pada prinsipnya dilakukan dengan cara menyusun sejumlah diphone yang bersesuaian sehingga diperoleh ucapan yang diinginkan. Diphone adalah unit suara yang terdiri atas gabungan dua fonem (Pande Made Mahendri Pramadewi, 2012:220).

Dalam mengubah teks menjadi audio, TTS engine menggunakan bermacam-macam metode, antara lain:

a. Penggabungan frasa kata.

Metode ini menggabungkan frasa kata yang sebelumnya telah direkam untuk membentuk sebuah kalimat dan merupakan metode yang paling mudah serta paling banyak digunakan saat ini. Kebanyakan sistem voice-mail menggunakan metode ini. Sebagai contoh, pesan pada voice-mail "Anda mempunyai [dua] buah pesan", ini merupakan pesan yang terdiri dari tiga bagian yaitu dua buah pesan yang bersifat statis "Anda mempunyai" dan "buah pesan" serta sebuah pesan yang bersifat dinamis tetapi telah dipersiapkan sebelumnya yaitu "dua".

b. Sintesis kata.

Metode ini menghasilkan sintesis atau tiruan kata secara elektronik dengan menerapkan algoritma perhitungan yang kompleks untuk mensimulasikan pita suara, rongga mulut, bentuk bibir dan posisi lidah. Suara yang dihasilkan dari metode ini seperti suara robot tetapi dengan algoritma yang telah ada pada text to speech engine menjadi seperti suara manusia.

c. Penggabungan frasa kata dan sintesis kata.

Metode ini menggabungkan segmen audio dan menggunakan algoritma perhitungan untuk menghaluskan jeda guna menghasilkan suara yang utuh. Contohnya adalah "hello", terdiri dari empat segmen. Text to speech mengkonversi teks dalam format suatu bahasa menjadi ucapan sesuai dengan pembacaan teks dalam bahasa yang digunakan. Sistem text to speech memerlukan dua proses konversi, yaitu konversi teks ke fonem dan konversi fonem ke ucapan. Kedua proses tersebut dilakukan secara berurutan dengan masukan berupa teks dan keluaran berupa ucapan.

Proses konversi teks ke fonem berfungsi untuk mengolah kalimat masukan dalam suatu bahasa tertentu yang berbentuk teks menjadi rangkaian kodekode bunyi yang direpresentasikan dengan kode fonem, durasi serta pitch-nya. Kode-kode fonem adalah kode yang merepresentasikan unit bunyi yang ingin diucapkan. Pengucapan kata atau kalimat pada prinsipnya adalah urutan bunyi atau secara simbolik adalah urutan kode fonem.

Konversi dari teks ke fonem sangat dipengaruhi oleh aturan-aturan yang berlaku dalam suatu bahasa. Pada prinsipnya proses ini melakukan konversi dari simbol-simbol tekstual menjadi simbol-simbol fonetik yang merepresentasikan unit bunyi terkecil dalam suatu bahasa. Setiap bahasa memiliki aturan cara pembacaan dan cara pengucapan teks yang sangat spesifik. Hal ini menyebabkan implementasi unit konverter teks ke fonem menjadi sangat spesifik terhadap suatu bahasa. Untuk mendapatkan ucapan yang lebih alami, ucapan yang dihasilkan harus memiliki intonasi (*prosody*). Secara kuantisasi, prosodi adalah perubahan nilai frekuensi dasar (*pitch*) selama pengucapan kalimat dilakukan atau *pitch* sebagai fungsi waktu (Nessa Putri Andayu, 2012).

6. Perubahan Teks Menjadi Ucapan

Sistem *Text to Speech* pada prinsipnya terdiri dari dua subsistem dasar, yaitu:

a. Subsistem konverter teks ke fonem

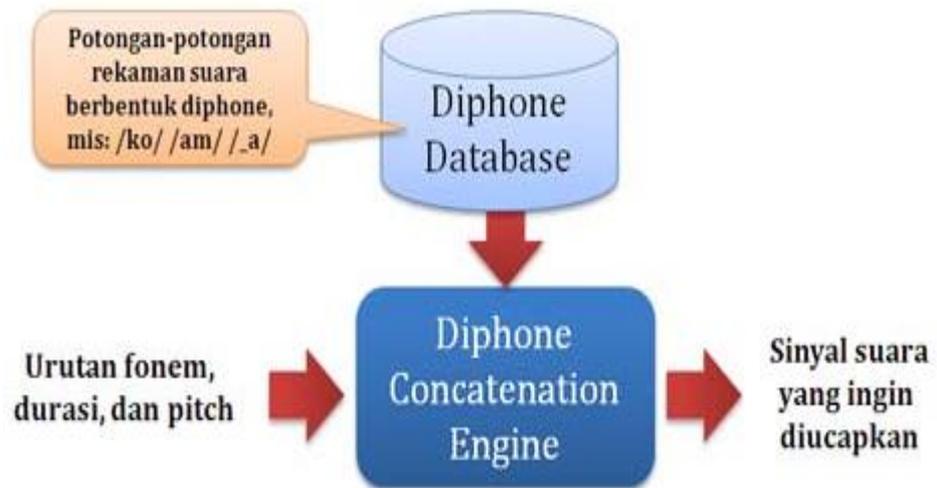
Subsistem konverter teks ke fonem yang memiliki dua fungsi utama. Pertama adalah mengambil kalimat masukan dalam suatu bahasa tertentu yang berbentuk barisan teks dan mengubah beberapa hal seperti nomor dan tanda ke dalam tulisan sesuai dengan bunyi yang seharusnya, sering disebut

dengan normalisasi teks (*text normalization*). Kemudian menentukan kode fonetik (*phonetic transcriptions*) untuk tiap kata beserta durasi dan nadanya. Kode fonem adalah kode yang merepresentasikan unit bunyi yang ingin diucapkan. Pengucapan kata atau kalimat pada prinsipnya adalah urutan bunyi atau secara simbolik adalah urutan kode fonem.

b. Subsistem konverter fonem ke ucapan

Subsistem konverter fonem ke ucapan yang akan menerima masukan kode-kode fonem serta nada dan durasi yang telah dihasilkan oleh bagian sebelumnya. Berdasarkan kode-kode tersebut bagian ini akan menghasilkan bunyi atau sinyal ucapan yang sesuai dengan kalimat yang ingin diucapkan. Ada beberapa alternatif teknik yang dapat digunakan untuk implementasi bagian ini. Salah satu teknik yang digunakan adalah penyambungan diphone (Iwan Iwut Tritoasmoro, 2006).

Pada sistem yang menggunakan teknik penyambungan diphone, sistem harus didukung oleh suatu basis data diphone yang berisi rekaman segmen-segmen ucapan yang berupa diphone. Ucapan dari suatu bahasa dibentuk dari satu set bunyi mungkin berbeda untuk setiap bahasa, oleh karena itu setiap bahasa harus dilengkapi dengan basis data diphone yang berbeda. Gambar 2.1 menunjukkan diagram alir subsistem pensintesis ucapan.



[<http://teknologibahasa.wordpress.com>]

Gambar 2.1 Subsistem Pensintesis Ucapan.

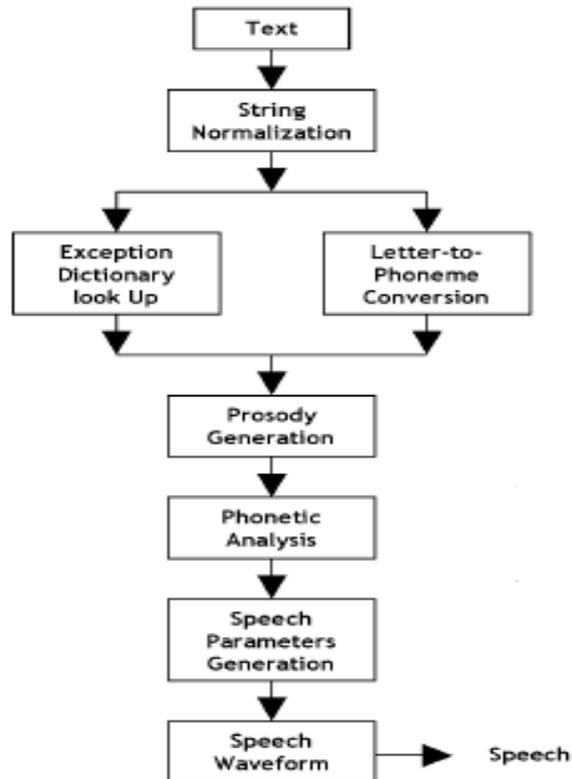
Sumber : Arry Ahmad Arman, 2004

Supaya pensitesa ucapan dapat mengucapkan semua kemungkinan kata atau kalimat yang ada dalam suatu bahasa, sehingga sistem tersebut harus didukung oleh diphone database yang terdiri dari semua kombinasi diphone yang ada dalam bahasa tersebut.

Diphone concatenation engine atau unit pemroses *Diphone* akan menerima masukan berupa daftar fonem yang ingin diucapkan, masing-masing disertai oleh durasi pengucapannya, serta pitch atau frekuensinya. Berdasarkan daftar fonem yang diterima, unit ini akan menentukan susunan diphone yang sesuai. Selanjutnya, unit ini akan melakukan smoothing sambungan antar diphone, melakukan manipulasi durasi pengucapan serta manipulasi pitch. Pada akhirnya, diphone concatenation engine akan menghasilkan sinyal ucapan yang sesuai (Iwan Iwut Tritoasmoro, 2006)

Setiap bahasa memiliki aturan cara pembacaan dan cara pengucapan teks yang sangat spesifik. Hal ini menyebabkan implementasi unit konverter teks ke

fonem menjadi sangat spesifik terhadap suatu bahasa. Tahapan-tahapan utama konversi dari teks menjadi ucapan dapat dilihat pada Gambar 2.2.



Gambar 2.2 Urutan Proses Konversi Dari Teks Ke Ucapan.

Sumber : Iwan Iwut Tritoasmoro, 2006

Tahapan-tahapan tersebut ialah normalisasi teks, konversi huruf ke fonem dan pengecualian, pembangkitan intonasi, analisis fonetik, dan generator sinyal ucapan. Berikut ini adalah penjelasan dari masing-masing tahapan.

7. Normalisasi Teks

Tahap pertama dari sistem konversi teks ke ucapan adalah tahap normalisasi teks. Pada tahap ini semua teks kalimat diubah menjadi teks yang secara lengkap memperlihatkan cara pengucapannya. Normalisasi teks meliputi perubahan singkatan, akronim, angka, tanggal, waktu, karakter-karakter khusus, dan simbol-

simbol dengan bentuk huruf alphabet lengkap sehingga tidak terjadi ambiguitas berkenaan dengan cara pengucapan (Iwan Iwut Tritoasmoro, 2006).

Berkaitan dengan perubahan teks kalimat, tahapan normalisasi teks sendiri terdiri dari beberapa blok bagian meliputi :

- a. *konverter angka*, bagian konverter angka berfungsi mengubah angka menjadi deretan huruf alphabet yang menggambarkan cara pengucapannya, sebagai contoh angka 12550 akan diubah menjadi dua belas ribu lima ratus lima puluh.
- b. Bagian *konverter akronim*, berfungsi mengubah akronim menjadi huruf tunggal atau deretan huruf alphabet yang menggambarkan cara pengucapannya, sebagai contoh singkatan “dlm” akan diubah menjadi “dalam”.
- c. Bagian *konverter simbol dan karakter khusus* akan mengubah karakter dan simbol menjadi format teks sesuai dengan cara pengucapannya, sebagai contoh simbol “%” pada kalimat “5%” akan terbaca “lima persen”.

Subsistem ini harus memiliki pustaka setiap unit ucapan dari suatu bahasa. Perubahan angka, akronim, singkatan, simbol, dan karakter khusus sangat tergantung pada basisdata yang digunakan sistem. Ucapan dari suatu bahasa dibentuk dari satu set ucapan bunyi yang mungkin berbeda untuk setiap bahasa, oleh karena itu setiap bahasa harus dilengkapi dengan basisdata yang berbeda (Iwan Iwut Tritoasmoro, 2006).

8. Konversi Huruf Ke Fonem Dan Pengecualian

Tahap selanjutnya setelah normalisasi teks adalah proses konversi huruf ke fonem. Tahapan ini bertujuan untuk mendapatkan ketentuan pengucapan dasar dari setiap kata teks yang telah dinormalisasi. Konversi huruf menjadi fonem biasanya dilakukan dengan dua pendekatan dasar yaitu :

- a. Pendekatan berdasarkan kamus.

Pendekatan ini menggunakan suatu pendekatan yang menggunakan sebuah kamus besar dari sebuah bahasa yang terdiri dari semua kata, dan pengucapannya yang benar kemudian disimpan pada sebuah program.

Pendekatan ini memiliki keuntungan lebih cepat dan teliti dalam proses pengubahan huruf ke fonem tetapi akan berbeda jika kata yang diberikan tidak ada dalam kamus. Selain itu kamus memiliki perbendaharaan kata yang sangat banyak sehingga dibutuhkan ruang memori yang cukup besar untuk penggunaannya.

- b. Pendekatan menurut aturan (*Rule-based approach*).

Pendekatan ini menggunakan aturan yang telah ditentukan untuk pengucapan kata berdasarkan ejaannya. Pendekatan ini bekerja pada masukan apa saja, tetapi kerumitan dari aturan berkembang sehingga membutuhkan ejaan dan pengucapan yang besarnya tidak tentu.

Pendekatan ini dapat diimplementasikan dengan tabel konversi yang berisi pasangan antara urutan huruf dan urutan fonem, bahkan mungkin hanya berisi satu huruf dan satu fonem. Aturan yang lebih sulit biasanya diimplementasikan dengan tabel konversi yang akan diterapkan jika kondisi rangkaian huruf tetangga kiri dan tetangga kanannya terpenuhi. Contoh bentuk aturan konversi huruf ke fonem yang memenuhi teknik tersebut adalah sebagai berikut:

teks-kiri[deretan huruf] teks-kanan = deretan fonem

Huruf tertentu yang ditunjuk dalam posisi [deretan huruf] akan diubah menjadi fonem dalam “deretan fonem” jika teks kiri dan teks kanan terpenuhi.

Bahasa Inggris termasuk bahasa yang mempunyai keteraturan yang rendah untuk proses konversi teks ke fonem. Suatu sistem TTS bahasa Inggris biasanya

dilengkapi dengan suatu basis data yang berisi ribuan kata serta konversi padanan urutan fonemnya.

Bahasa Indonesia termasuk bahasa yang jelas aturan konversinya. Sebagian besar kata dalam bahasa Indonesia dapat diubah menjadi fonem dengan aturan yang jelas dan sederhana, walaupun tetap ada kondisi-kondisi yang tidak dapat ditemukan keteraturannya, sebagai contoh simbol huruf e dapat diucapkan sebagai e pepet atau e taling, artinya harus dikonversikan menjadi fonem yang berbeda untuk kondisi yang berbeda. Kondisi yang masih dapat ditangani oleh aturan diimplementasikan dengan blok konversi huruf ke fonem. Konversi yang tidak teratur ditangani oleh bagian tabel pengecualian (Pande Made Mahendri Pramadewi, 2013).

9. Pembangkitan Intonasi

Gejala intonasi atau prosodi, mempunyai hubungan yang erat dengan struktur kalimat dan interelasi kalimat dalam suatu wacana. Karena sifatnya yang sangat subyektif, pembahasan intonasi lebih ditekankan pada aspek gramatikal dibandingkan aspek emosional (Pande Made Mahendri Pramadewi, 2013).

Sebuah ucapan dilafalkan oleh penuturnya dalam pola melodi tertentu yang diterima pendengarnya sebagai sebuah deret nada dengan tinggi yang berbeda. Tingkat tinggi nada tidak dapat ditentukan secara pasti, sangat bervariasi tergantung dari jenis kelamin, emosi atau sikap (Pande Made Mahendri Pramadewi, 2013).

Prosodi adalah perubahan nilai-nilai nada atau frekuensi dasar selama pengucapan kalimat dilakukan atau nada sebagai satuan waktu. Pada prakteknya, informasi pembentuk prosodi berupa data nada serta durasi pengucapannya untuk setiap fonem yang dibangkitkan. Prosodi bersifat sangat spesifik untuk setiap bahasa, sehingga model yang diperlukan untuk membangkitkan data-data prosodi menjadi

sangat spesifik untuk setiap bahasa. Beberapa metode pemodelan intonasi diantaranya adalah *Tone Sequence Model*, yang membangkitkan intonasi dengan cara merepresentasikan suatu kalimat dengan kuat lemahnya suku kata dan urutan tinggi rendahnya nada berkaitan dengan tekanan suku kata dan jeda prosodi (Pande Made Mahendri Pramadewi, 2013).

10. Analisis Fonetik

Satu tahap berikutnya yang masih sering dilakukan adalah analisis fonetik. Tahap ini dapat dikatakan sebagai tahap penyempurnaan, yaitu melakukan perbaikan di tingkat bunyi, sebagai contoh dalam bahasa Indonesia, fonem /k/ dalam kata “bapak” tidak pernah diucapkan secara tegas, atau adanya sisipan fonem /y/ dalam pengucapan kata “alamiah” antara fonem /i/ dan /a/ (Pande Made Mahendri Pramadewi, 2013).

a. *Speech Application Programming Interface*

Dua tipe dasar mesin SAPI adalah sistem teks ke ucapan (*Text-To-Speech*) dan pengenalan ucapan (*Speech Recognition*). Sistem TTS mensintesis string teks dan file ke dalam audio ucapan menggunakan suara sintetik. Pengenalan ucapan mengkonversi audio yang diucapkan manusia ke dalam string teks yang dapat dibaca dan file (Abdusy Syarif, 2011).

Aplikasi dapat mengontrol teks ke ucapan dengan menggunakan Component Object Model (COM) ISpVoice. Setelah aplikasi menciptakan suatu obyek ISpVoice, aplikasi hanya perlu untuk memanggil ISpVoice::Speak untuk menghasilkan keluaran ucapan dari beberapa data teks (Abdusy Syarif, 2011).

Sebuah aplikasi memiliki dua pilihan jenis mesin pengenalan ucapan (ISpRecognizer). Sebuah pengenalan bersama (shared recognition) yang mungkin

bisa dibagi dengan aplikasi pengenalan ucapan lainnya dianjurkan untuk sebagian besar aplikasi pengenalan ucapan. Untuk membuat ISpRecoContext untuk ISpRecognizer bersama (shared), aplikasi hanya perlu melakukan panggilan COM CoCreateInstance pada komponen CLSID_SpSharedRecoContext. Untuk aplikasi server besar yang akan berjalan sendirian pada sebuah sistem, dan kinerja adalah kunci, mesin pengenalan ucapan InProc yang lebih sesuai (Abdusy Syarif, 2011).

Dalam rangka menciptakan ISpRecoContext untuk ISpRecognizer InProc, pertama aplikasi harus memanggil CoCreateInstance pada komponen CLSID_SpInprocRecoInstance untuk membuat sendiri ISpRecognizer InProc. Selanjutnya, aplikasi dapat memanggil ISpRecognizer::CreateRecoContext untuk mendapatkan sebuah ISpRecoContext. Akhirnya, sebuah aplikasi pengenalan ucapan harus membuat (create), memuat (load), dan mengaktifkan ISpRecoGrammar, yang pada dasarnya menunjukkan jenis ujaran untuk dikenali, yaitu tata bahasa dikte (dictation) atau perintah dan kontrol (command and control). Pertama, aplikasi ini menciptakan ISpRecoGrammar menggunakan ISpRecoContext::CreateGrammar. Kemudian, memuat aplikasi tata bahasa yang sesuai, baik dengan memanggil ISpRecoGrammar::LoadDictation untuk dikte atau salah satu dari cara ISpRecoGrammar::LoadCmdxxx untuk perintah dan kontrol. Akhirnya, untuk mengaktifkan tata bahasa ini sehingga pengenalan ucapan dapat memulai aplikasi dengan memanggil ISpRecoGrammar::SetDictationState untuk dikte atau ISpRecoGrammar::SetRuleState atau ISpRecoGrammar::SetRuleIdState untuk perintah dan control (Abdusy Syarif, 2011).

b. Pemrosesan Bahasa Alami

Pada prinsipnya bahasa alami adalah suatu bentuk representasi dari suatu pesan yang ingin dikomunikasikan antar manusia. Bentuk utama representasinya

adalah berupa suara/ ucapan (spoken language), tetapi sering pula dinyatakan dalam bentuk tulisan (Pande Made Mahendri Pramadewi, 2012:220).

Suatu sistem pemrosesan bahasa alami secara lisan dapat dibentuk dari tiga sub sistem, yaitu pertama, sub sistem *Natural Language Processing* (NLP), berfungsi untuk melakukan pemrosesan secara simbolik terhadap bahasa tulisan. Beberapa bentuk aplikasi sub sistem ini adalah translator bahasa alami (misalnya dari Bahasa Inggris ke Bahasa Indonesia), sistem pemeriksa sintaks bahasa, sistem yang dapat “menyimpulkan” suatu narasi, dan sebagainya. Kedua, sub sistem Text to Speech (TTS), berfungsi untuk mengubah teks (bahasa tulisan) menjadi ucapan (bahasa lisan). Ketiga, sub sistem Speech Recognition (SR), merupakan kebalikan teknologi Text to Speech, yaitu sistem yang berfungsi untuk mengubah atau mengenali suatu ucapan (bahasa lisan) menjadi teks (bahasa tulisan) (Pande Made Mahendri Pramadewi, 2012:220).

c. *Voice Command*

Voice Command adalah fitur perintah suara untuk menjalankan suatu aplikasi yang terdapat pada handphone atau alat teknologi lainnya, untuk mempermudah pengguna dalam menggunakan fitur tersebut. Tapi kebanyakan pengguna jarang atau bahkan tak pernah menggunakan fitur *Voice Command* ini (Anna Dara Andriana, 2013).

d. *Text Reading*

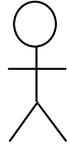
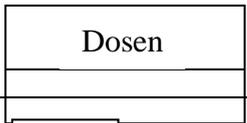
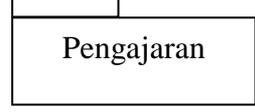
Text Reading adalah Perangkat lunak yang mengubah teks menjadi suara. *Text Reading* ini dapat membaca beberapa jenis jenis file tertentu dan halaman web dengan jelas. *Text Reading* juga menyoroti kata-kata dibaca, sehingga pengguna dapat mengikuti kata yang sedang dibaca (<https://www.understood.org/en/school->

learning/assistive-technology/assistive-technologies-basics/text-to-speech-software-what-it-is-and-how-it-works).

11. Pengklasifikasi (*Classifier*)

Pengklasifikasi (*classifier*) pada prinsipnya merupakan konsep diskret dalam model yang memiliki identitas (*identity*), state, perilaku (*behaviour*), serta relasi dengan pengklasifikasi yang lainnya (*relationship*). Jenis pengklasifikasi mencakup di dalamnya *use case*, komponen, simpul (*node*), serta subsistem.

Tabel 2.1. Beberapa jenis pengklasifikasi

Pengklasifikasi	Kegunaan	Notasi
<i>Actor</i>	Menggambarkan semua objek di luar sistem (bukan hanya pengguna sistem/perangkat lunak) yang berinteraksi dengan sistem yang dikembangkan.	 Mahasiswa
<i>Use Case</i>	Menggambarkan fungsionalitas yang dimiliki sistem	 Mengajar
Kelas (<i>Class</i>)	Menggambarkan konsep dasar pemodelan sistem.	 Dosen
Subsistem (<i>subsistem</i>)	Menggambarkan paket spesifikasi serta implementasi	 Pengajaran
Komponen (<i>Component</i>)	Menggambarkan bagian-bagian fisik sistem/perangkat lunak yang dikembangkan.	 Pengajaran

Antarmuka (<i>interface</i>)	Menggambarkan antarmuka pengiriman pesan (<i>message</i>) antarpengklasifikasi	 Antarmuka Pengajar- Mahasiswa
Simpul (<i>node</i>)	Menggambarkan sumber daya komputasional yang digunakan oleh sistem.	 Server

Sumber : Adi Nugroho (2010 : 16)

Kelas sesungguhnya merupakan himpunan dari objek-objek yang memiliki struktur-struktur yang serupa, serta memiliki perilaku dan relasi yang serupa pula. Ada beberapa himpunan objek yang bisa dipertimbangkan untuk menjadi kandidat-kandidat kelas, diantaranya:

a. Kelas Konsep

Konsep adalah ide atau pemahaman yang kita miliki tentang suatu fakta yang ada di dunia nyata. Kelas konsep tidak terukur, tetapi sering digunakan untuk berkomunikasi. Contoh dari kelas konsep adalah kinerja.

b. Kelas Event

Adalah sebuah titik dalam waktu yang harus direkam. Sesuatu yang berhubungan dengan event adalah pertanyaan-pertanyaan seperti kapan, siapa, apa, dimana, bagaimana atau mengapa. Contoh dari kelas event adalah pendaratan, interupsi, permintaan, penekanan serta pemesanan.

c. Kelas Organisasi

Adalah kumpulan orang-orang, sumber daya, fasilitas-fasilitas, atau kelompok yang dimiliki pengguna. Contohnya adalah departemen kuntansi, departemen riset, dan sebagainya.

d. Kelas Manusia

Kelas manusia menampilkan peran-peran (*roles*) yang berbeda dalam hubungannya dengan sistem/perangkat lunak yang kita kembangkan. Contoh kelas manusia adalah karyawan, dosen, direktur, serta manager.

e. Kelas Tempat

Kelas tempat meliputi lokasi fisik yang hanya dicatat dalam sistem. Misalnya : gedung, toko, kantor, dan sebagainya.

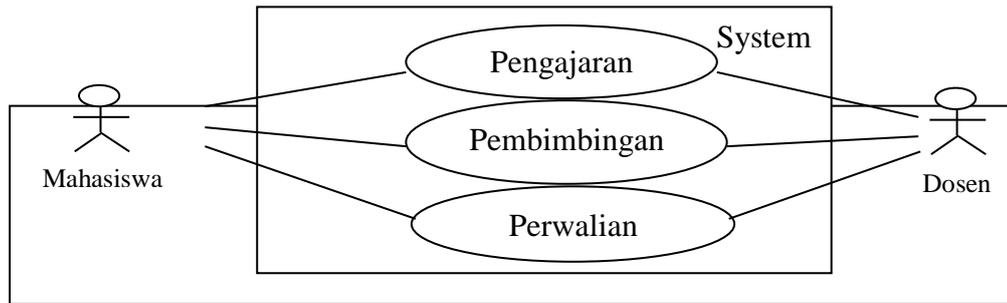
f. Sesuatu yang terukur serta kelas peralatan

Kelas ini meliputi objek fisik serta kelompok objek yang terukur serta peralatan-peralatan yang berinteraksi dengan sistem yang kita kembangkan. Contohnya, mobil adalah sesuatu yang terukur sedangkan sensor tekanan adalah peralatan. (Adi Nugroho; 2010)

12. Use Case Diagram

Use case merupakan unit koheren dari fungsionalitas sistem/perangkat lunak yang yang tampak dari luar dan diekspresikan sebagai urutan pesan-pesan yang dipertukarkan unit-unit sistem dengan satu atau lebih *actor* yang ada diluar sistem. Sebuah *actor* pada dasarnya menggambarkan interaksi pengguna-pengguna sistem dengan sistem /perangkat lunak yang sedang dikembangkan.

Masing-masing *actor* bisa saja berpartisipasi dalam satu atau lebih *use case* yang berinteraksi dengan *use case* dan saling bertukar pesan. Jadi kegunaan *use case* sesungguhnya adalah mendefenisikan suatu bagian perilaku sistem yang bersifat koheren tanpa perlu menyingkapkan struktur internal sistem/perangkat lunak yang sedang dikembangkan. Contoh penggambaran diagram *use case* terlihat pada gambar



Gambar 2.3 Use Case Diagram

Sumber : Adi Nugroho, 2010

Suatu *Use Case* dapat secara sederhana menggabungkan perilaku *use case* yang lainnya sebagai bagian dari perilakunya sendiri. Hal ini dinamakan sebagai relasi. (Adi Nugroho; 2010:34-36).

Tabel 2.2. Relasi-relasi dalam *Use Case*

Relasi	Fungsi	Notasi
Asosiasi (<i>association</i>)	Lintasan Komunikasi antara <i>actor</i> dengan <i>Use Case</i>	_____
Extend	Penambahan perilaku ke suatu <i>Use Case</i>	----->
Generalisasi <i>Use Case</i>	Menggambarkan hubungan antara <i>use case</i> yang bersifat umum dengan <i>use case-use case</i> yang bersifat lebih spesifik	→

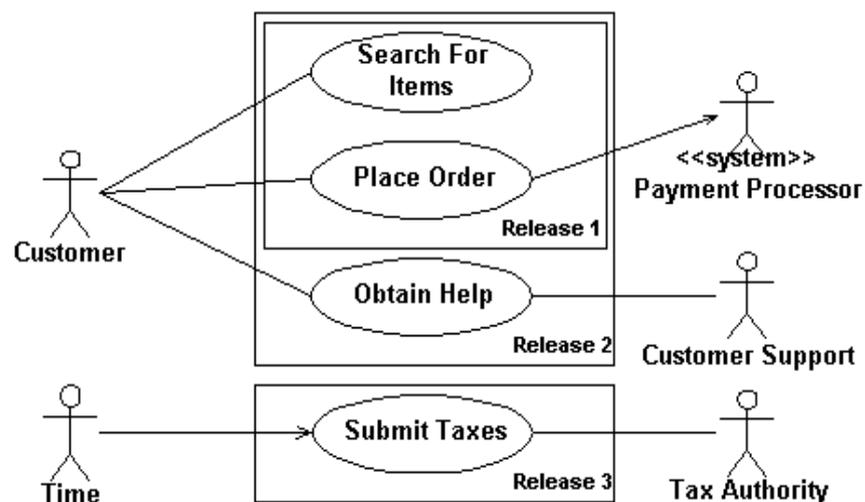
Include	Penambahan perilaku ke suatu <i>use case</i> dasar yang secara eksplisit mendeskripsikan penambahan tersebut	----->
---------	--	--------

Sumber : Adi Nugroho, 2010

Use Case Diagram

Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case*. *Use Case* memiliki dua istilah (Haviluddin, 2011:3)

- System *use case*; interaksi dengan sistem.
- Business *use case*; interaksi bisnis dengan konsumen atau kejadian nyata.



Gambar 2.4 Use Case Diagram

Sumber: Haviluddin, 2011:4

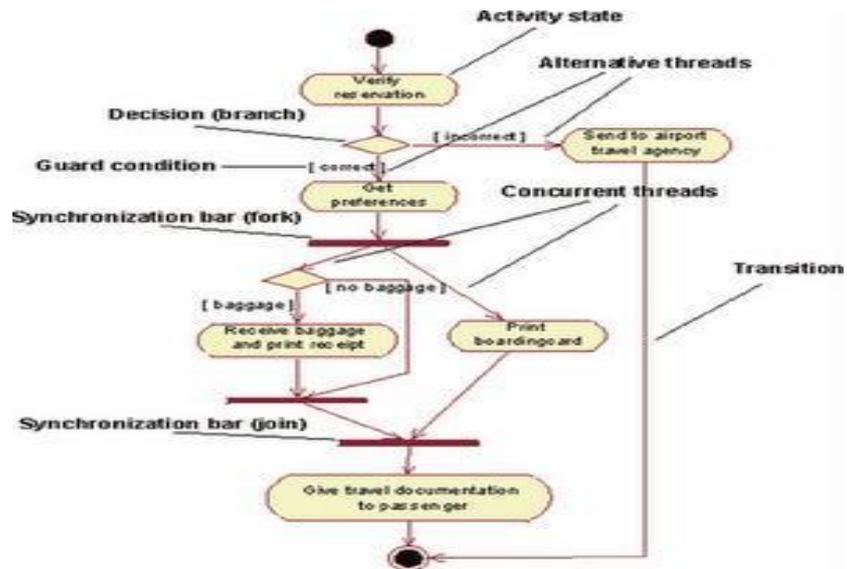
13. Activity Diagram

Diagram aktivitas sesungguhnya merupakan bentuk khusus dari state machine yang bertujuan memodelkan komputasi-komputasi dan aliran-aliran kerja yang terjadi dalam sistem/perangkat lunak yang sedang dikembangkan. Suatu diagram aktivitas memuat di dalamnya *activity state* di mana suatu *activity state* mempresentasikan suatu aktivitas pernyataan dalam dalam suatu prosedur atau kinerja suatu aktivitas dalam suatu aliran kerja.

Suatu diagram aktifitas memuat di dalamnya *activity state* dimana suatu *activity state* mempresentasikan eksekusi pernyataan dalam suatu prosedur atau kinerja suatu aktifitas dalam suatu aliran kerja. Suatu diagram aktivitas mungkin di dalamnya juga memuat suatu *action state*, yang mirip dengan *activity state*, kecuali mereka bersifat atomik dan tidak mengizinkan tansisi-transisi terjadi selama mereka aktif. Suatu diagram aktivitas pada dasarnya bertindak seperti diagram alir (*flow chart*) biasa kecuali doagram aktivitas memungkinkan terjadinya kendali konkuren sebagai tambahan pada kendali sekuensial biasa (sumber : Adi Nugroho; 2010)

Activity Diagram

Menggambarkan aktifitas-aktifitas, objek, state, transisi state dan event. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas (Haviluddin, 2011:4)

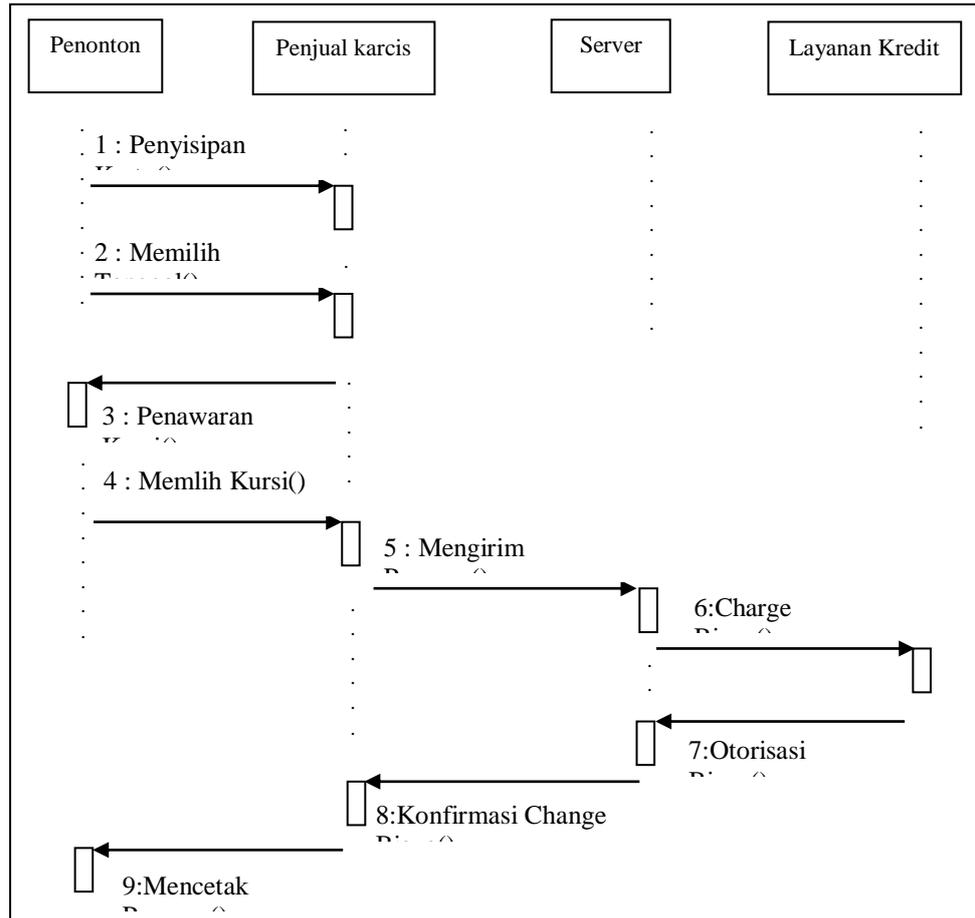


Gambar 2.5 Activity Diagram

Sumber: Haviluddin, 2011:4

14. Sequence Diagram

Sequence Diagram merupakan diagram interaksi yang memperlihatkan urutan-urutan pesan dalam kerangka waktu secara jelas, tetapi tidak secara eksplisit memperlihatkan relasi-relasi yang terjadi pada objek. Dan sequence diagram sangat bermanfaat untuk memperlihatkan skenario-skenario (Adi Nugroho; 2010). Diagram sequence terlihat pada gambar 2.4.



Gambar 2.6 Sequence Diagram

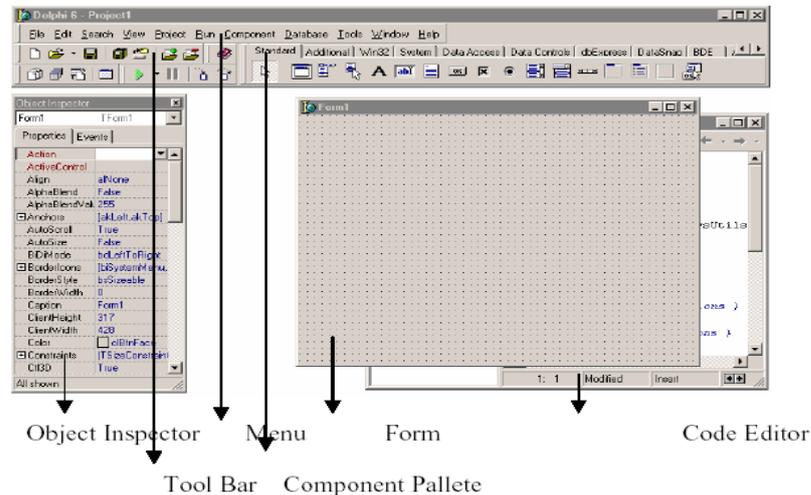
Sumber: Adi Nugroho (2010 : 42)

15. Borland Delphi 7

Borland Delphi merupakan program aplikasi database yang berbasis object Pascal dari Borland. Selain itu, Borland Delphi juga memberikan fasilitas pembuatan aplikasi visual. Borland Delphi memiliki komponen-komponen visual maupun non visual berintegrasi yang akan menghemat penulisan program. Terutama dalam hal perancangan antarmuka grafis (*Graphical User Interface*), kemampuan Borland Delphi untuk menggunakan Windows API (*Application Programming Interface*) ke dalam komponen-komponen visual menyebabkan pemrograman Borland Delphi yang bekerja dalam lingkungan Windows menjadi lebih mudah.

16. *Integrated Development Environment*

Tampilan bidang kerja yang disebut dengan IDE (*Integrated Development Environment*) Delphi 7 bisa dilihat pada gambar 2.3 di bawah ini. IDE ini secara garis besar terdiri atas tiga bagian utama, yaitu Window Utama, Object Inspector dan Editor.

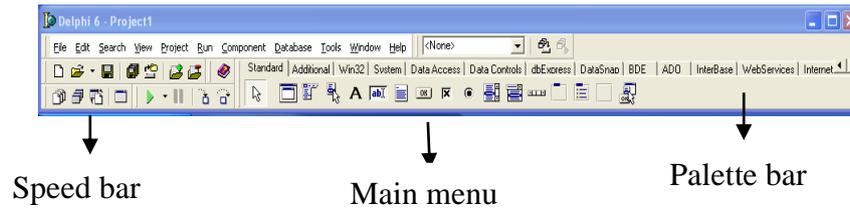


Gambar 2.7 Tampilan IDE Delphi 7.0

Sumber : Abdul Kadir, 2009

17. *Main menu, Speed Bar dan Palette Bar*

Di dalam *main menu*, terdapat fasilitas-fasilitas yang disediakan oleh Delphi untuk mengontrol serta menangani masalah yang berhubungan dengan pekerjaan kita (contohnya menu untuk menyimpan, menu untuk meng-*compile* dan masih banyak lagi). Kegunaan *speed bar* seperti *main menu*, tapi di *speed bar* berupa *icon* sehingga kita cukup mengklik saja untuk menjalankan fasilitas yang kita kehendaki. *Palette bar* adalah tempat *object-object* yang kita gunakan untuk menyusun sebuah program.

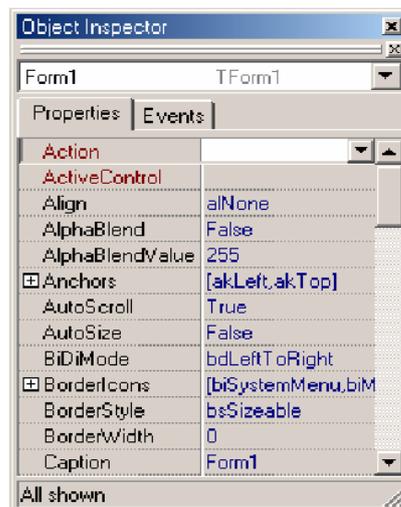


Gambar 2.8 Speed Bar, Main Menu dan Palette Bar

Sumber : Abdul Kadir 2009

18. Object Editor

Object inspector adalah sarana pengaturan objek yang kita pasang pada form, atau form itu sendiri. Dua hal penting yang bisa kita setel pada komponen adalah Properti dan event. Properti adalah yang terkait dengan sifat komponen seperti ukuran, warna dan sebagainya. Sedangkan event adalah kejadian atau peristiwa yang kita inginkan terpasang pada komponen tersebut kaitannya dengan proses pemakaian. Contoh *event* misalnya klik, klik ganda, *drag* (geser), *drop* dan sebagainya.

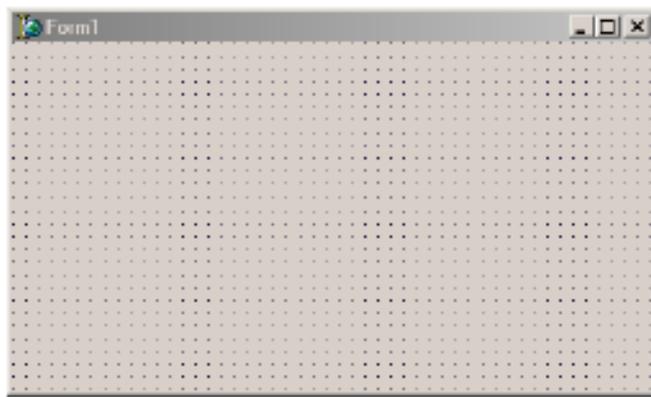


Gambar 2.9 Object Inspector

Sumber :, Abdul Kadir, 2009

19. *Form*

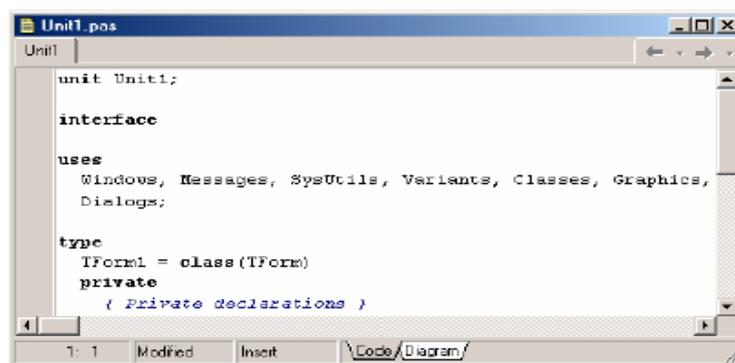
Form adalah bahan dasar yang akan menjadi jendela aplikasi kita. Pada form terdapat tiga tombol kontrol, yaitu *Minimize*, *Maximize/Restore* dan *Close*. Terdapat juga caption bar tempat kita menempatkan judul Form (yang kelak menjadi judul window) dan icon. Pembatas form juga bisa diubah ukurannya dengan cara drag (geser)-drop. Pada form kita bisa meletakkan komponen-komponen yang kita perlukan dalam suatu *User Interface*.



Gambar 2.10 Form
Sumber : Abdul Kadir, 2009

20. *Code Editor*

Code Editor adalah tempat kita menuliskan program dalam bahasa *Object Pascal*. Secara default *Code Editor* ini terletak di belakang Form Editor.



Gambar 2.11 Code Editor
Sumber : Abdul Kadir, 2009

BAB III

ANALISIS DAN PERANCANGAN

1. Analisis

Proses pembentukan suara dari suatu kata dilakukan dengan cara penggabungan per suku kata. Supaya lebih mudah penulis menjelaskannya dalam contoh kata dalam bentuk bahasa Inggris karena secara konsep bentuk kata nya sama dalam setiap fonem. Misalnya kata “meat” maka untuk menghasilkan suara dari kata tersebut diperlukan suara “me” dan “at”. Berdasarkan data tersebut *database fonem* dilakukan dengan mendata kemungkinan-kemungkinan kombinasi dari suku kata yang muncul dalam suatu kata. Berikut ini beberapa kemungkinan fonem-fonem dalam bahasa Inggris :

1. KVVK : Meat (me – at , me - et,)
2. KKVK : Crab (cra-b, g - rab,...)
3. KVKKVK : Banned (bann - ed, band-)
4. VKKVKV : Effect (Ef – Fect , Af - Fect)
5. KKVKK : Crash (Cra – sh , Cla – sh)

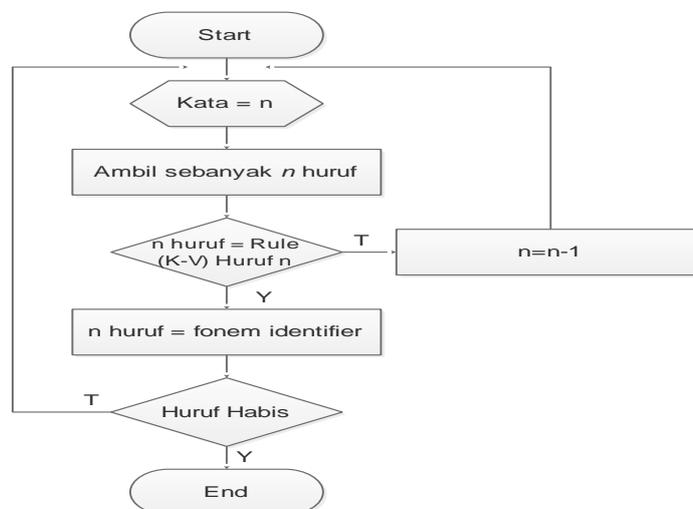
Setelah semua fonem ditentukan maka dikelompokkan menurut jumlah huruf dan susunan konsonan-vokalnya. Kemudian diurutkan secara alfabet, hal ini dilakukan agar terbentuk pola yang teratur sehingga memudahkan dalam pemrogramannya, huruf konsonan dan vokal digunakan untuk membagi setiap huruf jika didalam kata terdapat karakter vokal tetapi jika tidak terdapat huruf vokal tidak menjadi permasalahan seperti kata Sky, Rhythms, Myth, Fly, Dry, Shy, Gym.

2. Perancangan Konversi Teks

Kalimat yang diketik melalui keyboard akan dipecah menjadi per kata kemudian akan dipecah lagi menjadi per suku kata, pada proses ini *diphone concatenation* dilakukan. *diphone concatenation* digunakan untuk memecah kalimat menjadi per bagian kata dapat dilakukan dengan melihat tanda spasi antar kata yang satu dengan kata yang lain. Sedangkan untuk memecah kata menjadi suku kata memerlukan algoritma khusus.

Misal kalimat: “i am eating” jika dipecah per suku kata menjadi “i – am – eat - ing”. Pada bahasa Inggris satu suku kata dapat terdiri dari 1 huruf sampai dengan 3 huruf.

Berdasarkan karakteristik fonem-fonem tersebut dapat digunakan untuk mengenali fonem dari suatu kata. Untuk fonem yang berpola KKVK, KKV dan VKK menggunakan *look up table Diphone* karena jenis fonem ini memerlukan perlakuan khusus disebabkan komposisi hurufnya tidak beraturan. Pada gambar 3.1. merupakan flowchart konverter teks ke fonem.



Gambar 3.1 Flowchart Konversi Teks ke Fonem

Penjelasan dari flowchart pada gambar 3.1. adalah sebagai berikut:

1. Nilai n , diasumsikan nilai $n = 4$.

Jumlah huruf maksimum suatu fonem adalah 4 karakter. Dimulai dari nilai maksimum untuk menghindari kesalahan pengelompokan jenis fonem dengan nilai yang kurang dari 4. Oleh sebab itu nilai maksimum diprioritaskan terlebih dahulu.

2. Ambil sebanyak N huruf dari belakang.

Hal ini dilakukan untuk memprioritaskan jenis akhiran yang terdiri dari 4 huruf, seperti “mel-ting” dan sebagainya.

3. N huruf = rule (K-V) untuk N huruf?

Huruf yang sudah diambil sebanyak N buah susunan konsonan dan vokalnya akan dibandingkan dengan aturan untuk yang berjumlah N buah huruf. Jika tidak ada yang sama maka N huruf tersebut bukan merupakan fonem. $N-1$ Kurangi satu jumlah huruf yang akan diambil dan kembali lakukan lagi langkah 1 diatas. Jika N huruf sama dengan aturan untuk N huruf maka lakukan langkah 4 dibawah.

4. N huruf=Fonem.

Jika sama, maka N huruf tersebut adalah sebuah fonem. Geser pointer ke kiri. Dilakukan pergeseran tanda pointer 4 baris untuk bagian huruf sebelah kiri dari fonem tersebut.

5. Huruf habis?

Jika huruf pada bagian kiri sudah tidak ada maka proses konversi dari seluruh kalimat sudah selesai. Jika masih ada huruf pada bagian kiri maka proses dimulai lagi dari langkah 1 diatas, tetapi dengan nilai pointer yang sudah digeser kekiri 4 baris pada langkah 4 diatas.

Berikut ini adalah contoh dari proses konversi kata “ringan” menjadi per fonem “ri-ngan” mel-ting.

1. N=4. Pointer diletakkan pada huruf keempat dari kanan (huruf “n” tengah).
2. Diambil 4 huruf kekanan dimulai dari titik pointer sehingga didapatkan kombinasi karakter “ting”.
3. Susunan konsonan-vokal dari kombinasi ini yaitu KVKK akan dicocokkan dengan aturan untuk 4 huruf. Ternyata hasil dari pencocokan ini adalah sesuai.
4. Jika sesuai maka kombinasi huruf ini dinyatakan sebagai sebuah fonem. Geser pointer kekiri sebanyak 4 karakter.
5. Karena huruf sebelah kiri fonem “ting” masih ada maka proses akan dimulai lagi pada langkah 1.
6. N=4. Pointer diletakkan pada karakter keempat dari kanan setelah fonem tadi (karena melebihi batas karakter pertama maka pointer menunjuk karakter”[]”).
7. Diambil 4 huruf kekanan dimulai dari titik pointer sehingga didapatkan kombinasi karakter “[]mel”.
8. Susunan konsonan-vokal dari kombinasi ini yaitu []KVK akan dicocokkan dengan aturan untuk 4 huruf. Ternyata hasil dari pencocokan ini tidak ada yang sesuai. N-1, jumlah n karakter yang akan diambil untuk proses berikutnya dikurangi satu sehingga berjumlah 3 huruf.
9. N=3. Pointer diletakkan pada karakter ketiga dari kanan setelah fonem tadi (karena melebihi batas karakter pertama maka pointer menunjuk karakter”[]”).

10. Diambil 3 huruf kekanan dimulai dari titik pointer sehingga didapatkan kombinasi karakter “[]mel”.
11. Susunan konsonan-vokal dari kombinasi ini yaitu []KVK akan dicocokkan dengan aturan untuk 3 huruf. Ternyata hasil dari pencocokan ini tidak ada yang sesuai. N--,jumlah n karakter yang akan diambil untuk proses berikutnya dikurangi satu sehingga berjumlah 2 huruf.
12. N=2. Pointer diletakkan pada karakter kedua dari kanan setelah fonem tadi (huruf “m” pertama).
13. Diambil 3 huruf kekanan dimulai dari titik pointer sehingga didapatkan kombinasi karakter “mel”.
14. Susunan konsonan-vokal dari kombinasi ini yaitu KVK akan dicocokkan dengan aturan untuk 2 huruf. Ternyata hasil dari pencocokan ini adalah sesuai.
15. Jika sesuai maka kombinasi huruf ini dinyatakan sebagai sebuah fonem. Geser pointer kekiri sebanyak 4 karakter.
16. Karena huruf di sebelah kiri fonem “ri” tidak ada maka proses akan berakhir.

Langkah diatas adalah langkah mengenali kata yang akan diucapkan, secara konsep pengenalan kata menggunakan *diphone concatenation* tidak berbeda untuk semua kata dalam tatanan bahasa, yang berbeda hanya bagaimana proses pencacahan setiap kata.

3. Analisa Fonem Voice Command

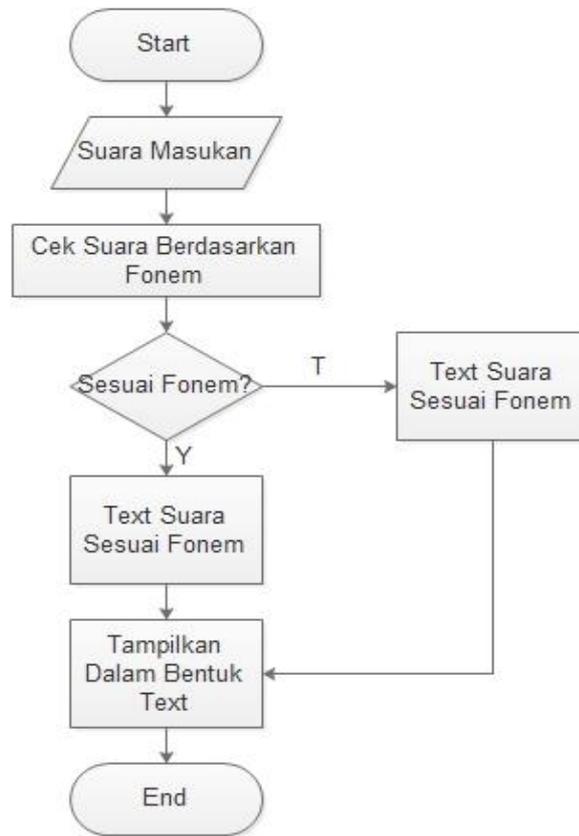
Voice command atau biasa disebut perintah suara merupakan salah satu bagian dari yang namanya *voice recognition*, pada *voice command* proses yang terjadi adalah *user* mengucapkan sebuah kata yang merupakan kata kunci untuk

memerintah suatu proses, untuk mengenali kata tersebut maka kata tersebut harus dikonversi kedalam bentuk fonem dengan struktur fonem sebagai berikut:

```
Phonemes: array[1..49] of String = (
  '-', '!', '&', ',', '!', '?', '_',
  '1', '2', 'aa', 'ae', 'ah', 'ao', 'aw',
  'ax', 'ay', 'b', 'ch', 'd', 'dh', 'eh',
  'er', 'ey', 'f', 'g', 'h', 'ih', 'iy',
  'jh', 'k', 'l', 'm', 'n', 'ng', 'ow',
  'oy', 'p', 'r', 's', 'sh', 't', 'th',
  'uh', 'uw', 'v', 'w', 'y', 'z', 'zh'
)
```

Sebagai contoh adalah WORD maka jika dikonversi kedalam bentuk fonem maka kata yang muncul adalah “ W ER D” bukan “WORD” hal ini dikarenakan *pronoun* ketika terucap beda dengan kata.

Berikut adalah flowchart voice command yang penulis rancang berdasarkan analisa yang penulis dapat.



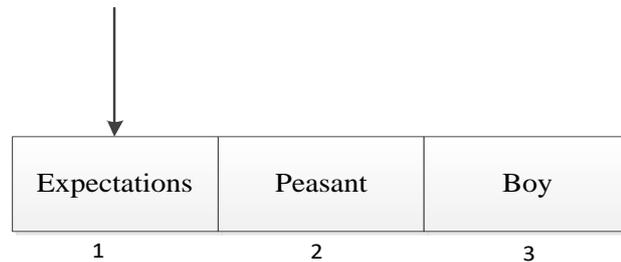
Gambar 3.2 Flowchart Pengenalan Suara

Berikut adalah keterangannya

1. Suara masukan berupa suara yang diucapkan secara langsung dengan bantuan *microphone* atau *headset*.
2. Suara yang diucapkan kemudian di proses untuk dikenali kata dari suara ucapan.
3. Kemudian suara yang ada di dibandingkan dengan suara yang diinginkan apakah sesuai atau tidak, jika sesuai maka suara akan memanggil program tertentu tetapi jika tidak hanya menampilkan text dari suara yang memiliki kemiripan fonem.
4. **Analisa String Matching Dalam Proses Konversi Suara**

Proses pencocokan string pada setiap kata yang diproses menggunakan *diphone concatenation* menggunakan *phonetic string matching*, proses *phonetic*

string matching sebenarnya tidak berbeda dengan proses pencocokan biasa, bedanya hanya pada pemeriksaan karakter vokal dan konsonan, berikut adalah analisisnya:



Gambar 3.3 Susunan Kalimat Per Kata Welcome

Pada gambar 3.2 tampak sebuah kalimat yang terdiri dari 3 buah susunan kata yang akan dikonversi kedalam bentuk suara, untuk kalimat pertama yang akan dicek adalah kata “Expectations”, berikut adalah analisisnya.

ArrVokal = {A,I,U,E,O}

VarKata = Expectations

VarKataFonem=""

nMax = Length(varkata)

for I = 1 to nMax

xkar=Mid(varkata,I,1)

if xkar = ArrVokal then

fonem= vokal(V)

else

fonem= konsonan(K)

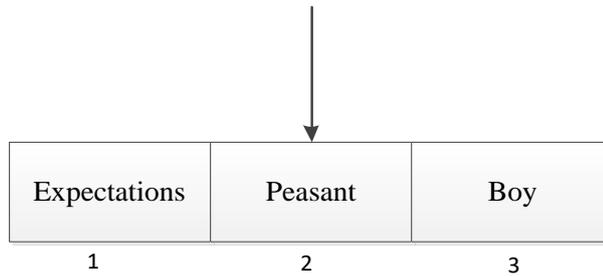
endif

VarKataFonem= VarKataFonem + Fonem

next

VarKataFonem=KVKKVKV

Untuk proses kata pertama sudah diselesaikan kemudian dilanjutkan dengan kata berikutnya, seperti gambar dibawah ini.



Gambar 3.4 Susunan Kalimat Per Kata To

Pada gambar 3.3 tampak sebuah kalimat yang terdiri dari 3 buah susunan kata yang akan dikonversi kedalam bentuk suara, untuk kalimat kedua yang akan dicek adalah kata “Peasant”, berikut adalah analisisnya.

ArrVokal = {A,I,U,E,O}

VarKata = Peasant

VarKataFonem=""

nMax = Length(varkata)

for I = 1 to nMax

xkar=Mid(varkata,I,1)

if xkar = ArrVokal then

fonem= vokal(V)

else

fonem= konsonan(K)

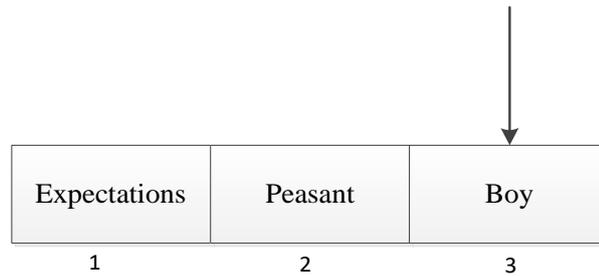
endif

VarKataFonem= VarKataFonem + Fonem

next

VarKataFonem=KV

Untuk proses kata kedua sudah diselesaikan kemudian dilanjutkan dengan kata berikutnya, seperti gambar dibawah ini:



Gambar 3.5 Susunan Kalimat Per Kata Prima

Pada gambar 3.4 tampak sebuah kalimat yang terdiri dari 3 buah susunan kata yang akan dikonversi kedalam bentuk suara, untuk kalimat ketiga yang akan dicek adalah kata “Boy”, berikut adalah analisisnya.

ArrVokal = {A,I,U,E,O}

VarKata = Boy

VarKataFonem=""

nMax = Length(varkata)

for I = 1 to nMax

xkar=Mid(varkata,I,1)

if xkar = ArrVokal then

fonem= vokal(V)

else

fonem= konsonan(K)

endif

VarKataFonem= VarKataFonem + Fonem

next

VarKataFonem= KKVKV

5. Proses Teks Menjadi Suara

Untuk implementasi perubahan teks menjadi suara dapat dilihat pada proses dibawah ini dimana penerapan string matching akan memeriksa kata yang ada didalam array data contohnya seperti dibawah ini:

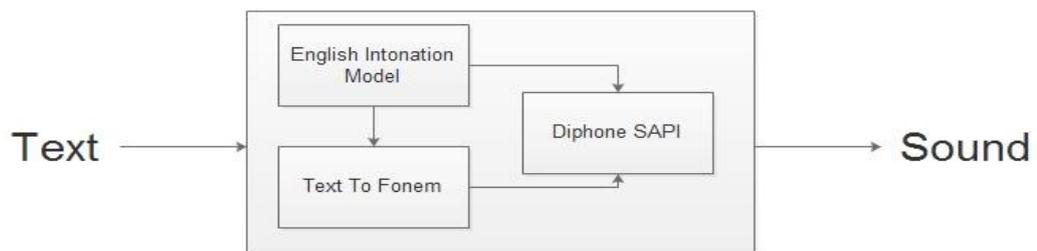
Tabel 3.1 Array Data

Eat	Drink	Called	Oil	Then	Hunting
Sound	Cup	Mouse	Computer	Ratio	Paper

Dari tabel array data diatas dilakukan proses *phonetic string matching* seperti pada sub bab 3.1.2 contohnya adalah kata ratio, proses *phonetic string matching* akan mencocokkan kata per kata dan ketika ketemu maka hasilnya akan berupa suara dan proses pengubahannya seperti pada sub bab 3.1.4 berikut.

6. Proses Diphone Bekerja

Diphone merupakan database yang berisi kata-kata dalam bahasa inggris yang nantinya akan dipergunakan untuk memeriksa kata yang dimasukkan dan membandingkan dengan yang ada didalam diphone, jika terdapat kata maka kata tersebut akan dikonversi menjadi suara, berikut diagram yang menerangkan penggunaan diphone.

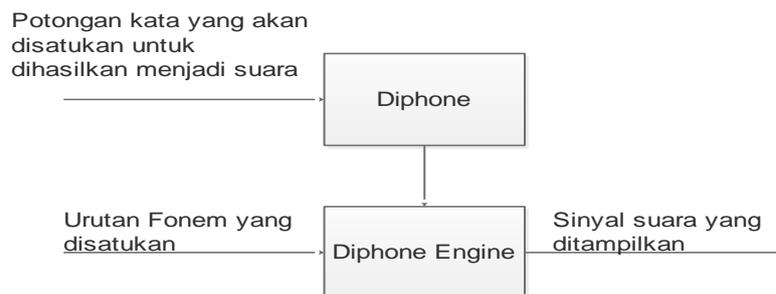


Gambar 3.6 Proses Diphone

Pada dasarnya *diphone* itu merupakan database kata Inggris dan yang mengelola informasi dari database adalah *Microsoft Speech Application Programming Interface* yang berupa tool sederhana.

Untuk proses perubahan teks menjadi suara dilakukan dengan fonem, Konverter fonem ke ucapan berfungsi untuk membangkitkan sinyal ucapan berdasarkan kode-kode fonem yang dihasilkan dari proses sebelumnya. Teknik yang sering digunakan pada bagian ini adalah teknik *diphone concatenation* yang harus didukung oleh suatu database *diphone* yang berisi rekaman segmen-segmen ucapan yang berupa diphone (gabungan dua buah fonem).

Ucapan dalam suatu bahasa dibentuk dari satu set bunyi yang mungkin berbeda untuk setiap bahasa, oleh karena itu setiap bahasa harus dilengkapi dengan diphone database yang berbeda, berikut adalah kinerja *diphone* dengan fonem yang penulis rancang:



Gambar 3.7 Diphone Fonem

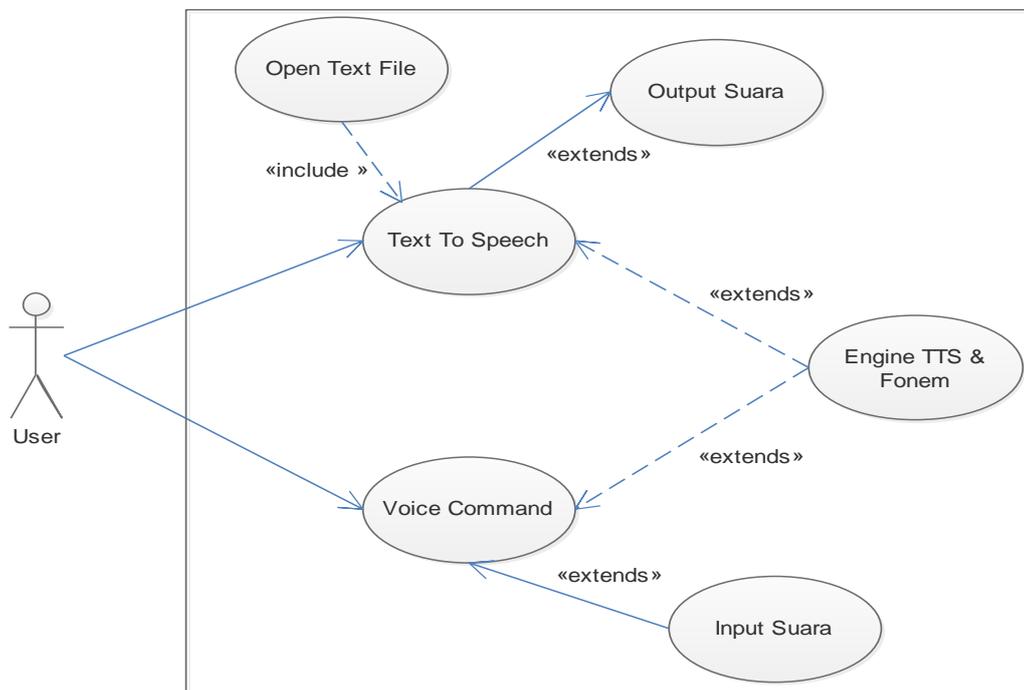
Berikut adalah penjelasannya:

1. Potongan kata yang akan disatukan untuk dihasilkan menjadi suara merupakan kata yang dimasukkan akan dibagi menjadi beberapa objek untuk kemudian diproses dengan menggunakan *diphone*.

2. Pada diphone engine urutan fonem yang dipisah kemudian disatukan menjadi bentuk asli untuk kemudian diproses hingga menghasilkan suara.
3. Sinyal suara yang ditampilkan merupakan bentuk suara hasil diphone yang memproses suara.

7. Use Case Dan Activity Diagram Serta Sequence Diagram

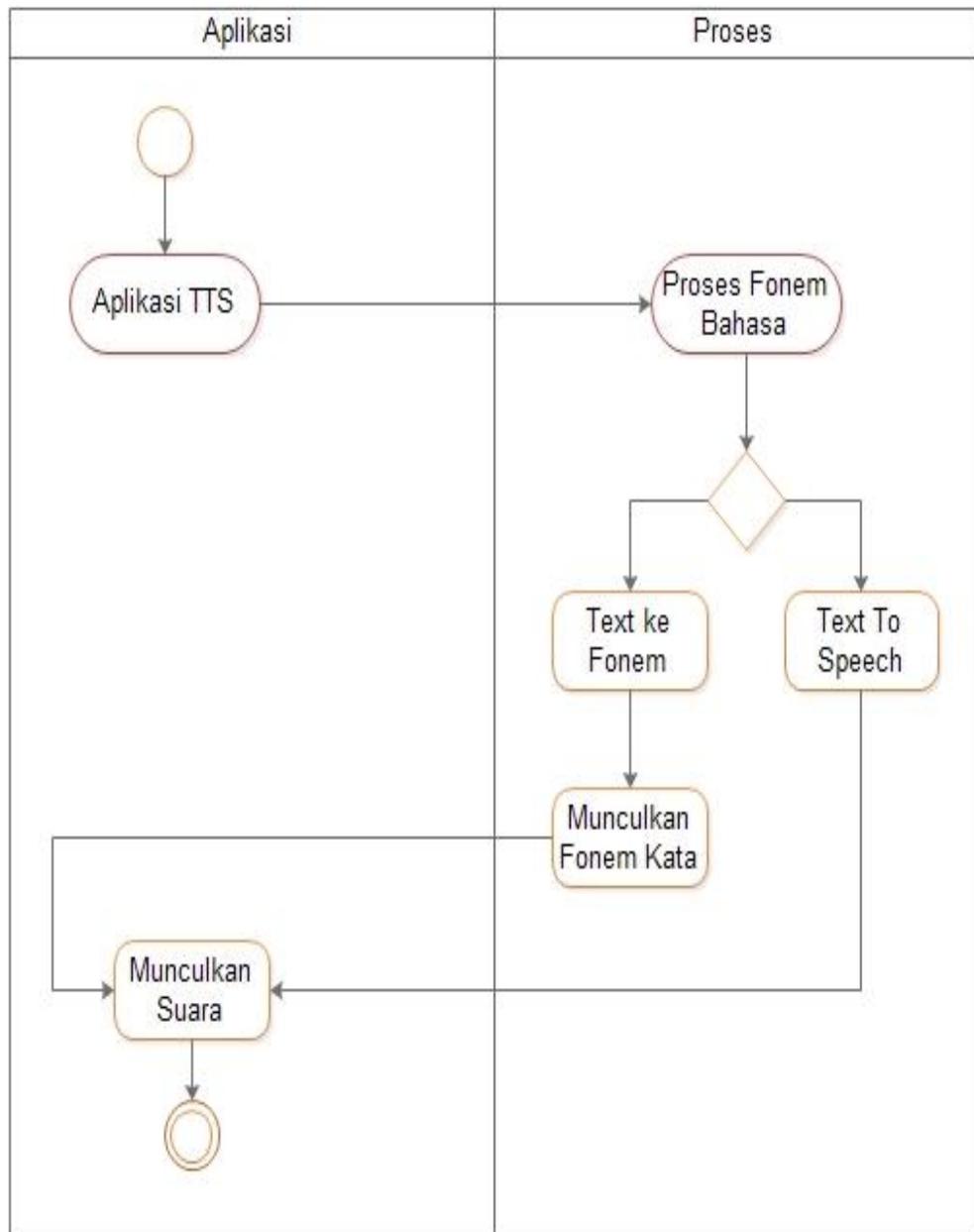
Use Case dan Activity diagram digunakan untuk menggambarkan system secara keseluruhan dalam bentuk diagram yang sistematis dan sederhana, adapun bentuk use case dan activity diagram yang penulis buat adalah sebagai berikut:



Gambar 3.8 Use Case Diagram Konversi Teks ke Suara dan Voice Command

Use case diagram diatas menggambarkan bagaimana interaksi pemakai aplikasi dengan aplikasi yang dirancang, pada dasarnya user hanya mengakses aplikasi dan memasukkan teks bahasa Inggris, kemudian system akan merubahnya menjadi suara bahasa Inggris.

Selain *Use Case* terdapat sebuah *activity diagram* yang dibuat untuk menggambarkan aktivitas aplikasi, perhatikan *activity diagram* dibawah ini:



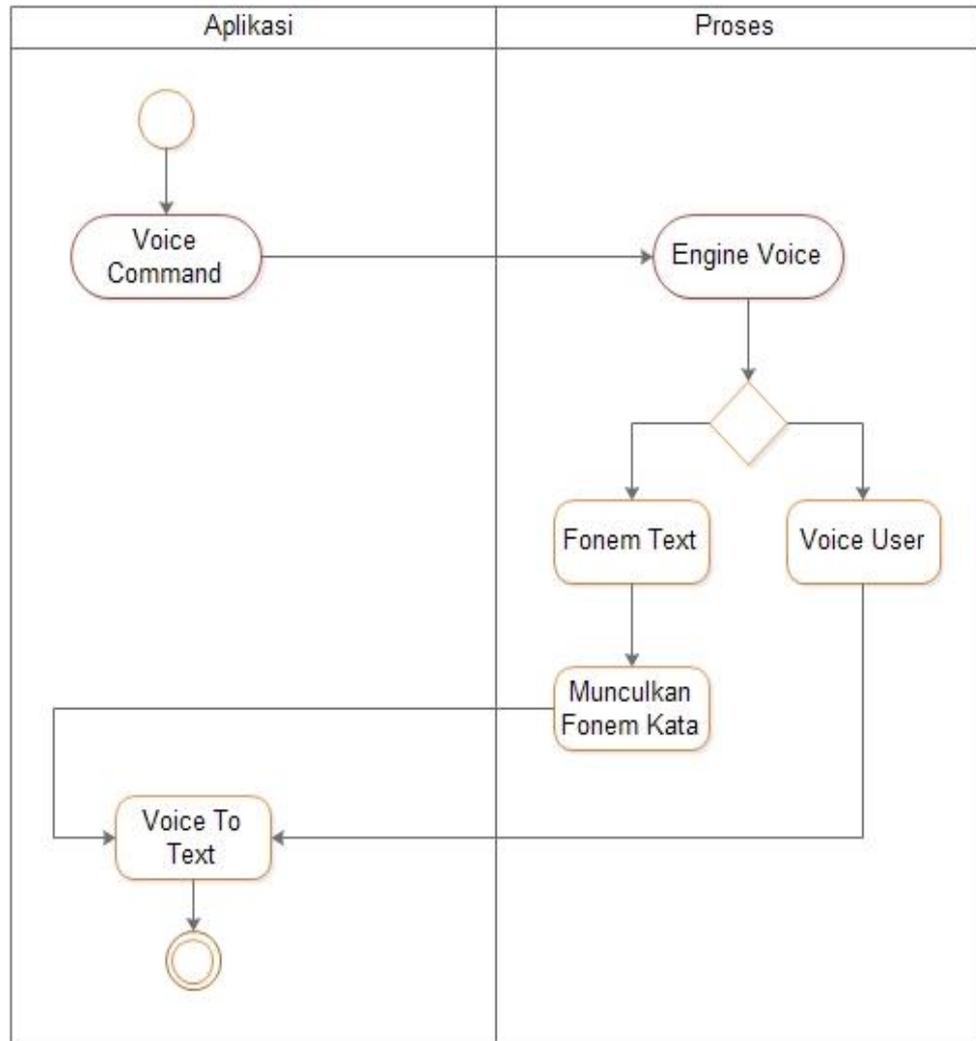
Gambar 3.9 Activity Diagram Konversi Teks ke Suara

Berikut adalah penjelasannya:

1. Aplikasi dijalankan dan kemudian mengakses fonem bahasa.
2. Pada proses fonem bahasa terdapat 2 (dua) proses yaitu *text to speech* dan *text ke fonem*.

3. *Text to speech* mengeluarkan suara sedangkan *text ke fonem* merupakan fungsi yang menampilkan fonem kata.

Selain *activity diagram* konversi teks ke suara berikutnya adalah *activity diagram* konversi suara kedalam bentuk teks, berikut *activity diagram*nya:



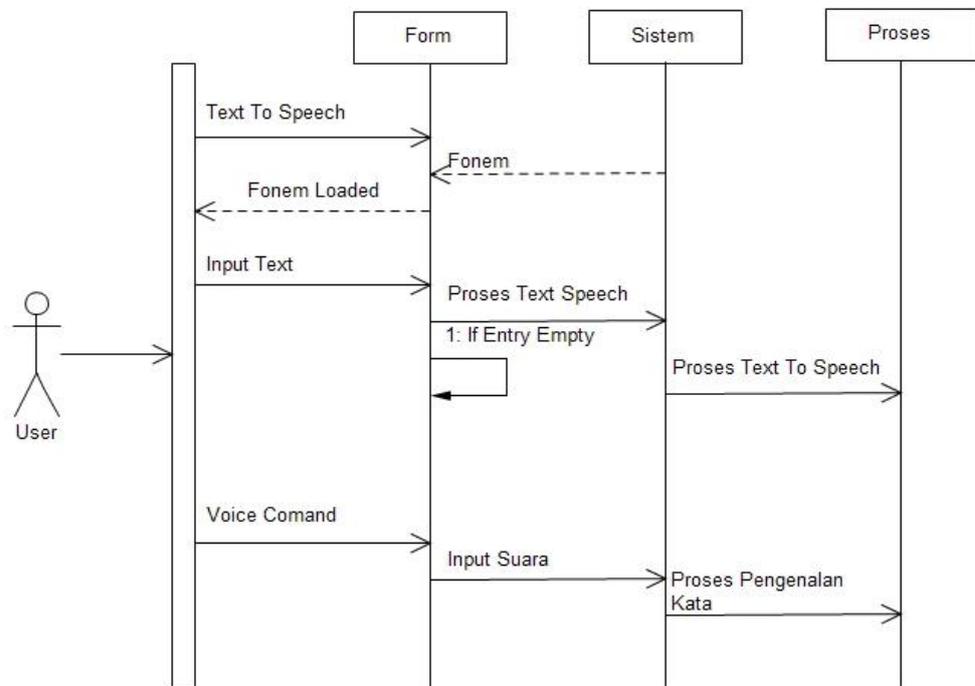
Gambar 3.10 Activity Diagram Konversi Voice Command

Berikut adalah penjelasannya

1. Aplikasi dijalankan dan kemudian mengakses engine voice.
2. Pada proses *engine voice* terdapat fungsi *fonem text* dan *voice user*.

- Fonem text merupakan fungsi yang menampilkan kata yang berhasil dikenali ketika proses input suara dilakukan.

Berikutnya adalah sequeunce diagram dari aplikasi yang penulis rancang, berikut adalah desainnya



Gambar 3.11 Sequence Diagram Aplikasi

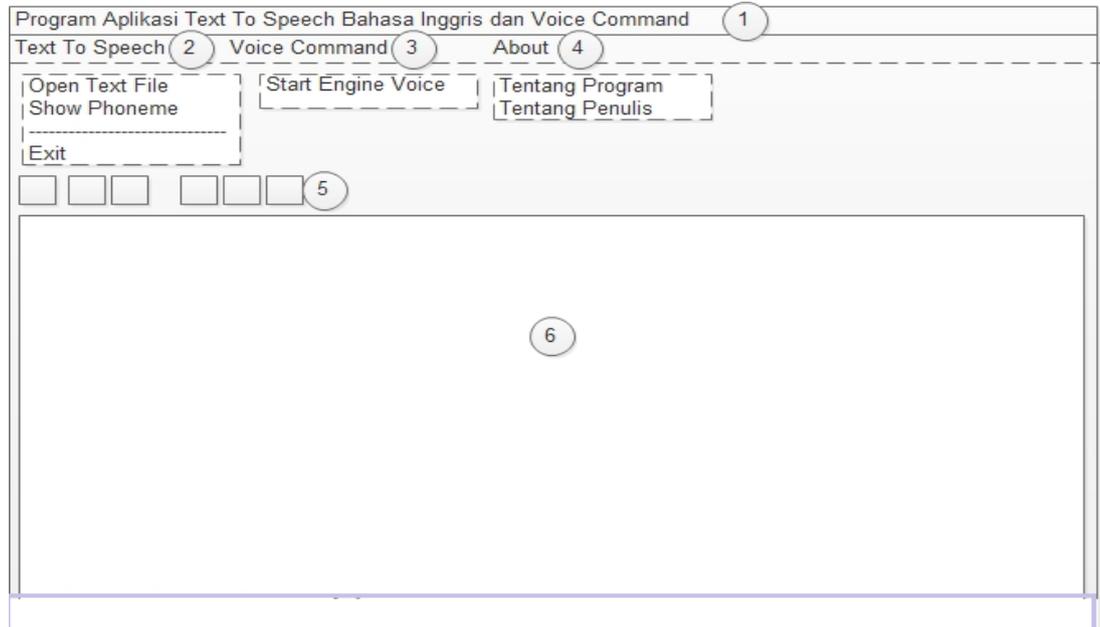
Berikut adalah penjelasannya

- Pengguna mengakses *text to speech* yang kemudian bisa mengakses fonem dan kemudian berikutnya adalah memasukkan text yang kemudian di proses untuk di konversi menjadi suara.
- Proses voice command dilakukan dengan memproses suara yang direkam dengan head set ataupun microphone dan suara yang ada akan dikenali kata nya.

8. Perancangan Aplikasi

Perancangan interface merupakan rancang bangun dari interaksi pemain sistem (administrasi) dengan komputer. Interaksi ini dapat berupa proses

penginputan data ke sistem, pengupdatean data dan menjalankan aplikasi. Dalam mengimplementasikan sistem ini, penulis menggunakan satu form utama berisi beberapa alternatif atau pilihan tombol–tombol untuk melakukan proses konversi menjadi suara ataupun untuk mengambil file yang yang berisi bahasa Inggris, berikut adalah rancangan formnya.



Gambar 3.12 Desain Aplikasi Utama

Adapun keterangannya sebagai berikut:

1. Objek ini digunakan untuk judul aplikasi.
2. Menu *text to speech* digunakan untuk menampilkan sub menu *open text file* dan *exit*.
 - a. *Open Text File*
Sub menu ini digunakan untuk membuka file text dari *harddisk*.
 - b. *Show Phoneme*
Sub menu ini digunakan untuk menampilkan informasi fonem dari proses *text to speech*.

c. *Exit*

Sub menu ini digunakan untuk menutup aplikasi.

3. Menu *voice command* digunakan untuk menampilkan sub menu *Start Engine Voice*.
 - a. Sub menu *Start Engine Voice* digunakan untuk mengaktifkan proses engine pengenalan suara.
4. Menu *about* digunakan untuk menampilkan sub menu tentang aplikasi dan tentang penulis.
 - a. Tentang Program

Sub menu ini digunakan untuk menampilkan info dari aplikasi yang buat.
 - b. Tentang Penulis

Sub menu ini digunakan untuk menampilkan info pembuat aplikasi.
5. Objek ini merupakan objek *toolbar* yang digunakan untuk mengakses beberapa proses seperti *play sound, pause, stop*.
6. Objek ini digunakan untuk menampilkan *text* bahasa inggris yang diambil dari *file text* ataupun yang diketikkan langsung oleh pengguna.
7. Menampilkan info singkat penulis.

BAB IV

IMPLEMENTASI PERANGKAT LUNAK

1. Kebutuhan Perangkat Lunak

Dalam sebuah sistem agar dapat berjalan seperti yang diinginkan karena sistem komputerisasi tidak dapat dipisahkan antara *Hardware* dan *Software*. Berikut adalah spesifikasi *hardware* yang penulis gunakan untuk merancang aplikasi *Text To Speech*:

- a. Intel Core i3 2370
- b. Harddisk 500 GB
- c. RAM 2 GB
- d. Keyboard
- e. Mouse
- f. HeadsetMicrophone/Speaker

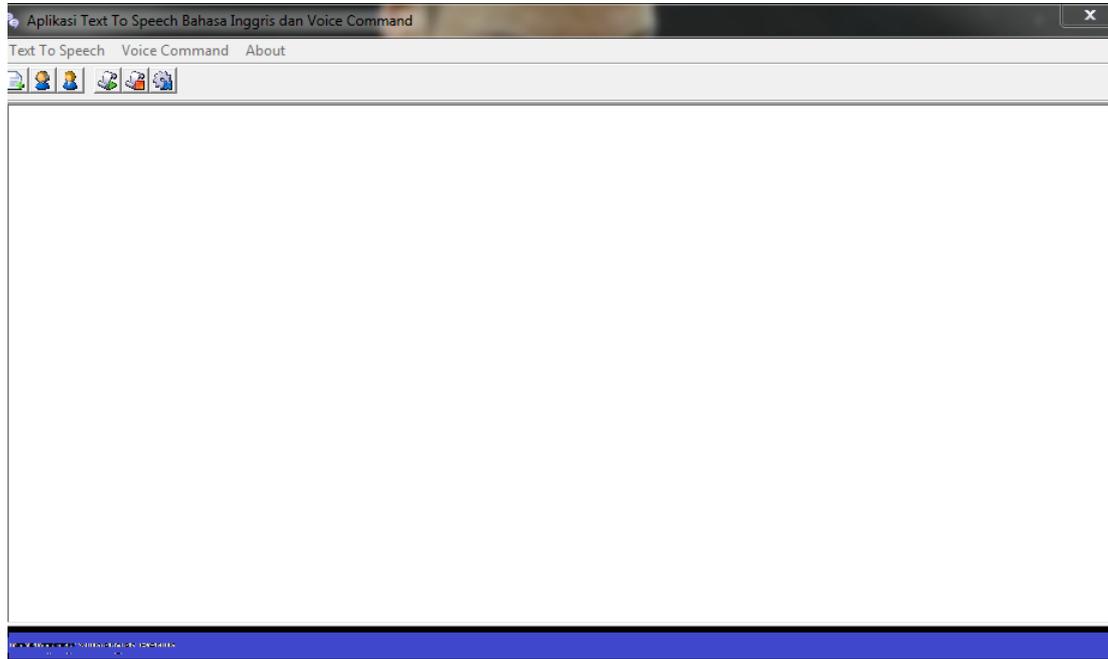
Kemudian berikut adalah spesifikasi *Software* yang penulis gunakan untuk merancang aplikasi *Text To Speech*:

- a. Sistem Operasi *Windows 7 Ultimate*
- b. *Borland Delphi 7*
- c. *Speech Engine Microsoft Speech Application Programming Interface (SAPI)*

2. Implementasi

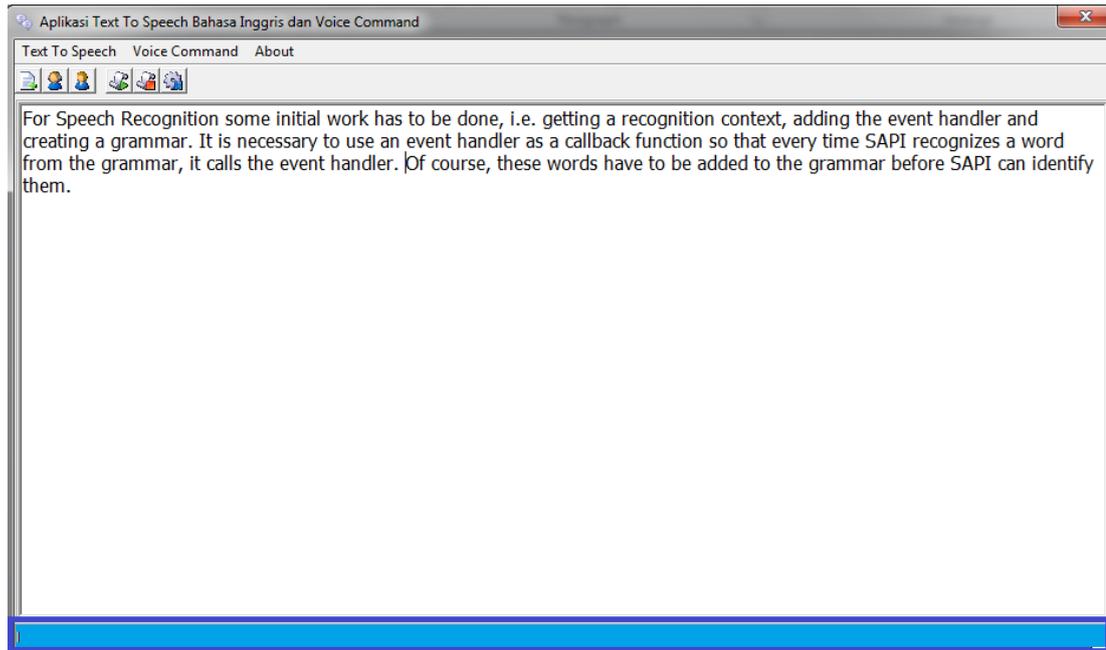
Pengujian sistem dilakukan dengan menjalankan program yang sudah dirancang dengan menggunakan bahasa pemrograman *borland delphi 7*, pad sistem yang dirancang terdiri dari 2 (dua) buah form yaitu form utama dan form *voice command*.

Tampilan utama aplikasi *text to speech* bahasa Inggris, tampilan yang penulis rancang dibuat semudah mungkin dalam hal penggunaan aplikasi dan memiliki *interface* yang cukup menarik, berikut adalah tampilan utama aplikasinya.



Gambar 4.1 Tampilan Aplikasi Utama

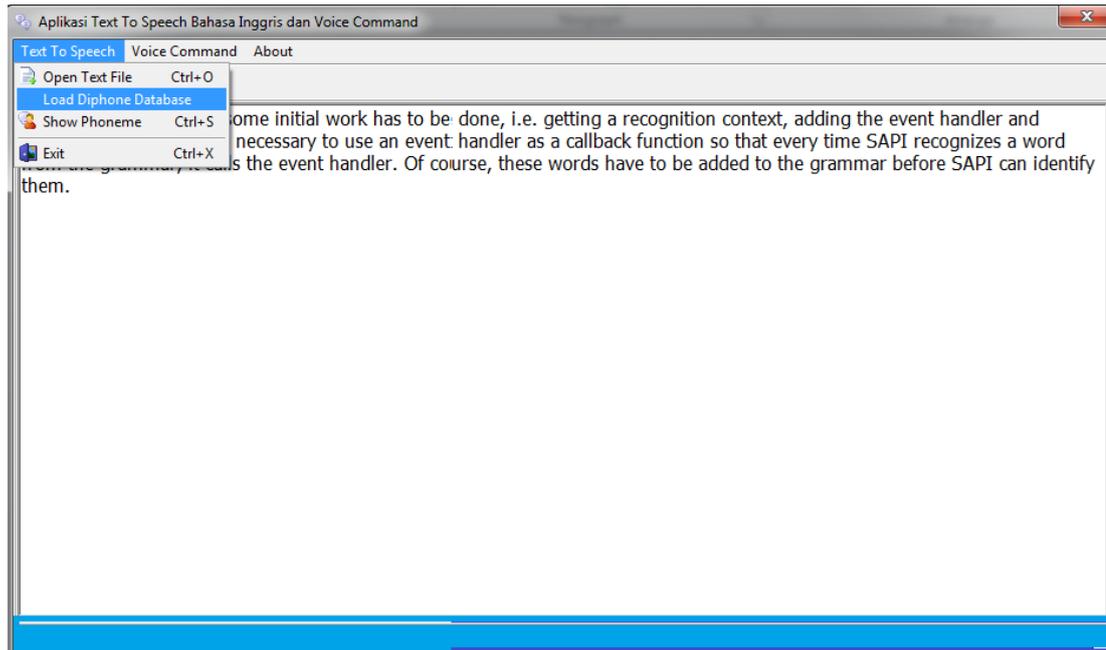
Aplikasi utama diatas yang digunakan untuk memasukkan teks bahasa Inggris yang nantinya teks tersebut akan dikonversi menjadi suara, untuk memasukkan teks bahasa Inggris bisa dilakukan dengan dua cara, yang pertama dengan mengambil file text yang terdapat pada komputer dan yang kedua dengan mengetikkan secara langsung, untuk pengujian pertama ini penulis memasukkan kata bahasa Inggris dengan kalimat seperti tampak pada gambar dibawah ini:



Gambar 4.2 Tampilan Aplikasi Dengan Isian Teks Input Manual

Kalimat diatas merupakan kalimat yang penulis masukkan secara manual dengan mengetikkan secara langsung, untuk kalimat yang dimasukkan dalam bahasa Inggris sebaiknya memperhatikan kaidah bahasa Inggris yang baik dan benar terutama kalimat baku, kalimat baku biasanya akan mudah diucapkan dikarenakan engine microsoft dan fonem kebanyakan menggunakan rangkaian kata baku.

Setelah kalimat dimasukkan secara manual dan maupun dari file text *.txt. maka kita memilih *load Diphone database* supaya *text to speech* maupun *voice command* dapat dijalankan, berikut tampilannya.



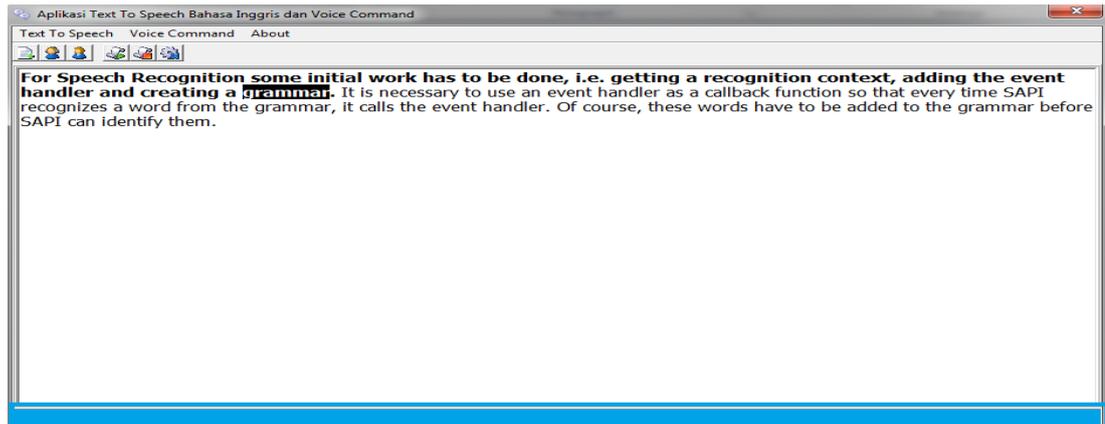
Gambar 4.3 Tampilan Load Diphine Database

Berikutnya melakukan proses *text to speech* terhadap kalimat bahasa Inggris tersebut, untuk melakukan proses *text to speech* bisa dilihat pada gambar dibawah ini:



Gambar 4.4 Tombol Proses Text To Speech

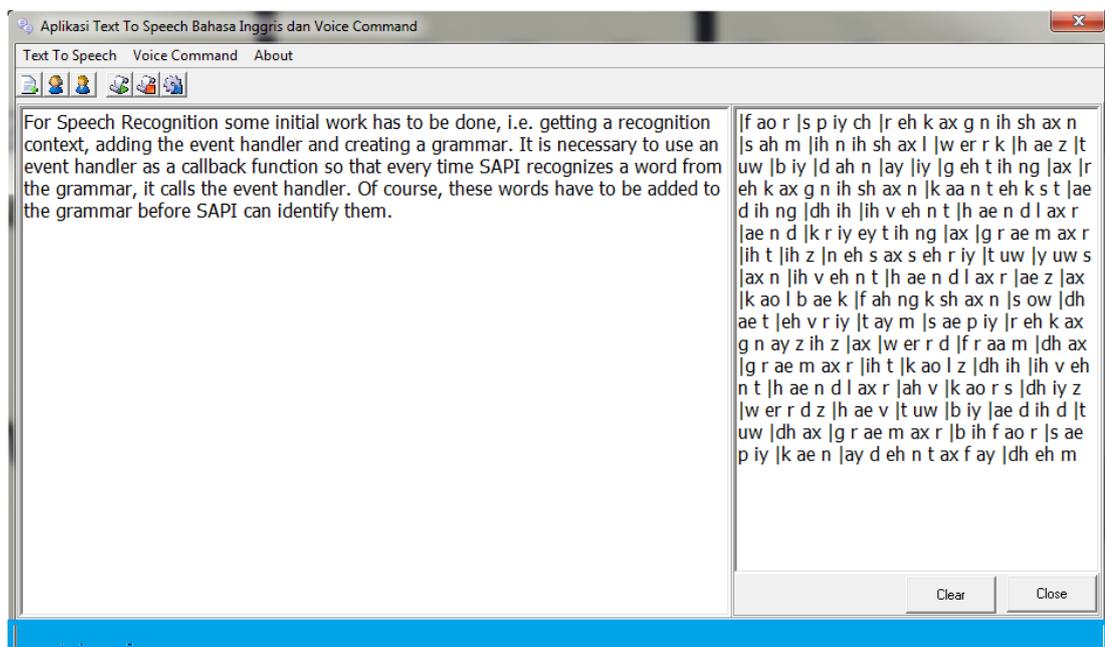
Gambar 4.4 menampilkan tombol yang bisa digunakan untuk melakukan proses suara, tombol warna hijau untuk memulai proses *text to speech*, tombol merah untuk memberhentikan proses dan warna biru untuk memberhentikan sementara proses *text to speech*, pada pengujian ini penulis memilih tombol warna hijau maka akan muncul proses seperti berikut:



Gambar 4.5 Proses Konversi Text To Speech

Kondisi pause akan menghentikan sementara proses *text to speech* dengan kondisi seperti gambar diatas, untuk melanjutkan kembali bisa digunakan tombol *play* pada *toolbar* maka proses *text to speech* akan dilanjutkan dari kata terakhir ketika *pause* dilakukan, untuk menghentikan *text to speech* bisa dilakukan tombol *stop*.

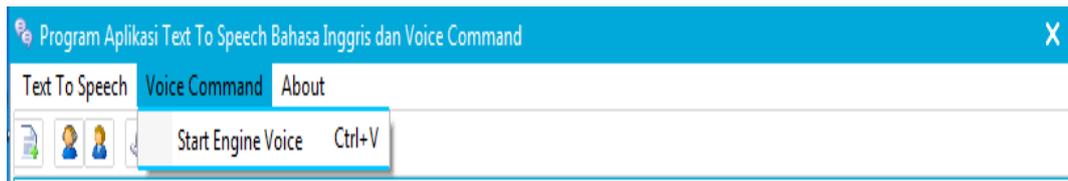
Sub menu *show phoneme* digunakan untuk menampilkan fonem, untuk menampilkannya bisa ditekan dan hasilnya seperti berikut:



Gambar 4.6 Hasil Phonem Kalimat

Gambar 4.8 merupakan hasil fonem dari proses *text to speech* dari kalimat yang dikonversi, hasil fonem berbeda dengan kalinat bahasa inggris, karena fonem merupakan pengucapan dan pengenalan kata yang di ucapkan.

Pengujian berikutnya yang dilakukan adalah pengujian *voice command* atau *voice recognition*, untuk pengujian bisa dilakukan dengan memilih menu *voice command – start engine voice*, berikut tampilannya:



Gambar 4.7 Start Engine Voice

Pada gambar 4.9 merupakan menu untuk memanggil *form voice command*, berikut adalah hasil formnya ketika dipanggil.

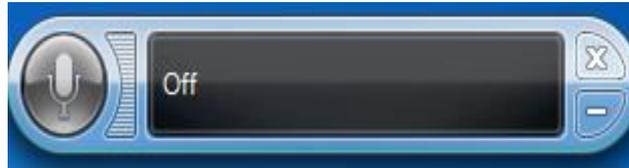
Kata Yang Dikenali Oleh Sistem	Hipotesis Kata Berdasarkan Fonem

Gambar 4.8 Form Voice Command

Gambar 4.10 merupakan form *voice command* yang penulis rancang dengan beberapa tombol yang bisa digunakan seperti tampak pada menu tombol bagian atas, berikut penjelasannya:

1. Engine Properties, fungsi ini digunakan untuk mengakses *speech recognition engine* sebagai proses pemeriksaan suara.
2. Exit, digunakan untuk menutup aplikasi.

Untuk memulai proses *recognition* terhadap suara terlebih dahulu harus mengaktifkan *speech recognition* seperti pada gambar dibawah ini:



Gambar 4.9 Speech Recognition Tool

BAB V

KESIMPULAN DAN SARAN

1. Kesimpulan

Kesimpulan dari pembuatan skripsi ini adalah :

- a. Tersedia *phonem log* pada aplikasi yang memberikan informasi bagaimana kata diproses ketika diucapkan oleh sistem.
- b. Cara aplikasi untuk bisa mengucapkan per kata maupun per kalimat yaitu menggunakan *Microsoft Speech Application Programming Interface (SAPI)*.
- c. *Voice command* sudah berjalan dengan baik walaupun tidak mengenali kata dengan semestinya.
- d. Terjadinya tumpang tindih didalam menghasilkan suara ketika membaca kata yang baru diucapkan dengan kata lain yang langsung diucapkan.

2. Saran

Dari pembuatan skripsi ini disarankan agar :

- a. Mengembangkan lagi perbendaharaan kata bahasa lain selain bahasa inggris yang disimpan didalam database eksternal untuk membaca kata yang berisi kata singkatan.
- b. Membuat sebuah *handle engine* terhadap pemeriksaan suara, apakah aplikasi sedang membaca atau tidak, sehingga ketika membaca yang baru tidak langsung terbaca hingga sebelumnya selesai dibacakan.
- c. Ditambahkannya fungsi pemeriksaan angka sehingga bisa dipisahkan mana yang berbentuk uang ataupun nomor biasa seperti nomor *handphone*.

- d. Perlu menambahkan pemeriksaan kata dan karakter yang lebih baik lagi dengan menerapkan algoritma *Knuth Morris Pratt* dan *Boyer Moore* dan penggunaan aplikasinya menggunakan perangkat mobile atau android.

DAFTAR PUSTAKA

- Abdusy Syari, Aplikasi *Speech Application Programming Interface* (SAPI) 5.1 Sebagai Perintah Untuk Pengoperasian Aplikasi Berbasis Windows, Seminar Nasional Aplikasi Teknologi Informasi 2011 (SNATI 2011) ISSN: 1907-5022, Yogyakarta. Agustinus Noertjahyana, Implementasi Sistem Pengenalan Suara Menggunakan Sapi
- Akbar, A. (2018). Pembangunan Model Electronic Government Pemerintahan Desa Menuju Smart Desa. *Jurnal Teknik dan Informatika*, 5(1), 1-5.
- Anggra Narullita, Aplikasi Pensintesa Ucapan Bahasa Indonesia Sebagai Pembaca Email, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro, Semarang, 2011.
- Anna Dara Andriana, Perangkat Lunak Untuk Membuka Aplikasi Pada Komputer Dengan Perintah Suara Menggunakan Metode *Mel Frequency Cepstrum Coefficients*, *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)* Vol. 2, No. 1, Maret 2013, ISSN : 2089-9033.
- Budhi Supriyono, Perancangan Aplikasi Voice User Interface Dengan Menggunakan *Microsoft Speech Api*, Universitas Diponegoro Semarang, 2012.
- Dhany, H. W., Izhari, F., Fahmi, H., Tulus, M., & Sutarman, M. (2017, October). Encryption and decryption using password based encryption, MD5, and DES. In *International Conference on Public Policy, Social Computing and Development 2017 (ICOPOSDev 2017)* (pp. 278-283). Atlantis Press.
- Hao Shi, *Speech-enabled windows application using Microsoft SAPI*, *IJCSNS International Journal of Computer Science and Network Security*, VOL.6 No.9A, September 2006.
- Hariyanto, E., & Rahim, R. (2016). Arnold's cat map algorithm in digital image encryption. *International Journal of Science and Research (IJSR)*, 5(10), 1363-1365.
- Havena, M., & Marlina, L. (2018). The Technology of Corn Processing as an Effort to Increase The Income of Kelambir V Village. *Journal of Saintech Transfer*, 1(1), 27-32.
- Haviluddin, Memahami Penggunaan UML (*Unified Modelling Language*), *Jurnal Informatika Mulawarman* Vol 6 No. 1 Februari 2011.

- Hendrawan, J. (2018). Rancang Bangun Aplikasi Mobile Learning Tuntunan Shalat. *INTECOMS: Journal of Information Technology and Computer Science*, 1(1), 44-59.
- Isrokah dan Mohamad Yasin, Aplikasi Konversi Teks Menjadi Suara Dengan Menggunakan Metode Penggal Kata Finite State Automata (FSA), Universitas Negeri Malang, 2013.
- Iwan Iwut Tritoasmoro, Text-To-Speech Bahasa Indonesia Menggunakan *Concatenation Synthesizer Berbasis Fonem*, Seminar Nasional Sistem dan Informatika 2006; Bali, November 17, 2006.
- Kurnia, D. (2017). Analisis QoS Pada Pembagian Bandwidth Dengan Metode Layer 7 Protocol, PCQ, HTB Dan Hotspot Di SMK Swasta Al-Washliyah Pasar Senen. *CESS (Journal of Computer Engineering, System and Science)*, 2(2), 102-111.
- Kurnia, D., Dafitri, H., & Siahaan, A. P. U. (2017). RSA 32-bit Implementation Technique. *Int. J. Recent Trends Eng. Res*, 3(7), 279-284.
- Mariance, U. C. (2018). Analisa dan Perancangan Media Promosi dan Pemasaran Berbasis Web Menggunakan Work System Framework (Studi Kasus di Toko Mandiri Prabot Kota Medan). *Jurnal Ilmiah Core IT: Community Research Information Technology*, 6(1).
- Marlina, L., Muslim, M., Siahaan, A. U., & Utama, P. (2016). Data Mining Classification Comparison (Naïve Bayes and C4. 5 Algorithms). *Int. J. Eng. Trends Technol*, 38(7), 380-383.
- Marlina, L., Putera, A., Siahaan, U., Kurniawan, H., & Sulistianingsih, I. (2017). Data Compression Using Elias Delta Code. *Int. J. Recent Trends Eng. Res*, 3(8), 210-217.
- Nessa Putri Andayu, Perancangan *Text To Speech Converter Engine* Dalam Pengucapan Kata Berbahasa Arab Sehari-Hari, Jurusan Teknik Elektro Fakultas Teknik Universitas Tanjungpura, 2013.
- Pande Made Mahendri Pramadewi, Pengembangan Aplikasi *Text to Speech* untuk Bahasa Bali, *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI) Volume 2, Nomor 3, Desember 2013 ISSN 2089-8673*.
- Putri, N. A. (2018). Sistem Pakar untuk Mengidentifikasi Kepribadian Siswa Menggunakan Metode Certainty Factor dalam Mendukung Pendekatan

- Guru. INTECOMS: Journal of Information Technology and Computer Science, 1(1), 78-90.
- Rahim, R., Aryza, S., Wibowo, P., Harahap, A. K. Z., Suleman, A. R., Sihombing, E. E., ... & Agustina, I. (2018). Prototype file transfer protocol application for LAN and Wi-Fi communication. *Int. J. Eng. Technol.*, 7(2.13), 345-347.
- Ruwaida, D., & Kurnia, D. (2018). Rancang Bangun File Transfer Protocol (FTP) dengan Pengamanan Open SSL pada Jaringan VPN Mikrotik di SMK Dwiwarna. *CESS (Journal of Computer Engineering, System and Science)*, 3(1), 45-49.
- Sarif, M. I. (2017). Penemuan Aturan yang Berkaitan dengan Pola dalam Deret Berkala (Time Series).
- Sarif, M. I. Classification Of Feasibility Of Basic Food Recipients In Kelurahan Tanjung Morawa A, Tanjung Morawa Sub-District Using Naïve Bayes Classifier Algorithm.
- Sumartono, I., Siahaan, A. P. U., & Mayasari, N. (2016). An overview of the RC4 algorithm. *IOSR J. Comput. Eng.*, 18(6), 67-73.
- Sutarman, *Indonesian Text-To-Speech System Using Diphone Concatenative Synthesis*, *International Journal of Software Engineering & Computer Sciences (IJSECS)* ISSN: 2289-8522 ,Volume 1, pp. 85-93, February 2015.