

**IMPLEMENTASI TEXT TO SPEECH
PADA SCREEN READER BERBASIS MBROLA**

SKRIPSI

Oleh :
Muchammad Faisol
NIM. 06550077



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
2013**

**IMPLEMENTASI *TEXT TO SPEECH*
PADA *SCREEN READER* BERBASIS *MBROLA***

SKRIPSI

Oleh:
MUCHAMAD FAISOL
NIM. 06550077



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2013**

**IMPLEMENTASI *TEXT TO SPEECH*
PADA *SCREEN READER* BERBASIS *MBROLA***

SKRIPSI

Diajukan kepada

Universitas Islam Negeri Maulana Malik Ibrahim Malang

Untuk Memenuhi Salah Satu Persyaratan

Dalam Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh:

MUCHAMAD FAISOL

NIM. 06550077

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
2013**

LEMBAR PERSETUJUAN

**IMPLEMENTASI *TEXT TO SPEECH*
PADA *SCREEN READER* BERBASIS *MBROLA***

SKRIPSI

Oleh:

Muchamad Faisol

NIM: 06550077

Telah disetujui oleh:

Pembimbing I

Pembimbing II

Totok Chamidy, M.Kom
NIP. 196912222006041001

Dr. Suhartono, M.Kom
NIP. 196805192003121001

22 Juli 2013

Mengetahui,

Ketua Jurusan Teknik Informatika

Ririen Kusumawati, M. Kom
NIP. 197203092005012002

LEMBAR PENGESAHAN
IMPLEMENTASI *TEXT TO SPEECH*
PADA *SCREEN READER* BERBASIS *MBROLA*

SKRIPSI

Dipersiapkan dan disusun oleh:
Muchamad Faisol
NIM. 06550077

Telah Dipertahankan Di Depan Dewan Penguji Skripsi
Dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal 22 Juli 2013

	Susunan Dewan Penguji:	Tanda Tangan
1. Penguji Utama:	Zainal Abidin, M.Kom NIP. 19760613 200501 1 004	()
2. Ketua Penguji:	M. Ainul Yaqin, M.Kom NIP.19761013 200604 1 004	()
3. Sekretaris Penguji:	Totok Chamidy, M.Kom NIP. 19791222 200604 1 001	()
4. Anggota Penguji:	Dr. Suhartono, M.Kom NIP. 19680519 200312 1 001	()

Mengetahui dan Mengesahkan
Ketua Jurusan Teknik Informatika
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Ririen Kusumawati, M.Kom
NIP. 19720309 200501 2 002

LEMBAR PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : Muchamad Faisol

NIM : 06550077

Jurusan : Teknik Informatika

Judul Skripsi : **IMPLEMENTASI TEXT TO SPEECH PADA SCREEN
READER BERBASIS MBROLA**

Dengan ini menyatakan bahwa:

1. Isi dari Skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama termaktub di isi dan tertulis di daftar pustaka dalam Skripsi ini.
2. Apabila kemudian hari ternyata Skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 23 Juli 2013

Yang membuat pernyataan

Muchamad Faisol
NIM. 06550077

MOTTO

Penderitaan jiwa mengarahkan keburukan. Putus asa adalah sumber kesesatan; dan kegelapan hati, pangkal penderitaan jiwa.

(Bediuzzaman Said Nursi)



PERSEMBAHAN

Dengan segala kerendahan hari dan ucapan syukur kehadiran Allah SWT,
kupersembahkan karya sederhana ini untuk Ayah tercinta
Untuk semua keluarga, dan untuk sahabat-sahabatku Jurusan Teknik Informatika
angkatan 2006 yang selalu ada ketika dibutuhkan
Terima kasih untuk semuanya



KATA PENGANTAR

Segala puji bagi Allah SWT yang telah melimpahkan karunia-Nya kepada penulis sehingga bisa menyelesaikan skripsi dengan judul “Implementasi *Text To Speech* Pada *Screen Reader* Berbasis *Mbrola*”.

Sholawat dan salam semoga selalu tercurah kepada Nabi Muhammad SAW karena bimbingan beliau, Islam telah disampaikan dan membawa umat dari kegelapan menuju zaman yang terang benderang.

Penyelesaian skripsi ini akan sulit terwujud tanpa bantuan dan sumbangsih berbagai pihak karena penulis memiliki keterbatasan kemampuan dan pengetahuan. Untuk itu, dengan segala kerendahan hati maka penulis mengucapkan terima kasih kepada:

1. Prof. Dr. H. Mudjia Rahardjo M.Sc selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang
2. Dr. Drh. Bayyinatul Muchtaromah, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang
3. Ririen Kusumawati, M.Kom, selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
4. Muhammad Faisal, M.T, selaku Dosen Wali yang sudah banyak meluangkan waktu untuk membimbing dan mengarahkan mulai awal sampai akhir.

5. Totok Chamidy, M.Kom selaku pembimbing I yang telah meluangkan banyak waktu untuk membimbing dan mengarahkan dalam penyelesaian skripsi
6. Dr. Suhartono, M.Kom selaku pembimbing II, yang membantu penulis serta banyak memberikan petunjuk dan nasehat dalam penyelesaian skripsi
7. Segenap Dosen Teknik Informatika yang telah memberikan bimbingan keilmuan kepada penulis selama masa studi.
8. Ayah dan Ibu, yang selalu memberikan dorongan moral, spiritual, dan material sehingga penulisan skripsi ini dapat terselesaikan dengan baik.
9. Teman-teman jurusan Teknik Informatika dan untuk seluruh civitas akademika Universitas Islam Negeri Maulana Malik Ibrahim Malang yang telah banyak membantu perjuangan dan berbagi pengalaman dalam kehidupan sehari-hari.
10. Semua pihak yang tidak mungkin penulis sebutkan satu persatu, atas segala yang telah diberikan kepada penulis dan dapat menjadi pelajaran.

Sebagai penutup, penulis menyadari masih banyak kekurangan dalam skripsi ini. Semoga apa yang menjadi kekurangan bisa disempurnakan oleh peneliti selanjutnya. Harapan penulis selanjutnya adalah semoga karya ini bermanfaat bagi kita semua. Aamiin.

Malang, 23 Juli 2013

Muchamad Faisol

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
LEMBAR PERSETUJUAN	iii
LEMBAR PENGESAHAN	iv
LEMBAR PERNYATAAN	v
MOTTO	vi
PERSEMBAHAN.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiii
ABSTRAK	xiv
BAB I: PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian	4
1.4 Manfaat Penelitian.....	4
1.5 Batasan Masalah	4
1.6 Metode Penelitian	4
1.7 Sistematika Penulisan Skripsi.....	4
BAB II: TINJAUAN PUSTAKA	6
2.1 Sistem Text to Speech	6
2.2 Konversi Teks Ke Ucapan	8
2.3 TTS Bahasa Indonesia	14
2.4 Fonem Bahasa Indonesia	16
2.5 Synthesizer	18
2.6 MBROLA	21
2.7 Database Diphone MBROLA.....	22
2.8 Screen Reader	24
2.8.1 Perkembangan Screen reader di Indonesia.....	24
2.9 Visual Basic	26
2.10 Pemrograman API	30
2.11 UML (<i>Unified Modeling Language</i>)	35
2.12 Pengenalan UML.....	36
2.13 MSAA.....	42
2.14 Bahasa dalam kajian Islam	42
BAB III: ANALISIS DAN PERANCANGAN SISTEM	45
3.1 Text To Speech Synthesis System.....	45
3.1.1 Text-pre-Processing.....	45
3.1.2 Prosody	47
3.1.3 Concatenation	48

3.2 Engine Speech Synthesizer Mbrola.....	48
3.3 Analisis Usecase Diagram.....	50
3.4 Analisis Squence Diagram.....	52
3.5 Analisis Activity Diagram.....	53
3.6 Rancangan Antar Muka.....	54
BAB IV: HASIL PEMBAHASAN.....	55
4.1 Implementasi.....	55
4.2 Ruang Lingkup Perangkat keras.....	55
4.3 Ruang Lingkup Perangkat Lunak.....	55
4.4 Sistem kerja Software.....	56
4.5 Hasil Uji coba.....	60
BAB V: PENUTUP.....	63
5.1 Kesimpulan.....	63
5.2 Saran.....	64
DAFTAR PUSTAKA.....	65
LAMPIRAN.....	66



DAFTAR TABEL

Tabel 2.1 Database Diphone ID1	23
Tabel 4.1 Masukan yang diproses oleh Mbrola	61
Tabel 4.2 Data hasil pengujian kecakapan pengucapan.....	61



DAFTAR GAMBAR

Gambar 2.1 Diagram Blok <i>Text to Speech</i> Bahasa Indonesia.....	6
Gambar 2.2 Urutan Proses Konversi Teks ke Ucapan.....	7
Gambar 2.3 Alur Konversi Teks ke Ucapan.....	9
Gambar 2.4. Urutan Proses Konversi dari Teks ke Ucapan.....	12
Gambar 2.5. Tahap Proses Konversi dari Teks ke Ucapan.....	14
Gambar 2.6 Konfigurasi Sistem IndoTTS	15
Gambar 2.7 Diagram Pengembangan Aplikasi Menggunakan IndoTTS.....	16
Gambar 2.8. Diagram Sistem <i>Text-to-Speech Synthesis</i>	19
Gambar 2.9 Interface Visual Basic	27
Gambar 2.10 Komponen standar dalam Toolbox	28
Gambar 2.11 Sebuah Kelas dari model UML.....	38
Gambar 2.12 Sebuah interface/antar-muka.....	38
Gambar 2.13 Collaborations	39
Gambar 2.14 Usecase.....	39
Gambar 2.15 Nodes.....	40
Gambar 2.16 <i>Dependency</i>	40
Gambar 2.17 Association	40
Gambar 2.18 Generalizations.....	41
Gambar 2.19 Realizations	41
Gambar 3.1 Blok diagram <i>Text to speech synthesis system</i>	45
Gambar 3.2 Blok diagram <i>text pre-processing</i>	46
Gambar 3.3 Blok diagram <i>prosody</i>	47
Gambar 3.4 <i>Use case diagram</i>	51
Gambar 3.5 <i>Squence Diagram Screen Reader</i>	52
Gambar 3.6 <i>Activity Diagram Screen Reader</i>	53
Gambar 3.7 Desain Menu Utama.....	54
Gambar 4.1 form <i>Add Database</i>	56
Gambar 4.2 <i>Window Mbrola settings</i>	57
Gambar 4.3 Registrasi .pho dan .phs.....	57
Gambar 4.4 Tampilan screen reader ketika membaca pergerakan mouse	60

ABSTRAK

Faisol, Muchamad. 2013. 06550077. **Implementasi *Text to Speech* Pada *Screen Reader* Berbasis *Mbrola***. Skripsi, Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (1) Totok Chamidy, M.Kom (2) Dr. Suhartono, M.Kom.

Kata Kunci: *Text to Speech, Screen Reader, Mbrola*

Text to speech adalah suatu sistem yang melakukan konversi dari teks menjadi ucapan secara otomatis dengan cara fonetisasi (penyusunan fonem-fonem untuk membentuk ucapan). Perbedaan sistem *text to speech* dengan *talking machine* biasa adalah keotomatisannya dalam mengucapkan kata-kata baru. Oleh karena itu sistem *text to speech* memungkinkan diimplementasikan pada bidang aplikasi yang beragam seperti *screen reader*.

Screen reader merupakan bentuk teknologi bantu (*assistive technology*) yang digunakan untuk membantu orang yang memiliki keterbatasan penglihatan (*low vision*) dalam mengakses komputer. Sebuah *screen reader* akan membaca isi (tulisan) dari layar komputer dengan menggunakan logat inggris. Dan umumnya *screen reader* tidak menggunakan teknologi *text to speech* melainkan menggunakan suara sintesis robot (*robotic synthetic voice*).

MBROLA adalah *Speech synthesizer* yang menggunakan teknik penggabungan segmen bunyi berdasarkan pangkalan *diphone* (*Diphone Concatenation*).

Teknik *diphone concatenation* bekerja dengan cara menggabung-gabungkan segmen-segmen bunyi yang telah direkam sebelumnya. Setiap segmen berupa *diphone* (gabungan dua buah fonem). *Diphone* merupakan unit bunyi ucapan yang paling sering digunakan sebagai unit database. Berdasarkan penelitian yang ada *synthesizer* ini dapat menghasilkan bunyi ucapan dengan tingkat kealamian yang tinggi.

Pada penelitian ini sistem *text to speech* dengan menggunakan *MBROLA Synthesizer* akan diimplementasikan pada *screen reader*, sehingga dapat menghasilkan suara yang memiliki interpretasi yang baik dan memiliki tingkat kealamian yang tinggi.

ABSTRACT

Faisol, Muchamad. 2013. 06550077. Implementation Text to Speech On Screen Reader MBROLA Based. Thesis, Department of Informatics, Faculty of Science and Technology of the State Islamic University of Maulana Malik Ibrahim Malang. Supervisor: (1) Totok Chamidy, M. Kom (2) Dr. Suhartono, M.Kom.

Keywords: *Text to Speech, Screen Reader, Mbrola*

Text to speech is a system that performs the conversion of text to speech automatically by means fonetisasi (preparation of phonemes to form words). Differences in text to speech system with regular talking machine is keotomatisannya in pronouncing new words. Therefore the text to speech system implemented on field enables diverse applications such as screen readers.

Screen readers are a form of assistive technology (assistive technology) which is used to help people who have limited vision (low vision) to access the computer. A screen reader will read the contents (text) of a computer screen using the English accent. And his general not to use a screen reader text to speech technology it uses robot voice synthesis (synthetic robotic voice).

MBROLA is Speech synthesizer which uses sound segment merging technique based on diphone base (Diphone Concatenation).

Diphone concatenation technique works by combining segments combined the sounds that have been recorded previously. Each segment of a diphone (combination of two phonemes). Diphone is a speech sound units are most often used as a database unit. Based on existing research synthesizer can produce speech sounds with a high level of naturalness.

In this study the system using a text to speech synthesizer MBROLA will be implemented on a screen reader, so as to produce a sound that has a good interpretation and have a high level of naturalness.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Bahasa merupakan sistem tanda bunyi yang disepakati untuk dipergunakan oleh para anggota kelompok masyarakat tertentu dalam bekerja sama, berkomunikasi, dan mengidentifikasi diri.

Bahasa sebagai sarana komunikasi mempunyai fungsi utama bahasa adalah bahwa komunikasi ialah penyampaian pesan atau makna oleh seseorang kepada orang lain. Sebagaimana yang difirmankan Allah dalam surah Ibrahim ayat 4 sebagai berikut :

وَمَا أَرْسَلْنَا مِنْ رَسُولٍ إِلَّا بِلِسَانِ قَوْمِهِ لِيُبَيِّنَ لَهُمْ فَيُضِلُّ اللَّهُ مَنْ يَشَاءُ
وَيَهْدِي مَنْ يَشَاءُ وَهُوَ الْعَزِيزُ الْحَكِيمُ ﴿٤﴾

Artinya :

Kami tidak mengutus seorang rasulpun, melainkan dengan bahasa kaumnya, supaya ia dapat memberi penjelasan dengan terang kepada mereka. Maka Allah menyesatkan siapa yang Dia kehendaki, dan memberi petunjuk kepada siapa yang Dia kehendaki. Dan Dia-lah Tuhan Yang Maha Kuasa lagi Maha Bijaksana. (QS. Ibrahim :4)

Dalam ayat di atas menunjukkan keterikatan dan keterkaitan bahasa dengan manusia menyebabkan bahasa tidak tetap dan selalu berubah seiring perubahan kegiatan manusia dalam kehidupannya di masyarakat, dalam hal ini Allah mengutus rasul dengan bahasa yang sesuai dengan kaumnya, agar pesan/ajaran yang disampaikan rasul bisa dipahami oleh kaumnya.

Bentuk utama representasi bahasa adalah suara/ucapan, tetapi sering pula dinyatakan dalam bentuk tulisan. Chomsky adalah orang yang pertama kali merepresentasikan bahasa sebagai rangkaian simbol. Chomsky berhasil memperlihatkan bahwa bahasa apapun dapat direpresentasikan dengan suatu cara yang universal. Pemikiran Chomsky yang merepresentasikan bahasa sebagai kumpulan simbol-simbol dan aturan yang mengatur susunan simbol-simbol tersebut telah membuka peluang untuk melakukan pemrosesan bahasa secara simbolik dengan teknologi komputer, sehingga melahirkan ilmu *Natural Language Processing* (pemrosesan bahasa alami). Suatu sistem pemrosesan bahasa alami secara lisan dapat dibentuk dari tiga sub sistem yang diantaranya adalah sistem *Text to Speech*.

Pada prinsipnya *Text to speech* adalah suatu sistem yang melakukan konversi dari teks menjadi ucapan secara otomatis dengan cara fonetisasi (penyusunan fonem-fonem untuk membentuk ucapan). Perbedaan sistem *text to speech* dengan *talking machine* biasa adalah keotomatisannya dalam mengucapkan kata-kata baru. Oleh karena itu sistem *text to speech* memungkinkan diimplementasikan pada bidang aplikasi yang beragam seperti *screen reader*.

Screen reader merupakan bentuk teknologi bantu (*assistive technology*) yang digunakan untuk membantu orang yang memiliki keterbatasan penglihatan (*low vision*) dalam mengakses komputer. Sebuah *screen reader* akan membaca isi (tulisan) dari layar komputer dengan menggunakan logat inggris. Dan umumnya *screen reader* tidak menggunakan teknologi *text to speech* melainkan menggunakan suara sintesis robot (*robotic synthetic voice*).

Solusi untuk keterbatasan tersebut dilakukan dengan mengimplementasikan teknologi *text to speech* pada *screen reader*. Dengan pengimplementasian sistem *text to speech* pada *screen reader*, maka suara yang dihasilkan memiliki tingkat kemudahan interpretasi yang baik.

Dalam proses sintesa sinyal ucapan terdapat tiga metode dasar diantaranya, Sintesis *articulatory*, yang berusaha memodelkan sistem produksi sinyal ucapan manusia dengan pendekatan fisik mekanis secara langsung, sintesis *formant*, yang memodelkan frekuensi *pole* suatu sinyal ucapan atau fungsi transfer yang berbasis *vocal track* atau model *source-filter*, sintesis perangkaian (*concatenation*), yang menggunakan panjang bagian yang berbeda dari suatu perekaman sinyal ucapan alami. Namun demikian teknik yang sering digunakan salah satunya adalah *diphone concatenation* yang terdapat pada *MBROLA synthesizer*.

MBROLA adalah *Speech synthesizer* yang menggunakan teknik penggabungan segmen bunyi berdasarkan pangkalan *diphone* (*Diphone Concatenation*).

Teknik *diphone concatenation* bekerja dengan cara menggabungkan segmen-segmen bunyi yang telah direkam sebelumnya. Setiap segmen berupa *diphone* (gabungan dua buah fonem). *Diphone* merupakan unit bunyi ucapan yang paling sering digunakan sebagai unit database. Berdasarkan penelitian yang ada *synthesizer* ini dapat menghasilkan bunyi ucapan dengan tingkat kealamian yang tinggi.

Pada penelitian ini sistem *text to speech* dengan menggunakan *MBROLA Synthesizer* akan diimplementasikan pada *screen reader*, sehingga dapat

menghasilkan suara yang memiliki interpretasi yang baik dan memiliki tingkat kealamian yang tinggi.

1.2 Rumusan Masalah

Dari latar belakang di atas, maka rumusan masalah untuk penelitian ini adalah bagaimana mengimplementasikan *text to speech* pada *screen reader*?

1.3 Tujuan Penelitian

Tujuan Penelitian ini adalah membangun Aplikasi screen reader dengan menggunakan MBROLA sebagai *speech synthesizer*-nya

1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah menghasilkan aplikasi screen reader berbahasa indonesia dengan tingkat kealamian yang tinggi

1.5 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut :

1. Penelitian menggunakan satu database bahasa yaitu database id1(database *MBROLA* berbahasa indonesia)
2. Metode yang digunakan adalah *Diphone concatenation* yang ada pada *engine* Mbrola

1.6 Sistematika Pembahasan

Sistematika penulisan penelitian ini dibagi menjadi 5 bab,yakni:

BAB I PENDAHULUAN

Berisi tentang latar belakang pemilihan judul Implementasi *Text to Speech* pada *screen reader* menggunakan *MBROLA BASED*, didalamnya terdapat rumusan masalah, batasan masalah, tujuan, manfaat penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini dibahas teori – teori yang menjadi acuan dalam pembuatan analisa dan pemecahan dari permasalahan yang dibahas, sehingga memudahkan penulis dalam menyelesaikan masalah.

BAB III ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini dijelaskan mengenai tahapan – tahapan yang dilalui dalam penyelesaian tugas akhir, diawali tahap perencanaan, desain dan perancangan sistem *text to speech* menggunakan *Diphone Concatenation*.

BAB IV HASIL DAN PEMBAHASAN

Hasil dan pembahasan berisi tentang deskripsi program, cara pemakaian aplikasi, database diphone id1 dan beberapa fungsi algoritma yang menjelaskan proses pengolahan teks ke ucapan dengan diphone Mbrola.

BAB V PENUTUP

Berisi tentang kesimpulan yang diambil dari pembahasan pembuatan sistem *text to speech* berbahasa indonesia dan saran untuk pengembangan.

BAB II

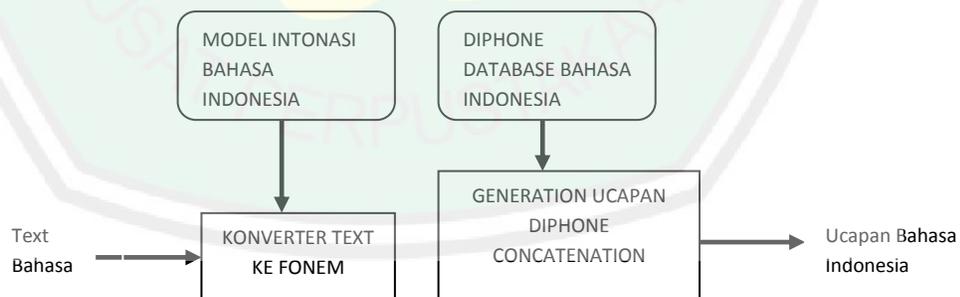
KAJIAN PUSTAKA

2.1 Sistem Text To Speech

Secara fungsional, *Text to Speech* melakukan proses sebaliknya dari system pengenalan ucapan. Namun demikian pendekatan implementasinya sama sekali berbeda. Artinya komponen-komponen pembentuk kedua system tersebut sama sekali berbeda.

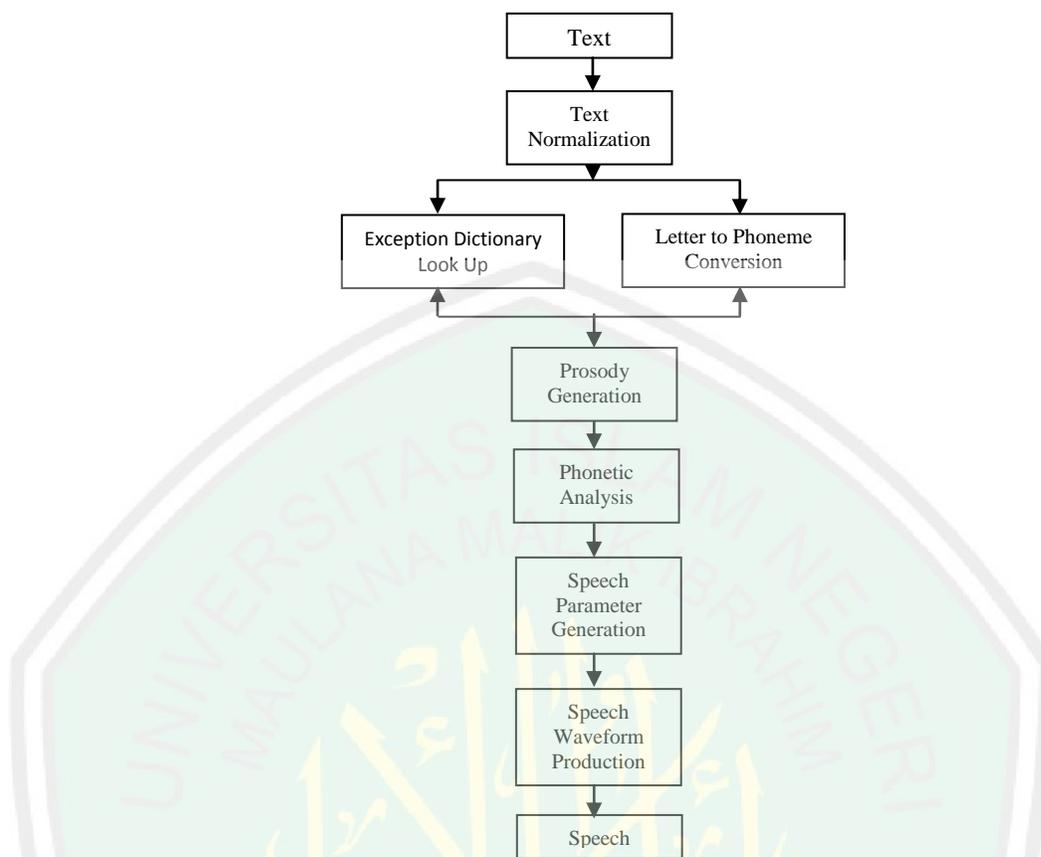
Pada dasarnya *Text to Speech* adalah suatu sistem yang dapat mengubah teks menjadi ucapan. Suatu system pesintesa ucapan atau *Text to Speech* pada prinsipnya terdiri dari dua sub-sistem, yaitu :

- 1) Bagian konverter teks ke fonem (*Text to Phoneme*), serta
- 2) Bagian converter fonem ke ucapan (*phoneme to speech*)



Gambar 2.1 Diagram Blok *Text to Speech* Bahasa Indonesia

[Sumber : Dr. Arry Akhmad Arman : 2001]



Gambar 2.2 Urutan Proses Konversi Teks ke Ucapan
[Sumber : Dr. Arry Akhmad Arman :2001]

Bagian konverter teks ke fonem berfungsi untuk mengolah kalimat masukan dalam suatu bahasa tertentu yang berbentuk teks menjadi urutan kode-kode 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.

Setiap fonem harus dilengkapi dengan informasi durasi dan pitch. Informasi durasi diperlukan untuk menentukan berapa lama suatu fonem diucapkan, sedangkan informasi pitch diperlukan untuk menentukan tinggi rendahnya nada pengucapan

suatu fonem. Durasi dan pitch bersama-sama akan membentuk intonasi suatu ucapan. Kedua informasi ini dalam suatu sistem TTS biasanya dibangkitkan oleh modul pembangkit/model intonasi.

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 (*language dependent*).

Bagian Konverter Fonem ke Ucapan akan menerima masukan kode-kode fonem serta pitch 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.

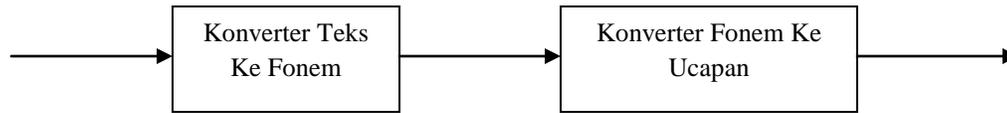
Ada beberapa alternatif teknik yang dapat digunakan untuk implementasi bagian konverter fonem ke ucapan. Dua teknik yang paling banyak digunakan adalah formant synthesizer, serta diphone concatenation. Saat ini, teknik kedua lebih banyak digunakan karena dapat menghasilkan ucapan dengan kualitas yang lebih alami.

Teknik diphone concatenation melakukan pembangkitan ucapan dengan cara menggabung-gabungkan segmen-segmen bunyi yang berupa diphone (dua fonem). Untuk mencapai kualitas yang lebih tinggi, beberapa TTS menggunakan penggabungan segmen bunyi yang berupa multi-phone.

2.2 Konversi Teks Ke Ucapan

Sistem Text to Speech pada prinsipnya terdiri dari dua sub sistem, yaitu :

- 1) bagian Konverter Teks ke Fonem (Text to Phoneme), serta
- 2) bagian Konverter Fonem to Ucapan (Phoneme to Speech).



Gambar 2.3 Alur Konversi Teks ke Ucapan

[Sumber : Konversi teks ke ucapan oleh Dr. Arry Akhmad Arman]

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. Bagian ini bersifat sangat language dependant. Untuk suatu bahasa baru, bagian ini harus dikembangkan secara lengkap khusus untuk bahasa tersebut.

Bagian Konverter Fonem ke Ucapan akan menerima masukan berupa kode-kode fonem serta pitch dan durasi yang dihasilkan oleh bagian sebelumnya. Berdasarkan kode-kode tersebut, bagian Konverter Fonem ke Ucapan akan menghasilkan bunyi atau sinyal ucapan yang sesuai dengan kalimat yang ingin diucapkan. Ada beberapa alternatif teknik yang dapat digunakan untuk implementasi bagian ini. Dua teknik yang banyak digunakan adalah formant synthesizer, serta diphone concatenation.

Formant synthesizer bekerja berdasarkan suatu model matematis yang akan melakukan komputasi untuk menghasilkan sinyal ucapan yang diinginkan. Synthesizer jenis ini telah lama digunakan pada berbagai aplikasi. Walaupun dapat menghasilkan ucapan dengan tingkat kemudahan interpretasi yang baik, synthesizer ini tidak dapat menghasilkan ucapan dengan tingkat kealamian yang tinggi.

Synthesizer yang menggunakan teknik diphone concatenation bekerja dengan cara menggabung-gabungkan segmen-segmen bunyi yang telah direkam

sebelumnya. Setiap segmen berupa diphone (gabungan dua buah fonem). Synthesizer jenis ini dapat menghasilkan bunyi ucapan dengan tingkat kealamian (naturalness) yang tinggi. Struktur sistem seperti di atas pada prinsipnya merupakan konfigurasi tipikal yang digunakan pada berbagai sistem Text to Speech berbagai bahasa. Namun demikian, pada setiap sub-sistem terdapat sifat-sifat serta proses-proses yang sangat spesifik dan sangat tergantung dari bahasanya.

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 pitch (frekuensi dasar) selama pengucapan kalimat dilakukan atau pitch sebagai fungsi waktu. Pada prakteknya, informasi pembentuk prosodi berupa data-data pitch serta durasi pengucapannya untuk setiap fonem yang dibangkitkan. Nilai-nilai yang dihasilkan diperoleh dari suatu model prosodi. Prosodi bersifat sangat spesifik untuk setiap bahasa, sehingga model yang diperlukan untuk membangkitkan data-data prosodi menjadi sangat spesifik juga untuk suatu bahasa. Beberapa model umum prosodi pernah dikembangkan, tetapi untuk digunakan pada suatu bahasa masih perlu banyak penyesuaian yang harus dilakukan.

Konverter fonem ke ucapan berfungsi untuk membangkitkan sinyal ucapan berdasarkan kode-kode fonem yang dihasilkan dari proses sebelumnya. Sub sistem ini harus memiliki pustaka setiap unit ucapan dari suatu bahasa. Pada sistem yang menggunakan teknik diphone concatenation, sistem harus didukung oleh suatu diphone database yang berisi

rekaman segmen-segmen ucapan yang berupa diphone. 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.

Tahapan-tahapan utama konversi dari teks menjadi ucapan dapat dinyatakan dengan diagram seperti terlihat pada Gambar 2.4.

Tahap normalisasi teks berfungsi untuk mengubah semua teks kalimat yang ingin diucapkan menjadi teks yang secara lengkap memperlihatkan cara pengucapannya. Lihat contoh kalimat dan hasil normalisasinya pada Gambar 2.5.

Tahap berikutnya adalah melakukan konversi dari teks yang sudah secara lengkap merepresentasikan kalimat yang ingin diucapkan menjadi kode-kode fonem. Konversi teks menjadi fonem biasanya dilakukan dengan dua cara. Sebagian proses konversi dapat dilakukan dengan aturan konversi yang sederhana dan berlaku umum untuk berbagai kondisi. Sebagian proses lainnya bersifat kondisional, tergantung dari huruf-huruf atau fonem-fonem tetangganya, bahkan terdapat bentuk-bentuk translasi yang tidak dapat ditemukan keteraturannya.

Konversi yang teratur 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 kanannya terpenuhi. Contoh bentuk aturan konversi huruf ke fonem yang memenuhi teknik tersebut adalah sebagai berikut.

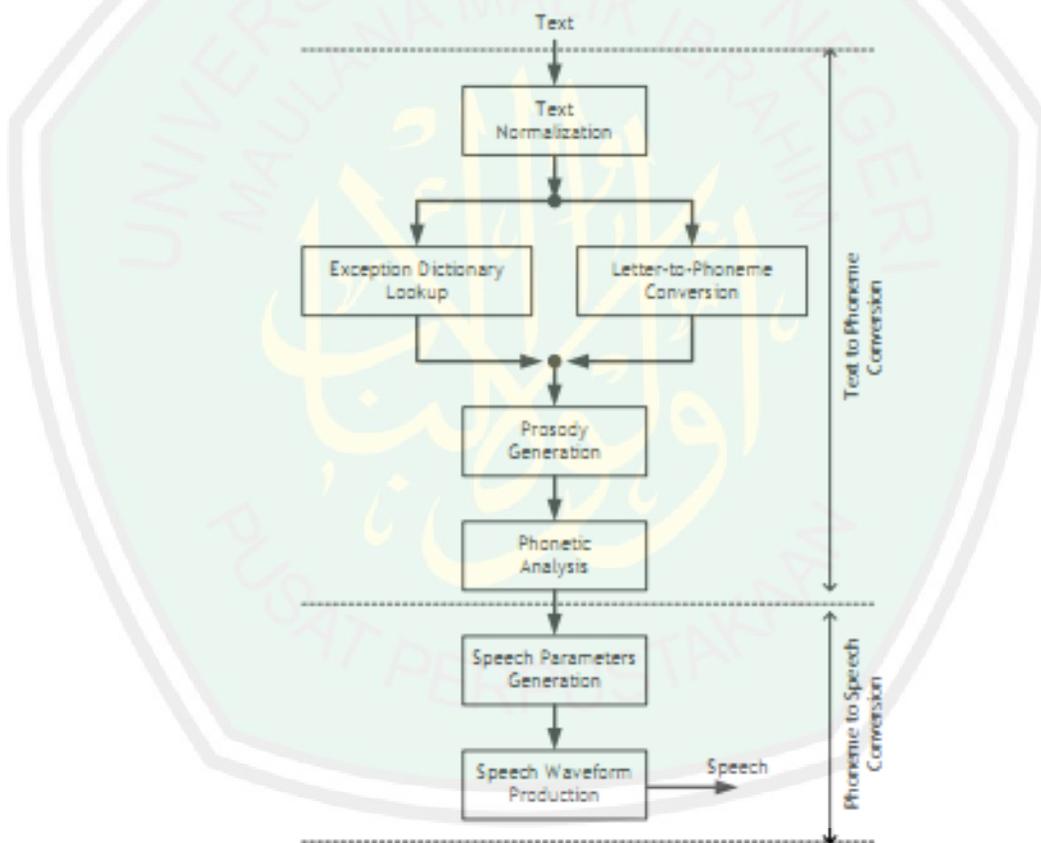
Left-context [letter-set] right-context = phoneme string

Huruf tertentu yang ditunjuk dalam posisi [letter-set] akan dikonversikan menjadi suatu fonem dalam “phoneme string” jika left-context dan right context terpenuhi. Bahasa Inggris termasuk bahasa yang mempunyai keteraturan yang rendah untuk proses konversi teks ke fonem. Suatu 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 dikonversikan 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. Dalam blok diagram di atas, kondisi yang masih dapat ditangani oleh aturan diimplementasikan dengan blok Letter to Phoneme Conversion. Konversi yang tidak teratur ditangani oleh bagian Exception Dictionary Lookup.

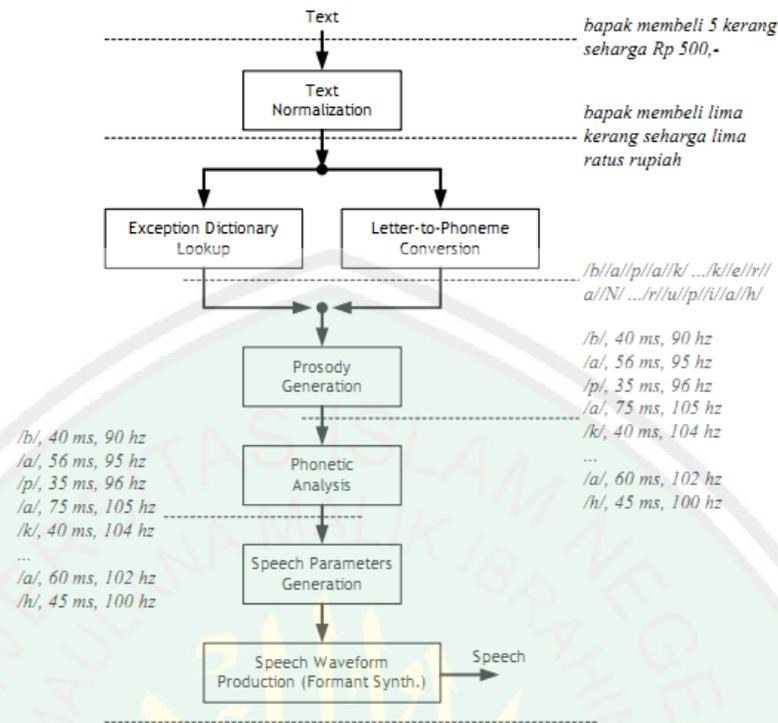
Hasil dari tahap tersebut adalah rangkaian fonem yang merepresentasikan bunyi kalimat yang ingin diucapkan. Bagian prosody generator akan melengkapi setiap unit fonem yang dihasilkan dengan data durasi pengucapannya serta pitchnya. Data durasi serta pitch diperoleh berdasarkan kombinasi antara tabel atau database

serta model prosodi. Secara simbolik, hasil dari bagian ini sudah menghasilkan informasi yang cukup untuk menghasilkan ucapan yang diinginkan.

Satu tahap berikutnya yang masih sering dilakukan adalah Phonetic Analysis. 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/.



Gambar 2.4. Urutan Proses Konversi dari Teks ke Ucapan
(dimodifikasi dari Pelton, 1992)



Gambar 2.5. Besaran-besaran Dalam Setiap Tahap Proses Konversi dari Teks ke Ucapan (dimodifikasi dari Pelton, 1992)

2.3 TTS Bahasa Indonesia

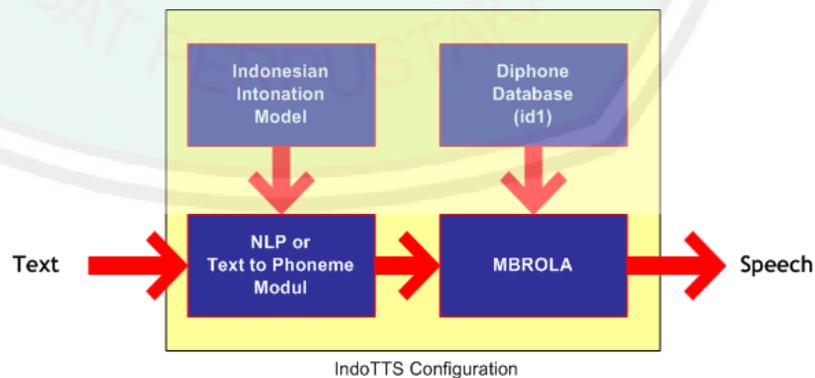
Pada saat ini telah ada TTS bahasa Indonesia yang dikenal dengan nama IndoTTS yang dikembangkan oleh Arry Akhmad Arman dosen Departemen Elektro Institut Teknologi Bandung (ITB).

IndoTTS dibangun dengan teknologi *diphone concatenation*, yaitu suatu teknik untuk membangkitkan ucapan dengan cara menyambung-nyambung *diphone* yang direkam dari ucapan manusia. *Diphone-diphone* tersebut dikumpulkan dan disimpan di dalam sebuah *diphone database* yang dalam hal ini berupa *diphone database* bahasa Indonesia.

Diphone adalah dua fonem berurutan. Dari 35 fonem akan tercipta 1.296 *diphone* yang di dalamnya sudah termasuk *diphone "silence"* untuk awal dan akhir kata. Unit terkecil bentuk tulisan adalah huruf, sedangkan unit terkecil bentuk lisan adalah fonem. Fonem tidak identik dengan huruf, contohnya gabungan huruf "n" dan "g" akan menjadi fonem "ng". Karena hasil akhir TTS adalah berupa ucapan, maka yang menjadi fokus di dalam pengembangannya adalah fonem, bukan huruf. Di dalam bahasa Indonesia, fonem yang ada berjumlah 35 buah.

Setelah *diphone database* tersedia, maka IndoTTS siap bekerja. Rangkaian huruf yang dimasukkan akan diubah menjadi rangkaian fonem lengkap dengan durasi dan frekuensinya. Data-data tersebut kemudian diumpankan ke modul Mbrola, yaitu metode *speech synthesizer* (penyusunan ucapan) dengan menggunakan metode *diphone* produk Belgia, yang kemudian akan menyambung *diphone* sesuai dengan database yang disediakan serta memanipulasi durasi dan frekuensinya. Akhirnya, keluarlah ucapan dari IndoTTS yang beraksen Indonesia.

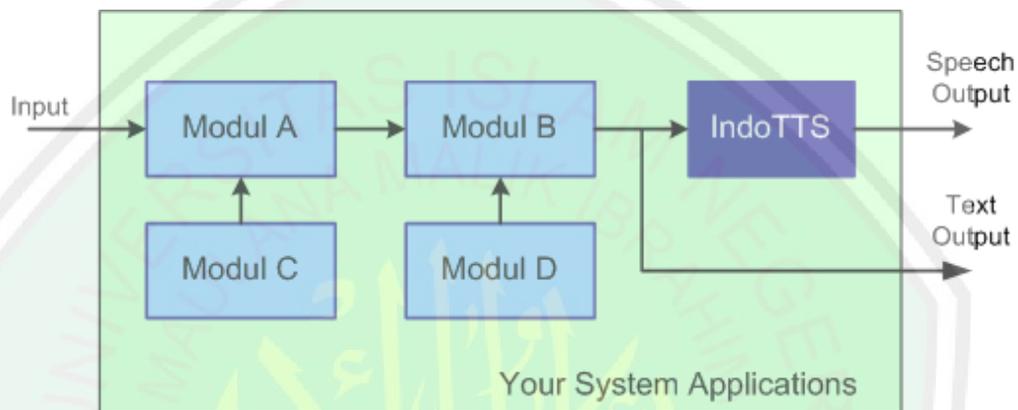
Konfigurasi sistem IndoTTS ditunjukkan pada Gambar 2.6 berikut ini:



Gambar 2.6. Konfigurasi Sistem IndoTTS

[Sumber : <http://indotts.melsa.net.id/index.html>]

IndoTTS juga dapat digabungkan dengan program lain yang berbahasa Borland Delphi, Visual Basic atau C++ dengan memanfaatkan *System Development Kit* (SDK) yang tersedia sehingga menghasilkan program aplikasi baru. Diagram pengembangannya adalah sebagai berikut :



Gambar 2.7 Diagram Pengembangan Aplikasi Menggunakan IndoTTS

[Sumber : <http://indotts.melsa.net.id/index.html>]

2.4 Fonem Bahasa Indonesia

Fonem adalah satuan bunyi bahasa terkecil yang dapat membedakan arti. Ilmu yang mempelajari tentang fonem disebut fonemik. Fonemik merupakan bagian dari fonologi. Fonologi ini khusus mempelajari bunyi bahasa. Untuk mengetahui suatu fonem harus diperlukan pasangan minimal.

Contoh: harus – arus ? /h/ adalah fonem karena membedakan arti kata harus dan arus.

Fonem dalam bahasa Indonesia terdiri atas vokal dan konsonan. Vokal adalah bunyi ujaran yang tidak mendapatkan rintangan saat dikeluarkan dari paru-

paru. Vokal dibagi menjadi dua, yaitu vokal tunggal (monoftong) yang meliputi a, i, u, e, o dan vokal rangkap (diftong), yang meliputi ai, au, oi. Konsonan adalah bunyi ujaran yang dihasilkan dari paru-paru dan mengalami rintangan saat keluarinya. Contoh konsonan antara lain p, b, m, w, f, v, t, d, n, c, j, k, g, h. Konsonan rangkap disebut kluster. Contoh kluster pada kata drama, tradisi, film, modern.

Perubahan fonem bahasa Indonesia bisa terjadi karena pengucapan bunyi ujaran memiliki pengaruh timbal balik antara fonem yang satu dengan yang lain. Macam perubahan fonem antara lain (a) alofon; (b) asimilasi; (c) desimilasi; (d) diftongisasi; (e) monoftongisasi; (f) nasalisasi.

- a. Alofon adalah variasi fonem karena pengaruh lingkungan suku kata. Contoh : simpul-simpulan. Fonem /u/ pada kata [simpul] berada pada lingkungan suku tertutup dan fonem /u/ pada kata [simpulan] berada pada lingkungan suku terbuka. Jadi, fonem /u/ mempunyai dua alofon, yaitu [u] dan (u).
- b. Asimilasi adalah proses perubahan bunyi dari tidak sama menjadi sama atau hampir sama. Contoh: in + moral ? immoral ? imoral.
- c. Desimilasi adalah proses perubahan bunyi yang sama menjadi tidak sama. Contoh : sajjana menjadi sarjana.
- d. Diftongisasi adalah perubahan monoftong menjadi diftong. Contoh: anggota menjadi anggauta.
- e. Monoftongisasi adalah proses perubahan diftong menjadi monoftong. Contoh: ramai, menjadi rame.
- f. Nasalisasi adalah persengauan atau proses memasukkan huruf nasal (n, m, ng, ny) pada suatu fonem. Contoh : me/m/ pukul menjadi memukul.

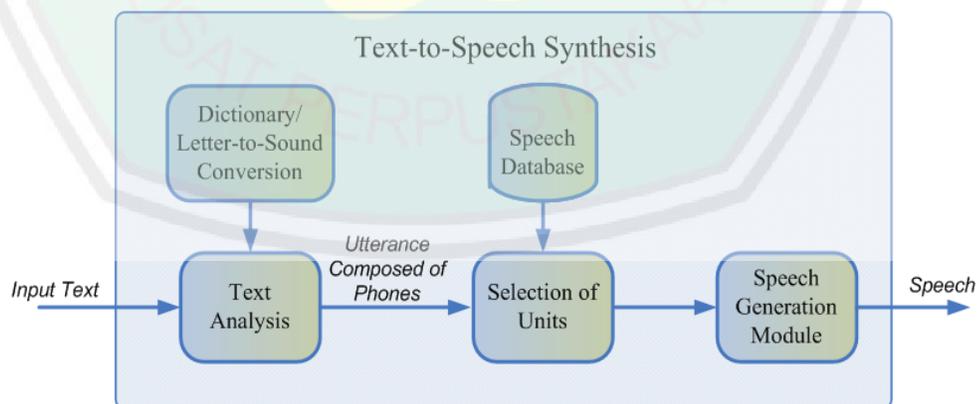
2.5 Synthesizer

Speech synthesizer dalam bahasa Indonesia diartikan sebagai sebuah sistem yang dapat mengeluarkan suara manusia sintesis. Suara yang dihasilkan dari sebuah *speech synthesizer* ini nantinya dapat dipublikasikan di hardware atau software. *Singing synthesizer* adalah sebuah sistem yang dapat menghasilkan suara manusia dengan nada yang dapat diatur sesuai partitur musik sehingga suara yang dihasilkan menyerupai suara manusia yang sedang bernyanyi.

Penelitian tentang *speech synthesis* dan *singing synthesis* sangat erat berhubungan. Namun, ada perbedaan antara nyanyian dan ucapan. Dalam nyanyian, “*voiced sound*” lebih banyak terdengar dibandingkan dengan didalam ucapan. Dalam bahasa Inggris perbedaan antara *voiced sound* pada nyanyian dan ucapan adalah 90% berbanding 60%. “*Voiced sound*” memiliki sifat yang periodik sehingga dapat lebih mudah untuk dianalisis. Perbedaan lainnya adalah dalam nyanyian, frekuensi pitch yang digunakan lebih lebar variasinya dibanding dalam mengeluarkan ucapan. Frekuensi pitch dalam nyanyian biasanya daitur terlebih dahulu sedangkan dalam ucapan, frekuensi yang tercipta secara spontan. Pada ucapan, frekuensi pitch dan durasi juga tergantung pada emosi dari pembicara. Dalam publikasi yang berjudul “*Generation of Emotion by a morphing technique in English, French and Spanish*”, Philippe Boula de Mareuil, Philippe Celener, dan Jaques Toen menjelaskan tentang variasi volume, pitch dan durasi dalam sebuah ucapan berdasarkan emosi manusia. Untuk bahasa Inggris, ketika marah, durasi pause akan memndek dan suku kata yang mengalami penekanan akan memanjang dan mendapat penambahan energi. Untuk

emosi senang, dalam pengucapan kalimat bahasa Inggris pitch rata-rata dan energi meningkat.

Speech Synthesis atau lebih dikenal dengan istilah *Text to Speech (TTS)* adalah suara buatan yang dibuat dari suara manusia sebagai *speech engine* agar dapat berkata seperti manusia untuk beragam macam aplikasi. TTS menterjemahkan teks atau kata ke dalam suara yang berguna untuk memberikan informasi dari aplikasi ke manusia. Pada awalnya, TTS dibuat dengan cara merekam suara manusia dan disimpan ke dalam database. Akan tetapi kualitas suara yang direkam tidak dapat dimengerti dan tidak terdengar dengan jelas seperti suara robot. Namun seiring dengan perkembangan teknologi, kualitas suara tersebut dapat terdengar seperti suara manusia. TTS berisi data untuk mentransformasikan teks ke suara secara *real time* dan dapat membacakan teks dengan intonasi suara serta kecepatan membaca yang jelas. Pada awal tahun 1990 – an, banyak komputer yang menerapkan *speech synthesizer* untuk memberikan kemudahan dalam pengoperasian komputer bagi orang dengan berkebutuhan.



Gambar 2.8 Diagram Sistem *Text-to-Speech Synthesis*
 Sumber : (Lazardis, Alexandros, Nikos Fakotakis, etc. , 2010)

Tahap pemrosesan terakhir dari sistem TTS adalah sintesa sinyal ucapan. Secara umum terdapat tiga metode dasar untuk sintesa sinyal ucapan. Sintesis *articulatory*, yang berusaha memodelkan sistem produksi sinyal ucapan manusia dengan pendekatan fisik mekanis secara langsung, *formant synthesizer* yang memodelkan frekuensi pole suatu sinyal ucapan atau fungsi transfer yang berbasis *vocal track* atau model *source-filter, synthesizer* perangkaian (*concatenation*), yang menggunakan panjang bagian yang berbeda dari suatu perekaman sinyal ucapan alami. Namun demikian dua teknik yang sering digunakan adalah *formant synthesizer* dan *diphone concatenation*.

formant synthesizer didasarkan pada pemodelan *source-filter*, merupakan pensintesis dengan pendekatan deskripsi *acousticphonetic*. Model dasar akustik berupa model sumber atau filter. Filter dibuat dengan beberapa set formant, yang menggambarkan artikulasi dalam suatu ucapan suara. Sumber menggambarkan *phonation*. Model ini memodelkan aliran suara (*glottal flow*) atau sinyal eksitasi *noise*. Baik sumber dan filter dikontrol oleh suatu set aturan fonetis (biasanya ratusan). Walaupun dapat menghasilkan ucapan dengan tingkat kemudahan interpretasi yang baik, namun *synthesizer* ini tidak dapat menghasilkan ucapan dengan tingkat kealamian yang tinggi.

Synthesizer perangkaian bekerja dengan merangkai unit ucapan yang telah tersimpan sebelumnya (database unit ucapan). Merangkai rekaman bagian ucapan alami merupakan cara termudah untuk membuat sinyal ucapan yang jelas dan alami. Tetapi, pesintesis perangkaian hanya terbatas pada seorang pengucap, dan satu suara, dan biasanya membutuhkan kapasitas memori yang lebih banyak daripada metode

yang lain. Kata (*word*) mungkin menjadi unit ucapan yang paling alami untuk suatu tulisan *text* tetapi dengan keterbatasan perbendaharaan kata. Perangkaian kata relatif lebih mudah untuk dibuat dan pengaruh koartikulasi antar kata dapat diatasi. Dengan banyaknya kata yang harus tercakup, maka penggunaan kata sebagai unit ucapan kurang cocok untuk sistem TTS. Jumlah suku kata yang berbeda pada suatu bahasa biasanya lebih kecil dari pada kata, tetapi banyaknya unit dalam database masih terlalu banyak untuk sistem TTS.

Modul sintesis adalah bagian akhir dari sistem TTS yang bertugas mengkonversi informasi fonetik dan prosodi yang ada menjadi sinyal ucapan.

2.6 MBROLA

Mbrola adalah sebuah aplikasi yang menyediakan multi bahasa dengan Synthesizer (sintesis pengolah suara).

Tujuan dari proyek MBROLA, diprakarsai oleh Lab TCTS dari Faculté Polytechnique de Mons (Belgia), adalah untuk mendapatkan satu set synthesizer bagi banyak pidato bahasa yang mungkin, dan menyediakan mereka gratis untuk aplikasi non-komersial. The ultimate goal is to boost academic research on speech synthesis, and particularly on prosody generation, known as one of the biggest challenges taken up by Text-To-Speech synthesizers for the years to come. Tujuan utamanya adalah untuk meningkatkan penelitian akademis tentang sintesis pidato, dan terutama pada generasi ilmu persajakan, yang dikenal sebagai salah satu tantangan terbesar yang diambil oleh synthesizer Text-To-Speech untuk tahun yang akan datang.

Dibutuhkan daftar fonem sebagai input, bersama dengan informasi prosodi (durasi fonem dan deskripsi sesepenggal linear pitch), dan menghasilkan contoh

pidato pada 16 bit (linear), pada frekuensi sampling dari diphone database yang digunakan (oleh karena itu BUKAN Text-To-Speech (TTS) synthesizer, karena tidak menerima teks baku sebagai input). synthesizer ini disediakan secara gratis, untuk non komersial, aplikasi militer non saja.

Diphone database disesuaikan dengan format Mbrola diperlukan untuk menjalankan synthesizer,. Perancis Sebuah suara telah disediakan oleh penulis MBROLA dan proyek MBROLA sendiri telah disusun sedemikian rupa sehingga menghasut laboratorium penelitian lain atau perusahaan untuk berbagi database diphone mereka. The terms of this sharing policy can be summarized as follows :
Ketentuan kebijakan sharing dapat diringkas sebagai berikut:

Setelah beberapa perjanjian resmi antara penulis MBROLA dan pemilik database diphone, database diproses oleh penulis dan disesuaikan dengan format Mbrola, gratis. Database Mbrola diphone yang dihasilkan dibuat tersedia untuk non-komersial, penggunaan non-militer sebagai bagian dari proyek MBROLA. Hak Komersial di database Mbrola tetap dengan penyedia database untuk penggunaan eksklusif dengan software Mbrola.

2.7 Database Diphone MBROLA

MBROLA adalah proyek pensintesa ucapan berkualitas tinggi yang diprekarsai Oleh Dr. Thierry Dutoit dari *Faculte Polytechnique de Mons, TCTS Lab, Belgia*. Tujuan utama proyek ini untuk meningkatkan penelitian akademika pada bidang pensintesa ucapan, khususnya yang berkaitan dengan *prosody*. MBROLA menggunakan teknik *diphone* yang direkam dalam format 16 *bit* sehingga kualitas suara yang dihasilkan sangat baik.

Diphone adalah dua unit *fonem* yang disusun berurutan. Teknik penyusunan fonem berurut ini disebut *diphone concatenation*. Suatu *database diphone* menyimpan semua kemungkinan pasangan fonem dalam satu file, sehingga jumlah *diphone* dalam satu *database diphone* dapat mencapai ratusan sampai ribuan *diphone*. Jumlah daftar fonem dalam suatu *database diphone* tergantung dari hasil penelitian masing-masing pembuat *database*.

Database diphone bahasa Indonesia (ID1) dibuat tahun 2000 oleh Dr. Arry Akhmad Arman dari institut Teknologi Bandung, memiliki 30 fonem termasuk spasi. Tabel fonem untuk *database* ID1 terdapat pada Tabel dibawah ini :

Tabel 2.1 Fonem Database Diphone ID1

[Sumber : Speech Sythesizer Bahasa Indonesia Berbasis Diphone MBROLA oleh stephanus priyowidodo dan dedi irawan]

FONEM	SAMPA	CONTOH KATA
p	P	<u>p</u> anas, apa, siap
b	B	<u>b</u> aca, sebab
t	T	<u>t</u> anam, a <u>t</u> ap, penat
D	D	<u>d</u> atang, tad <u>i</u>
K	K	<u>k</u> ami, a <u>k</u> an, tidak
G	G	<u>g</u> ula, agar, reog
c	tS	<u>c</u> ari, renc <u>a</u> na,
j	dZ	<u>j</u> adi, be <u>j</u> ana
f	F	<u>f</u> aktor, kaf <u>a</u> n, alif
s	S	<u>s</u> aran, ka <u>s</u> ar, ber <u>a</u> s
z	Z	<u>z</u> ebra, azim <u>u</u> th
H	H	<u>h</u> itam, se <u>h</u> at, sa <u>h</u>
M	M	<u>m</u> usim, a <u>m</u> an, hit <u>a</u> m
N	N	<u>n</u> usa, a <u>n</u> da, hut <u>a</u> n
ng	N	<u>a</u> ngka, si <u>a</u> ng
R	R	<u>r</u> obot, a <u>r</u> mada, sona <u>r</u>
L	L	<u>l</u> ima, a <u>l</u> amat, kes <u>a</u> l

FONEM	SAMPA	CONTOH KATA
w	W	<u>w</u> aktu, ba <u>w</u> a
y	J	<u>y</u> akin, ba <u>y</u> aya
ny	nY	<u>n</u> yaris, ke <u>n</u> yang
a	V	<u>a</u> kh <u>i</u> r, ba <u>g</u> us, n <u>i</u> ra
e	@	<u>e</u> nggan, pe <u>t</u> ang
e	E	<u>e</u> sa, me <u>d</u> an, bu <u>l</u> e
i	I	<u>i</u> kal, bi <u>l</u> a, ba <u>l</u> i
o	Q	<u>o</u> rang, mo <u>h</u> on, joku
u	U	<u>u</u> saha, bu <u>k</u> an, bi <u>r</u> u
ai	aI	sant <u>a</u> i, land <u>a</u> i
oi	OI	Ambo <u>i</u>
au	aU	lamp <u>a</u> u, pul <u>a</u> u
Spasi	-	-

2.8 Screen Reader

Screen Reader adalah merupakan sebuah piranti lunak (*software*) yang berguna untuk membantu orang-orang yang menderita kelemahan dalam penglihatan (*low vision*) menggunakan komputer.

Screen reader sengaja dibuat untuk orang-orang yang menderita kelemahan dalam penglihatan ("low vision") sehingga mereka mudah menggunakan Microsoft Windows secara personal. Dengan alat ini tentunya penderita tunanetra dan penderita lemah daya penglihatan (*low vision*) mudah mengakses komputer dan bahkan bisa melepaskan ketergantungan pada orang lain dalam menggunakannya.

2.8.1 Perkembangan Screen reader di Indonesia

Menurut Abimanyu dari Yayasan Mitra Netra penggunaan screen reader di Indonesia diawali pada tahun 1990-an. Screen reader masuk ke Indonesia dan penggunaannya dikawal oleh Yayasan Mitra Netra yang berdiri sejak 14 Mei 1991

atas gagasan beberapa tunanetra yang menyadari kesadaran untuk kemudahan akses bagi sesamanya. Karena kesadaran inilah akhirnya screen reader masuk ke Indonesia. Pada awal masuknya pemanfaatan screen reader baru sebatas pada penggunaan untuk Microsoft Office karena saat itu sistem internet belum siap dalam penggunaan JAWS screen reader. Karena harga screen reader yang cukup mahal penyebarannya di Indonesia cukup tersendat, terhitung hingga saat ini yayasan yang menyediakan akses komputer dan internet untuk tunanetra (dengan screen reader) baru dua, yaitu Yayasan Mitra Nusantara dan Yayasan Kartika Destarata. Yayasan Mitra Netra menyelenggarakan kursus komputer bicara (komputer dengan screen reader) untuk para tunanetra. Peserta kursus didominasi oleh siswa dan mahasiswa tunanetra yang sedang menempuh pendidikan secara inklusif di sekolah umum serta perguruan tinggi. Barulah pada tahun 1999, Yayasan Mitra Netra mulai merentangkan sayapnya dengan program kursus serupa di Yayasan Mitra Netra Perwakilan Bandung. Cara yang digunakan untuk memperluas akses tunanetra di seluruh Indonesia terhadap teknologi komputer dan Internet adalah melalui kerja sama dengan Microsoft Indonesia, pada tahun 2003, Yayasan Mitra Netra mendirikan Community Training and Learning Center (CTLIC) di beberapa organisasi ketunanetraan dan Sekolah Luar Biasa (SLB) untuk tunanetra di Jakarta, Bandung, Medan, dan Makasar. Melalui CTLIC yang terdiri dari lima lembaga ini (Yayasan Mitra Netra Jakarta, Kartika Destarata Jakarta, Yayasan Mitra Netra Bandung, YAPTI Makasar dan Yapentra Medan), Yayasan Mitra Netra menyelenggarakan program pelatihan komputer bicara bagi generasi muda tunanetra.

2.9 Visual Basic

Bahasa Basic pada dasarnya adalah bahasa yang mudah dimengerti sehingga pemrograman di dalam bahasa Basic dapat dengan mudah dilakukan meskipun oleh orang yang baru belajar membuat program. Hal ini lebih mudah lagi setelah hadirnya Microsoft Visual Basic, yang dibangun dari ide untuk membuat bahasa yang sederhana dan mudah dalam pembuatan scriptnya (simple scripting language) untuk graphic user interface yang dikembangkan dalam sistem operasi Microsoft Windows.

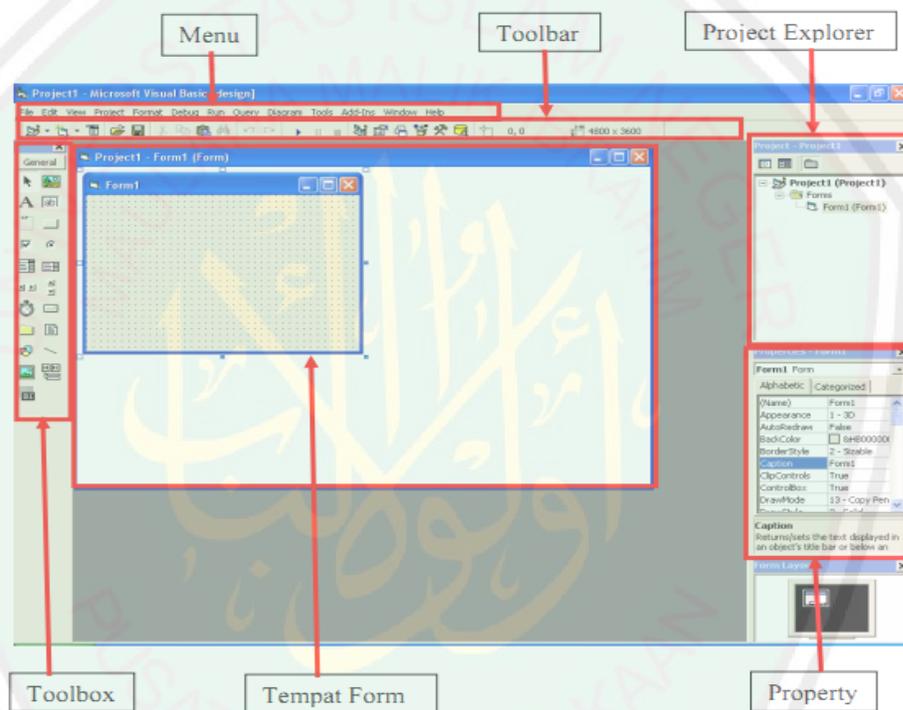
Visual Basic merupakan bahasa pemrograman yang sangat mudah dipelajari, dengan teknik pemrograman visual yang memungkinkan pengguna untuk berkreasi lebih baik dalam menghasilkan suatu program aplikasi. Ini terlihat dari dasar pembuatan dalam visual basic adalah FORM, dimana pengguna dapat mengatur tampilan form kemudian dijalankan dalam script yang sangat mudah.

Ledakan pemakaian Visual Basic ditandai dengan kemampuan Visual Basic untuk dapat berinteraksi dengan aplikasi lain di dalam sistem operasi Windows dengan komponen ActiveX Control. Dengan komponen ini memungkinkan pengguna untuk memanggil dan menggunakan semua model data yang ada di dalam sistem operasi windows. Hal ini juga ditunjang dengan teknik pemrograman di dalam Visual Basic yang mengadopsi dua macam jenis pemrograman yaitu Pemrograman Visual dan Object Oriented Programming (OOP).

Visual Basic 6.0 sebetulnya perkembangan dari versi sebelumnya dengan beberapa penambahan komponen yang sedang tren saat ini, seperti kemampuan pemrograman internet dengan DHTML (Dynamic HyperText Mark Language), dan beberapa penambahan fitur database dan multimedia yang semakin baik. Sampai saat

buku ini ditulis bisa dikatakan bahwa Visual Basic 6.0 masih merupakan pilihan pertama di dalam membuat program aplikasi yang ada di pasar perangkat lunak nasional. Hal ini disebabkan oleh kemudahan dalam melakukan proses development dari aplikasi yang dibuat.

Interface antar muka Visual Basic 6.0, berisi menu, toolbar, toolbox, form, project explorer dan property seperti terlihat pada gambar . berikut:



Gambar 2.9. Interface antar muka Visual Basic 6.0

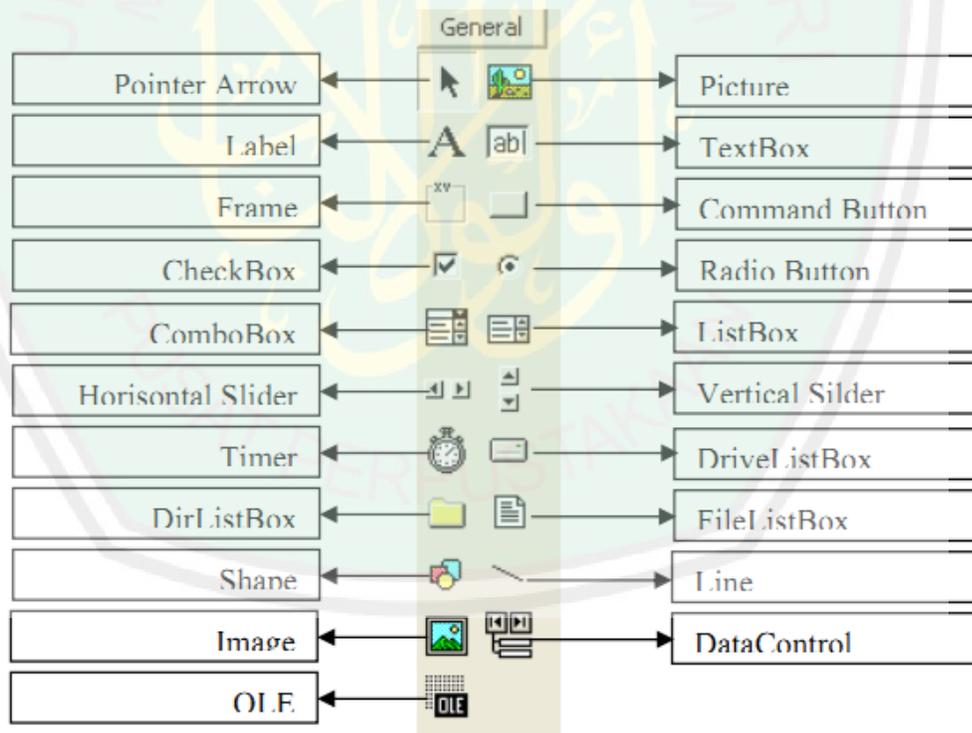
[Sumber : Algoritma Pemrograman 2 Visual basic, Ahmad Basuki]

Pembuatan program aplikasi menggunakan Visual Basic dilakukan dengan membuat tampilan aplikasi pada form, kemudian diberi script program di dalam komponen-komponen yang diperlukan. Form disusun oleh komponen-komponen

yang berada di [Toolbox], dan setiap komponen yang dipakai harus diatur propertinya lewat jendela [Property].

Menu pada dasarnya adalah operasional standar di dalam sistem operasi windows, seperti membuat form baru, membuat project baru, membuka project dan menyimpan project. Di samping itu terdapat fasilitas-fasilitas pemakaian visual basic pada menu. Untuk lebih jelasnya Visual Basic menyediakan bantuan yang sangat lengkap dan detail dalam MSDN.

Toolbox berisi komponen-komponen yang bisa digunakan oleh suatu project aktif, artinya isi komponen dalam toolbox sangat tergantung pada jenis project yang dibangun. Komponen standar dalam toolbox dapat dilihat pada gambar berikut ini.



Gambar 2.10 Komponen standar dalam Toolbox

[Sumber : Algoritma Pemrograman 2 Visual basic, Ahmad Basuki]

Konsep dasar pemrograman Visual Basic 6.0, adalah pembuatan form dengan mengikuti aturan pemrograman Property, Metode dan Event. Hal ini berarti:

- 1) Property: Setiap komponen di dalam pemrograman Visual Basic dapat diatur propertinya sesuai dengan kebutuhan aplikasi. Property yang tidak boleh dilupakan pada setiap komponen adalah "Name", yang berarti nama variabel (komponen) yang akan digunakan dalam scripting. Properti "Name" ini hanya bisa diatur melalui jendela Property, sedangkan nilai peroperti yang lain bisa diatur melalui script seperti

```
Command1.Caption="Play"  
Text1.Text="Visual Basic"  
Label1.Visible=False  
Timer1.Enable=True
```

- 2) Metode: Bahwa jalannya program dapat diatur sesuai aplikasi dengan menggunakan metode pemrograman yang diatur sebagai aksi dari setiap komponen. Metode inilah tempat untuk mengekspresikan logika pemrograman dari pembuatan suatu prgram aplikasi.
- 3) Event: Setiap komponen dapat beraksi melalui event, seperti event click pada command button yang tertulis dalam layar script Command1_Click, atau event Mouse Down pada picture yang tertulis dengan Picture1_MouseDown.

Pengaturan event dalam setiap komponen yang akan menjalankan semua metode yang dibuat.

2.10 Pemrograman API

API atau (*Application Programming Interface*) merupakan fungsi-fungsi eksternal yang terdapat di dalam file-file pustaka windows (*library Windows*) yang dapat dimanfaatkan oleh suatu aplikasi.

Setiap tugas (task) yang dilakukan oleh windows biasanya memanfaatkan fungsi API yang berada di dalam file “.dll”. File .dll ini umumnya berada pada folder system (*C:Windows|Winnt|System|System32*). Pada pemrograman API, kita dapat memanfaatkan fungsi-fungsi yang terdapat pada file .dll tersebut, antara lain :

- a) Base Service, menyediakan fitur-fitur untuk mengakses sistem operasi seperti file sistem, devices, processes and threads, mengakses registry, dan menangani error. Fungsi-fungsi ini berada di *kernel.exe*, *krnl286.exe* atau *krnl386.exe*, untuk Windows 16-bit. *kernel32.dll* dan *advapi32.dll* di Windows 32-bit.
- b) Graphics Device Interface, menyediakan kemampuan untuk tampilan grafik ke monitor, printer, dan device lainnya. Fungsi ini berada di *gdi.exe* pada Windows 16-bit dan *gdi32.dll* pada Windows 32-bit.
- c) User Interface, menyediakan kemampuan untuk membuat serta mengatur tampilan dan juga mengatur kontrol lainnya, seperti scrollbar, tombol-tombol, menerima masukan dari mouse dan keyboard, dan lainnya yang berhubungan dengan GUI (*Graphic User Interface*). Fungsi ini berada di *user.exe* pada

Windows 16-bit, dan *user32.dll* pada Windows 32-bit. Sejak versi Windows XP, fungsi ini berada di *comctl32.dll*.

- d) Common Dialog Box Library, menyediakan aplikasi kotak dialog untuk membuka dan menyimpan file, memilih bentuk huruf (*font*), dan lain sebagainya. Fungsi ini berada di *commdlg.dll* pada Windows 16-bit, dan *comdlg32.dll* pada Windows 32-bit.
- e) Common Control Library, berfungsi menyediakan akses suatu aplikasi ke beberapa kontrol yang disediakan oleh sistem operasi, seperti status bar, progress bars, toolbars, dan tabs. Fungsi ini berada di *commctrl.dll* pada Windows 16-bit, dan *comctl32.dll* pada Windows 32-bit.
- f) Windows Shell, merupakan komponen Windows API yang memberi akses suatu aplikasi ke fungsi-fungsi yang disediakan oleh shell dari sistem operasi (*operating system shell*). Fungsi ini berada di *shell.dll* pada Windows 16-bit, dan *shell32.dll* dan *shlwapi.dll* pada Windows 32-bit.
- g) Network Services, memberikan akses ke berbagai perangkat jaringan, seperti *NetBIOS*, *Winsock*, dan lain sebagainya.

Ada dua langkah yang diperlukan untuk fungsi Windows API:

- a. Menyatakan fungsi dalam modul atau bentuk.
- b. Memanggil fungsi dalam program Anda seperti yang akan Anda fungsi lain.

Topik "Menyatakan prosedur DLL" dalam "bagian 4 mengakses dll dan Windows API "dari"Visual Basic komponen alat panduan"menjelaskan bagaimana menyatakan fungsi dalam modul. Topik juga menunjukkan bagaimana untuk

menggunakan API Peninjau teks untuk menemukan sesuai Deklarasi dan copy dan paste Deklarasi untuk proyek Anda.

Sebagai contoh, untuk menyatakan GetCaretBlinkTime fungsi dalam modul, salin teks berikut ke jendela kode modul:

```
Declare Function GetCaretBlinkTime Lib "user32" Alias _
    "GetCaretBlinkTime" () As Long
```

Jika ingin menggunakan fungsi yang sama hanya dalam bentuk tertentu, tambahkan kata kunci "Swasta" ke awal deklarasi fungsi. The Deklarasi fungsi GetCaretBlinkTime ditulis dalam jendela kode bentuk sebagai berikut:

```
Private Declare Function GetCaretBlinkTime Lib "user32" Alias _
    "GetCaretBlinkTime" () As Long
```

Fungsi ini mengembalikan nilai panjang yang kursor berkedip tingkat di milidetik. Kata kunci Lib mendahului nama file yang berisi fungsi; dalam kasus ini, User32.DLL. Kata kunci Alias digunakan untuk mendefinisikan fungsi spesifik dalam DLL file. Klausul ini opsional ketika fungsi yang dinyatakan dengan nama yang sama seperti dalam DLL.

Akan sering melihat notasi pribadi yang digunakan dalam Basis Pengetahuan Microsoft Artikel karena tidak menggunakan modul arus pelaksanaan contoh kode. Jika perlu untuk menggunakan fungsi Windows API di lebih dari satu file sumber dalam proyek, itu harus dinyatakan dalam modul.

Untuk informasi lebih lanjut tentang pernyataan menetapkan, lihat "Menyatakan pernyataan" dalam bahasa referensi dalam buku secara Online.

Setelah telah menyatakan fungsi, Anda dapat menyebutnya seperti yang akan Anda salah Visual Basic fungsi dan pass apapun sesuai parameter. Berikut Contoh menunjukkan bagaimana memanggil fungsi GetCaretBlinkTime:

```
Dim lngCaretBlinkTime as Long ' Blink Rate variable
lngCaretBlinkTime = GetCaretBlinkTime
```

Untuk fungsi yang membutuhkan parameter, dapat melewati parameter referensi atau dengan nilai.

Parameter berlalu dengan referensi melewati 32-bit alamat lokasi di mana nilai disimpan. Secara default, Visual Basic lewat parameter referensi. Dapat juga secara eksplisit melewati parameter referensi menggunakan ByRef kata kunci.

Nilai-nilai parameter yang disahkan oleh referensi dapat berubah dalam disebut prosedur. Misalnya, fungsi InvertRect membalik tertentu persegi panjang dengan dengan nilai setiap pixel. Fungsi memerlukan menangani dengan konteks perangkat yang mengandung persegi panjang dan alamat pengguna didefinisikan variabel tipe (UDT) dengan koordinat logis persegi panjang. Alamat variabel buatan pengguna disahkan oleh referensi.

Ketika memanggil fungsi ini, adalah logis koordinat persegi panjang ditempatkan dalam variabel yang ditetapkan pengguna.

Untuk menggunakan fungsi ini, tambahkan berikut Deklarasi untuk modul:

```
Public Type RECT
    Left As Long
    Top As Long
    Right As Long
```

```

Bottom As Long

End Type

Declare Function InvertRect Lib "user32" Alias "InvertRect" _
    (ByVal hdc As Long, ByVal lpRect As RECT) As Long

```

Deklarasi pertama menentukan jenis ditetapkan pengguna yang diperlukan untuk parameter fungsi InvertRect sementara deklarasi yang kedua untuk fungsi.

Melewati nilai parameter mengirimkan nilai sebenarnya dari parameter untuk fungsi. Melewati parameter nilai Jika parameter tidak diubah oleh prosedur yang disebut. Untuk melewati parameter nilai, tambahkan kata kunci ByVal untuk setiap parameter ingin lulus untuk fungsi.

Fungsi SetCaretBlinkTime memerlukan tingkat dalam milidetik parameter untuk diteruskan oleh nilai fungsi. Menyatakan fungsi ini dengan menambahkan pernyataan berikut untuk modul:

```

Declare Function SetCaretBlinkTime Lib "user32" Alias _
    "SetCaretBlinkTime" (ByVal wMSeconds As Long) As Long

```

Untuk menggunakan fungsi ini untuk mengatur kursor blink rate untuk 250 milidetik, tambahkan berikut untuk proyek :

```

Dim IBlinkRate as Long ' Blink rate variable
Dim IResult as Long ' Result Variable

IBlinkRate = 250

IResult = SetCaretBlinkTime (IBlinkRate)

If IResult = 0 Then
    ' Error code

```

End If

2.11 UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem software (<http://www.omg.org>).

Pendekatan analisa & rancangan dengan menggunakan model OO mulai diperkenalkan sekitar pertengahan 1970 hingga akhir 1980 dikarenakan pada saat itu aplikasi software sudah meningkat dan mulai kompleks. Jumlah yang menggunakan metoda OO mulai diuji cobakan dan diaplikasikan antara 1989 hingga 1994, seperti halnya oleh Grady Booch dari *Rational Software Co.*, dikenal dengan OOSE (*Object-Oriented Software Engineering*), serta James Rumbaugh dari *General Electric*, dikenal dengan OMT (*Object Modelling Technique*).

Kelemahan saat itu disadari oleh Booch maupun Rumbaugh adalah tidak adanya standar penggunaan model yang berbasis OO, ketika mereka bertemu ditemani rekan lainnya Ivar Jacobson dari *Objectory* mulai mendiskusikan untuk mengadopsi masing-masing pendekatan metoda OO untuk membuat suatu model bahasa yang uniform / seragam yang disebut UML (*Unified Modeling Language*) dan dapat digunakan oleh seluruh dunia.

Secara resmi bahasa UML dimulai pada bulan oktober 1994, ketika Rumbaugh bergabung Booch untuk membuat sebuah project pendekatan metoda yang uniform/seragam dari masing-masing metoda mereka. Saat itu baru dikembangkan draft metoda UML version 0.8 dan diselesaikan serta di release pada bulan oktober 1995. Bersamaan dengan saat itu, Jacobson bergabung dan UML tersebut diperkaya ruang lingkupnya dengan metoda OOSE sehingga muncul release version 0.9 pada bulan Juni 1996. Hingga saat ini sejak Juni 1998 UML version 1.3 telah diperkaya dan direspons oleh OMG (Object Management Group), Anderson Consulting, Ericsson, Platinum Technology, ObjectTime Limited, dll serta di pelihara oleh OMG yang dipimpin oleh Cris Kobryn.

UML adalah standar dunia yang dibuat oleh *Object Management Group* (OMG), sebuah badan yang bertugas mengeluarkan standar-standar teknologi *object oriented* dan *software component*.

2.12 Pengenalan UML

UML sebagai sebuah bahasa yang memberikan *vocabulary* dan tatanan penulisan kata-kata dalam 'MS Word' untuk kegunaan komunikasi. Sebuah bahasa model adalah sebuah bahasa yang mempunyai *vocabulary* dan konsep tatanan / aturan penulisan serta secara fisik mempresentasikan dari sebuah sistem. Seperti halnya UML adalah sebuah bahasa standard untuk pengembangan sebuah software yang dapat menyampaikan bagaimana membuat dan membentuk model-model, tetapi tidak menyampaikan apa dan kapan model yang seharusnya dibuat yang merupakan salah satu proses implementasi pengembangan software.

UML tidak hanya merupakan sebuah bahasa pemrograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemrograman, seperti JAVA, C++, Visual Basic, atau bahkan dihubungkan secara langsung ke dalam sebuah object-oriented database. Begitu juga mengenai pendokumentasian dapat dilakukan seperti; *requirements*, arsitektur, *design*, *source code*, *project plan*, *tests*, dan *prototypes*.

Untuk dapat memahami UML membutuhkan bentuk konsep dari sebuah bahasa model, dan mempelajari 3 (tiga) elemen utama dari UML seperti *building block*, aturan-aturan yang menyatakan bagaimana *building block* diletakkan secara bersamaan, dan beberapa mekanisme umum (common).

A. Building blocks

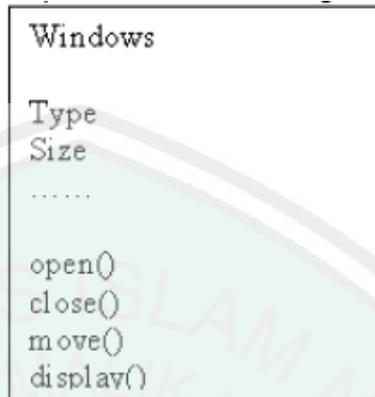
3 (tiga) macam yang terdapat dalam building block adalah kategori benda/Things, hubungan, dan diagram. Benda/things adalah abstraksi yang pertama dalam sebuah model, hubungan sebagai alat komunikasi dari benda-benda, dan diagram sebagai kumpulan / group dari benda-benda/things.

1. Benda/Things

Adalah hal yang sangat mendasar dalam model UML, juga merupakan bagian paling statik dari sebuah model, serta menjelaskan elemen-elemen lainnya dari sebuah konsep dan atau fisik. Bentuk dari beberapa benda/thing adalah sebagai berikut:

Pertama, adalah sebuah kelas yang diuraikan sebagai sekelompok dari object yang mempunyai attribute, operasi, hubungan yang semantik. Sebuah kelas mengimplementasikan 1 atau lebih interfaces. Sebuah kelas dapat digambarkan

sebagai sebuah persegi panjang, yang mempunyai sebuah nama, attribute, dan metoda pengoperasiannya, seperti terlihat dalam gambar 2.11.



Gambar 2.11. Sebuah Kelas dari model UML

Kedua, yang menggambarkan *'interface'* merupakan sebuah antar-muka yang menghubungkan dan melayani antar kelas dan atau elemen. *'Interface'* / antarmuka mendefinisikan sebuah set / kelompok dari spesifikasi pengoperasian, umumnya digambarkan dengan sebuah lingkaran yang disertai dengan namanya.

Sebuah antar-muka berdiri sendiri dan umumnya merupakan pelengkap dari kelas atau komponen, seperti dalam gambar 2.12



Gambar 2.12 Sebuah interface/antar-muka

Ketiga, adalah *collaboration* yang didefinisikan dengan interaksi dan sebuah kumpulan / kelompok dari kelas-kelas/lemen-lemen yang bekerja secara bersama-sama. *Collaborations* mempunyai struktura dan dimensi. Pemberian sebuah kelas

memungkinkan berpartisipasi didalam beberapa *collaborations* dan digambarkan dengan sebuah ‘*elips*’ dengan garis terpotong-potong.



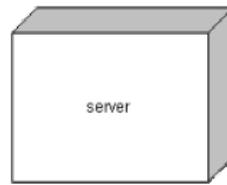
Gambar 2.13. Collaborations

Keempat, sebuah ‘*use case*’ adalah rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. ‘*use case*’ digunakan untuk membentuk tingkah-laku benda/ *things* dalam sebuah model serta di realisasikan oleh sebuah *collaboration*. Umumnya ‘*use case*’ digambarkan dengan sebuah ‘*elips*’ dengan garis yang solid, biasanya mengandung nama, seperti terlihat dalam gambar 2.14.



Gambar2.14. Use Case

Kelima, sebuah *node* merupakan fisik dari elemen-elemen yang ada pada saat dijalankannya sebuah sistem, contohnya adalah sebuah komputer, umumnya mempunyai sedikitnya *memory* dan *processor*. Sekelompok komponen mungkin terletak pada sebuah *node* dan juga mungkin akan berpindah dari *node* satu ke *node* lainnya. Umumnya *node* ini digambarkan seperti kubus serta hanya mengandung namanya, seperti terlihat dalam gambar 2.15.



Gambar 2.15. Nodes

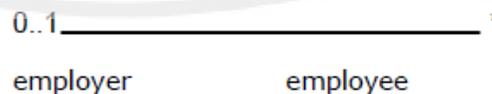
2. Hubungan / Relationship

Ada 4 macam hubungan didalam penggunaan UML, yaitu; *dependency*, *association*, *generalization*, dan *realization*.

Pertama, sebuah *dependency* adalah hubungan semantik antara duabenda/things yang mana sebuah benda berubah mengakibatkan benda satunya akan berubah pula. Umumnya sebuah *dependency* digambarkan sebuah panah dengan garis terputus-putus seperti terlihat dalam gambar 6.

Gambar 2.16. *Dependency*

Kedua, sebuah *association* adalah hubungan antar benda struktural yang terhubung diantara obyek. Kesatuan obyek yang terhubung merupakan hubungan khusus, yang menggambarkan sebuah hubungan struktural diantara seluruh atau sebagian. Umumnya *association* digambarkan dengan sebuah garis yang dilengkapi dengan sebuah label, nama, dan status hubungannya seperti terlihat dalam gambar 2.17.



Gambar 2.17. Association

Ketiga, sebuah *generalization* adalah menggambarkan hubungan khusus dalam obyek anak/*child* yang menggantikan obyek *parent* / induk . Dalam hal ini,

obyek anak memberikan pengaruhnya dalam hal struktur dan tingkah lakunya kepada obyek induk. Digambarkan dengan garis panah seperti terlihat dalam gambar 2.18.



Gambar 2.18. Generalizations

Keempat, sebuah *realization* merupakan hubungan semantik antara pengelompokan yang menjamin adanya ikatan diantaranya. Hubungan ini dapat diwujudkan diantara *interface* dan kelas atau *elements*, serta antara *use cases* dan *collaborations*. Model dari sebuah hubungan realization seperti terlihat dalam gambar 2.19.



Gambar 2.19. Realizations

3. Diagram

UML sendiri terdiri atas pengelompokan **diagram-diagram** sistem menurut aspek atau sudut pandang tertentu. Diagram adalah yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model. UML mempunyai 9 diagram, yaitu; *use-case*, *class*, *object*, *state*, *sequence*, *collaboration*, *activity*, *component*, dan *deployment diagram*. Diagram pertama adalah *use case* menggambarkan sekelompok *use cases* dan aktor yang disertai dengan hubungan diantaranya. Diagram *use cases* ini menjelaskan dan menerangkan kebutuhan / requirement yang diinginkan/dikehendaki *user*/pengguna, serta sangat berguna dalam menentukan struktur organisasi dan model dari pada sebuah sistem.

2.13 MSAA

Microsoft Active Accessibility (MSAA) merupakan Application Programming Interface (API) untuk aksesibilitas antarmuka pengguna. MSAA diperkenalkan sebagai platform add-on untuk Microsoft Windows 95 pada tahun 1997. MSAA dirancang untuk membantu Assistive Technology (AT) berinteraksi dengan antarmuka standar dan custom user interface (UI) elemen dari aplikasi (atau sistem operasi), serta untuk mengakses, mengidentifikasi, dan memanipulasi elemen UI sebuah aplikasi. AT produk bekerja dengan *MSAA Enable Application* untuk menyediakan akses yang lebih baik bagi individu yang mengalami kesulitan, gangguan, atau cacat fisik atau kognitif. Beberapa contoh produk AT pembaca layar untuk pengguna dengan penglihatan terbatas, pada keyboard layar untuk pengguna dengan akses fisik yang terbatas, atau narrator untuk pengguna dengan pendengaran terbatas. MSAA juga dapat digunakan untuk alat pengujian otomatis, dan aplikasi pelatihan berbasis komputer.

Spesifikasi saat ini dan terbaru MSAA ditemukan dalam bagian dari *Microsoft UI Automation Community Promise Specification*.

2.14 Bahasa dalam kajian Islam

Tidak ada sejarah satu bahasa tanpa ada sejarah manusia. Kapan dan dimana manusia berada dan diciptakan, maka saat itulah bahasa ada. Bahasa tercipta bersamaan dengan penciptaan manusia. Sebagaimana dijelaskan dalam Al-qur'an dalam surah Al-baqarah ayat 31-33 :

وَعَلَّمَ آدَمَ الْأَسْمَاءَ كُلَّهَا ثُمَّ عَرَضَهُمْ عَلَى الْمَلَائِكَةِ فَقَالَ أَنْذِئُونِي بِأَسْمَاءِ
هَٰؤُلَاءِ إِنْ كُنْتُمْ صَادِقِينَ ﴿٣١﴾

Artinya:

Dan Dia mengajarkan kepada Adam nama-nama (benda-benda) seluruhnya, kemudian mengemukakannya kepada para Malaikat lalu berfirman: "Sebutkanlah kepada-Ku nama benda-benda itu jika kamu mamang benar orang-orang yang benar!" (QS. Al-baqarah: 31)

قَالُوا سُبْحَانَكَ لَا عِلْمَ لَنَا إِلَّا مَا عَلَّمْتَنَا إِنَّكَ أَنْتَ الْعَلِيمُ
الْحَكِيمُ ﴿٣٢﴾

Artinya:

Mereka menjawab: "Maha Suci Engkau, tidak ada yang kami ketahui selain dari apa yang telah Engkau ajarkan kepada kami; sesungguhnya Engkaulah Yang Maha Mengetahui lagi Maha Bijaksana". (QS. Al-baqarah: 2)

قَالَ يَا آدَمُ أَنْذِئُهُمْ بِأَسْمَائِهِمْ فَلَمَّا أَنْذَبَهُمْ بِأَسْمَائِهِمْ قَالَ أَلَمْ أَقُلْ لَكُمْ
إِنِّي أَعْلَمُ غَيْبَ السَّمَوَاتِ وَالْأَرْضِ وَأَعْلَمُ مَا تُبْدُونَ وَمَا كُنْتُمْ
تَكْتُمُونَ ﴿٣٣﴾

Artinya :

Allah berfirman: "Hai Adam, beritahukanlah kepada mereka nama-nama benda ini." Maka setelah diberitahukannya kepada mereka nama-nama benda itu, Allah berfirman: "Bukankah sudah Ku katakan kepadamu, bahwa sesungguhnya Aku mengetahui rahasia langit dan bumi dan mengetahui apa yang kamu lahirkan dan apa yang kamu sembunyikan?" (QS. Al-baqarah: 33)

Dialog antara Allah, Malaikat dan Adam di atas memberikan sebuah pemahaman tentang asal usul dan sejarah manusia. Ayat di atas juga hendak

menegaskan bahwa manusia berbeda dengan hewan atau makhluk lainnya bahkan dengan malaikat sekalipun. Dengan bahasa dan akal, manusia menciptakan berbagai karya kebudayaan dan membentuk peradaban. Bahasa dan akal merupakan factor penentu realisasi amanah Allah bagi manusia sebagai khalifah di bumi.

Dilihat dari sisi fungsi bahasa merupakan sarana komunikasi mempunyai fungsi utama bahasa adalah bahwa komunikasi ialah penyampaian pesan atau makna oleh seseorang kepada orang lain. Sebagaimana yang difirmankan Allah dalam surah Ibrahim ayat 4 sebagai berikut :

وَمَا أَرْسَلْنَا مِنْ رَّسُولٍ إِلَّا بِلِسَانِ قَوْمِهِ لِيُبَيِّنَ لَهُمْ فَيُضِلُّ اللَّهُ مَنْ يَشَاءُ وَيَهْدِي مَنْ يَشَاءُ وَهُوَ الْعَزِيزُ الْحَكِيمُ ﴿٤﴾

Artinya :

Kami tidak mengutus seorang rasulpun, melainkan dengan bahasa kaumnya, supaya ia dapat memberi penjelasan dengan terang kepada mereka. Maka Allah menyesatkan siapa yang Dia kehendaki, dan memberi petunjuk kepada siapa yang Dia kehendaki. Dan Dia-lah Tuhan Yang Maha Kuasa lagi Maha Bijaksana. (QS. Ibrahim: 4)

pengusaan terhadap suatu bahasa merupakan kunci dalam berkomunikasi.

Sebagaimana dijelaskan dalam surah Al-kahfi ayat 93 sebagai berikut:

حَتَّىٰ إِذَا بَلَغَ بَيْنَ السَّدَّيْنِ وَجَدَ مِنْ دُونِهِمَا قَوْمًا لَا يَكَادُونَ يَفْقَهُونَ
قَوْلًا ﴿٩٣﴾

Artinya:

Hingga apabila dia telah sampai di antara dua buah gunung, dia mendapati di hadapan kedua bukit itu suatu kaum yang hampir tidak mengerti pembicaraan.(QS. Al-Kahfi :93)

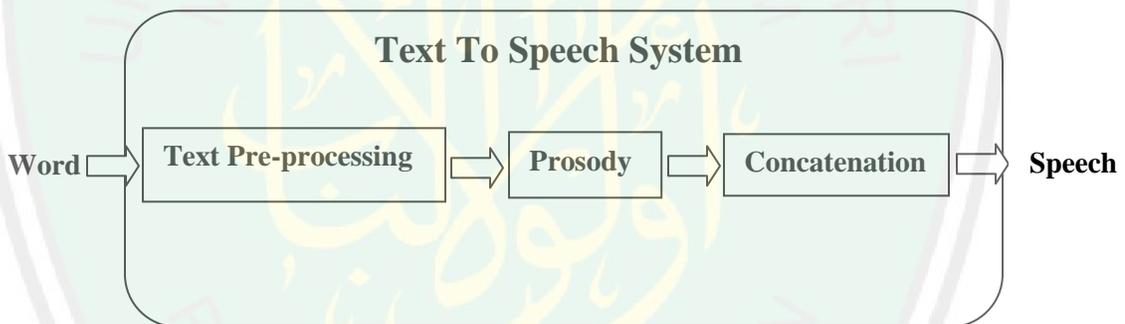
BAB III

ANALISIS DAN PERANCANGAN SISTEM

Dalam bab ini akan dibahas mengenai perancangan sistem *text to speech* menggunakan *diphone concatenation*.

3.1 Text To Speech Synthesis System

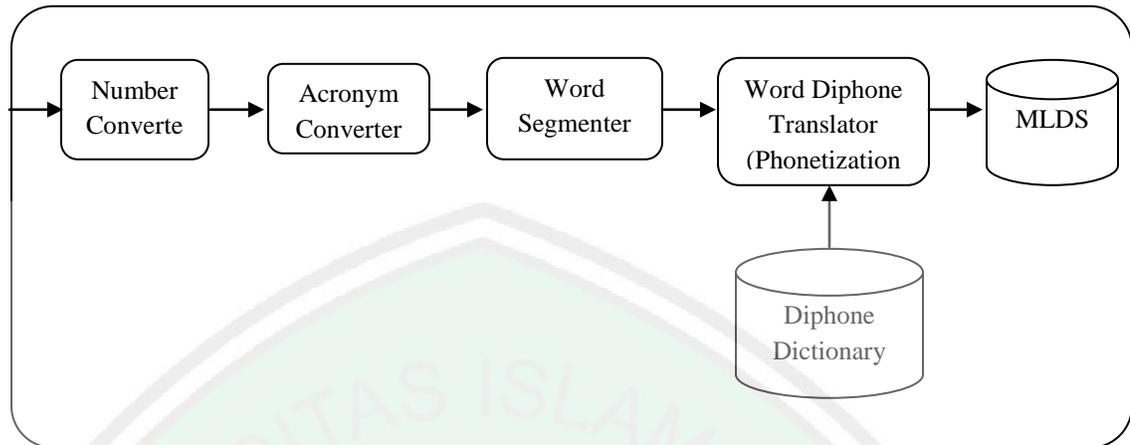
Text To Speech synthesis system terdiri dari 3 bagian, yaitu *text pre-processing*, pembangkitan *prosody* dan *concatenation*. Di bawah ini adalah blok diagram *text to speech* system :



Gambar 3.1 Blok diagram Text to speech synthesis system

3.1.1 Text Pre-processing

Yaitu pengkonversian dari input yang berupa teks menjadi *diphone* (gabungan dua buah fonem). Ketika input yang berupa teks, akronim (singkatan) ataupun angka maka bagian ini akan mengkonversikan menjadi *diphone* yang telah tersedia di database *diphone*. Diagram blok untuk proses *text pre-processing* adalah :



Gambar 3.2 Blok diagram *text pre-processing*

Dari blok diagram diatas dapat dijelaskan cara kerja system yaitu :

a. Number Converter

Jika input pada system berupa angka, maka system mengkonversikan angka ke dalam representasi diphone (gabungan dua buah fonem)

b. Acronym Converter

Jika input pada system berupa kata singkatan dalam bahasa Indonesia, maka system mengkonversikan singkatan ke dalam representasi diphone (gabungan dua buah fonem).

c. Word Segmenter

Jika input pada system berupa kata atau kalimat maka system mengonversikan kata atau kalimat ke dalam representasi diphone (gabungan dua buah fonem).

d. Diphone Dictionary

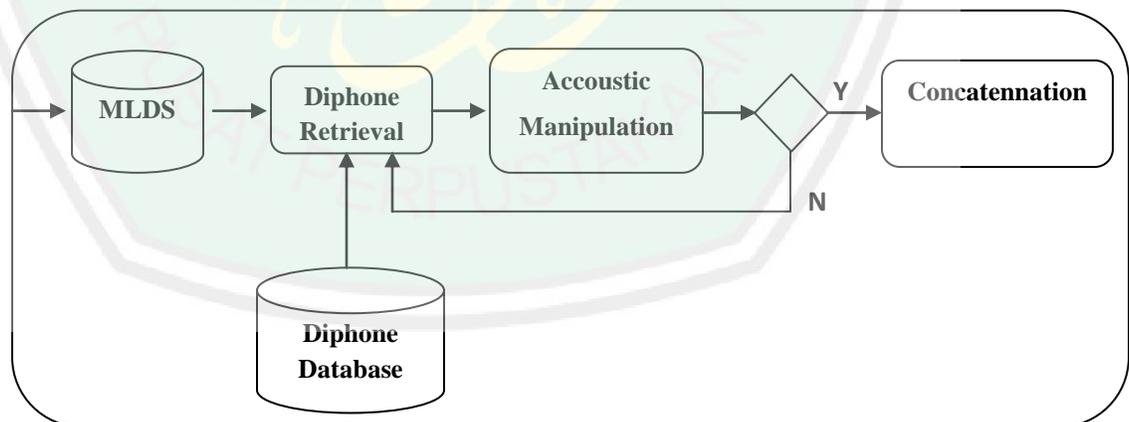
Merupakan database yang berupa kumpulan dari diphone-diphone. Pembuatan diphone dilakukan dengan melakukan pelabelan pada sinyal ucapan. Jumlah diphone yang telah terkumpul sebanyak 1024 diphone.

e. MDLS (Multi Level Data Structure)

Terdiri dari semua data yang diperlukan untuk sub system berikutnya, dalam hal ini adalah proses prosody. MDLS terdiri dari representasi diphone-diphone hasil pengkonversian inputan

3.1.2 Prosody

Yaitu untuk mendapatkan ucapan yang lebih alami, ucapan yang dihasilkan harus memiliki intonasi (*prosody*). Secara kuantisasi, prosodi adalah perubahan nilai *pitch* (frekuensi dasar) selama pengucapan kalimat dilakukan atau *pitch* sebagai fungsi waktu. Prosodi bersifat sangat spesifik untuk setiap bahasa, sehingga model yang diperlukan untuk membangkitkan data-data prosodi menjadi sangat spesifik juga untuk suatu bahasa. Diagram blok untuk prosodi adalah :



Gambar 3.3 Blok diagram *prosody*

- a) MLDS (Multi Level Data Structure), terdiri dari semua data yang diperlukan untuk sub system berikutnya. MLDS terdiri atas kata, representasi *diphone*,

Prosodic parameter untuk tiap diphone (ini perpaduan antara level kata dan level *prosody* kalimat). MLDS mengizinkan untuk modulasi.

- b) Diphone Retrieval didalamnya terdapat tiga tahapan yang terjadi, yaitu database hasil perekaman *diphone*, menyimpan bentuk gelombang *diphone* dan *prosodic* parameter dalam variabel.
- c) Accoustic Manipulation di dalamnya terdapat proses pengenalan file-file gelombang .wav (load, play, write), vast array dari peralatan signal processing, built-in function, ease debugging, GUI capable.

3.1.3 Concatenation

Yaitu penggabungan segmen-segmen bunyi yang telah direkam sebelumnya. Setiap segmen berupa *diphone* (gabungan dua buah fonem). Pada perekaman suara dilakukan beberapa kali agar mendapatkan hasil yang akurat.

3.2 Engine Speech Synthesizer Mbrola

Engine Speech Synthesizer mbrola membutuhkan rangkaian input agar dapat melakukan sintesa terhadap suatu fonem. Sebagai contoh, input “V 50 25 114” memerintahkan *engine* mbrola untuk menyuarakan fonem huruf “a” berdurasi 50 *mili-second* dengan 25% durasi ditetapkan nilai pitch-nya 114 Hz. Nilai parameter presentasi dan *pitch* berkaitan dengan *prosody* dengan tujuan agar sintesa suara yang dihasilkan lebih alami. Bila kedua parameter dihilangkan maka akan diperoleh suara yang datar.

Berikut contoh input *engine* mbrola untuk ucapan bahasa Indonesia “saya sedang bekerja”, dengan database diphone ID1 tanpa dilengkapi *prosody*.

_300

s 126

V 86

j 116

V 120

s 139

@ 44

d 63

V 114

N 126

b 97

@ 42

k 97

@ 42

r 100

dZ 88

V 129

_300

Agar hasil sintesa terdengar lebih alami harus ditambahkan *prosody*, setiap fonem dilengkapi dengan parameter presentasi dan durasi. Contoh masukan yang telah ditambahkan parameter presentasi dan *pitch* sebagai berikut :

_300

s 126 94 116

V 86 0 119 23 121 46 122 69 122 92 123

j 116 0 123 17 124 34 125 51 126 68 130 85 131

V 120 0 132 16 133 33 133 49 134 66 135 82 134

s 139 0 133 85 470

@ 44 0 124 45 128 90 128

d 63 0 124 31 120 62 117 94 120

V 114 0 120 17 120 35 118 52 116 70 114 87 112

N 126 0 113 15 115 31 116 47 116 63 115 78 114

b 97 82 131

@ 42 0 131 47 125 94 122

k 97 82 131

@ 42 0 131 47 125 94 122

r 100 0 122 33 121 67 117

dZ 88 0 110 90 111

V 129 0 114 15 113 30 108 46 104 61 101 77 99

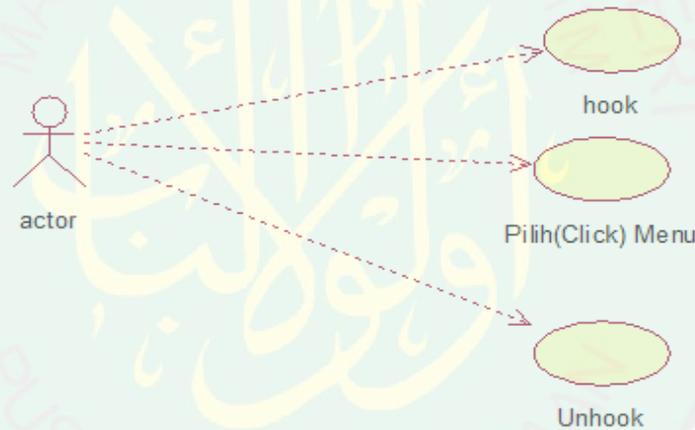
_300

Hasil dari sintesa terhadap input di atas dapat dibedakan apakah mesin sedang melakukan pertanyaan atau pernyataan.

3.3 Analisis Usecase Diagram

Pada sub bab ini akan dilakukan pemodelan sistem dengan menggunakan use case diagram. Pemodelan dengan use case ini dimaksudkan untuk menggambarkan proses aktivitas yang terjadi pada system. Berikut adalah gambaran secara global use case diagram dari system yang akan dibuat:

- a) Actor : segala sesuatu yang berinteraksi dengan sistem aplikasi komputer. Jadi, actor bis berupa orang, perangkat keras, atau mungkin juga obyek lain dalam system yang sama. Biasanya yang dilakukan oleh actor adalah memberikan informasi pada system dan atau memerintahkan system untuk melakukan sesuatu. Dalam sistem ini actor berupa pengguna/*user* aplikasi.
- b) Use case : menjelaskan urutan kegiatan yang dilakukan actor dan sistem untuk mencapai suatu tujuan tertentu. Walaupun menjelaskan kegiatan namun use case hanya menjelaskan apa yang dilakukan oleh actor dan sistem, bukan menjelaskan bagaimana actor dan sistem melakukan kegiatan tersebut.



Gambar 3.4 Use case diagram

Dari gambar use case diagram diatas, maka deskripsi dari masing-masing use case dapat dijelaskan sebagai berikut :

- a. *Use case* : Hook

Actor : *User* (pengguna)

Deskripsi : *use case* ini digunakan untuk mengaktifkan fungsi pembaca layar.

- b. *Use case* : Pilih menu

Actor : *User* (pengguna)

Deskripsi : *use case* menunjukkan kegiatan user ketika memilih menu pada layar computer.

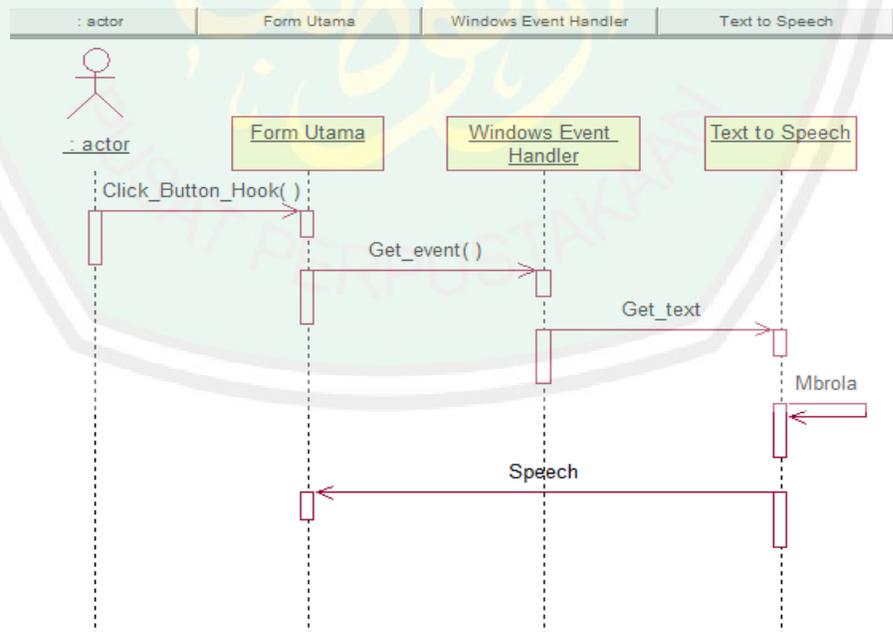
c. *Use case* : Unhook

Actor : *User* (pengguna)

Deskripsi : *use case* ini digunakan untuk menonaktifkan pembaca layar.

3.4 Analisis Sequence Diagram

Sequence diagram merupakan diagram interaksi yang disusun berdasarkan urutan waktu. *Sequence diagram* ini digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh obyek-obyek ini di dalam *use case*. Pada sistem *text-to-speech* menggunakan metode *diphone concatenation*, terdapat beberapa diagram, antara lain :

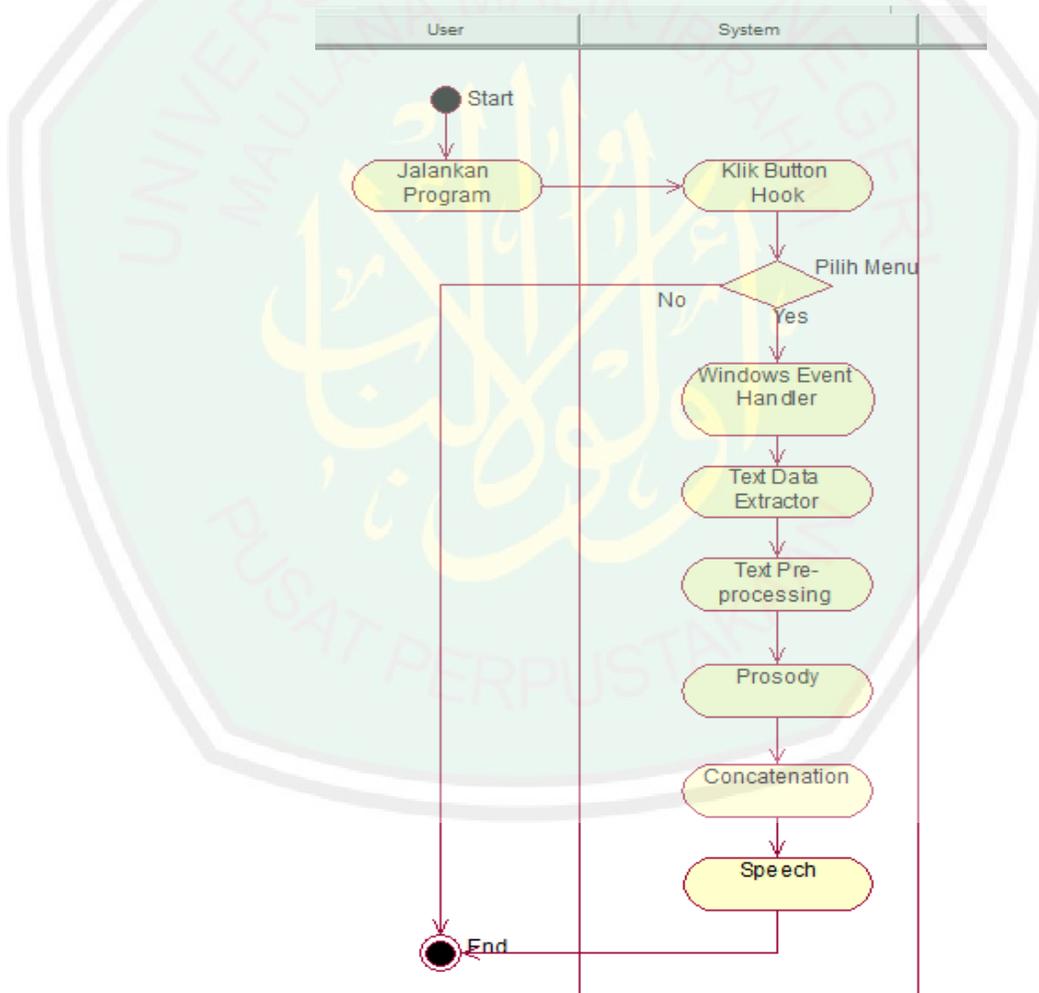


Gambar 3.5 *Sequence Diagram* Screen Reader

3.5 Analisis Activity Diagram

Activity diagram merupakan bentuk khusus dari state machine yang bertujuan memodelkan komputasi-komputasi dan aliran-aliran kerja yang terjadi dalam sistem/perangkat lunak yang sedang dikembangkan.

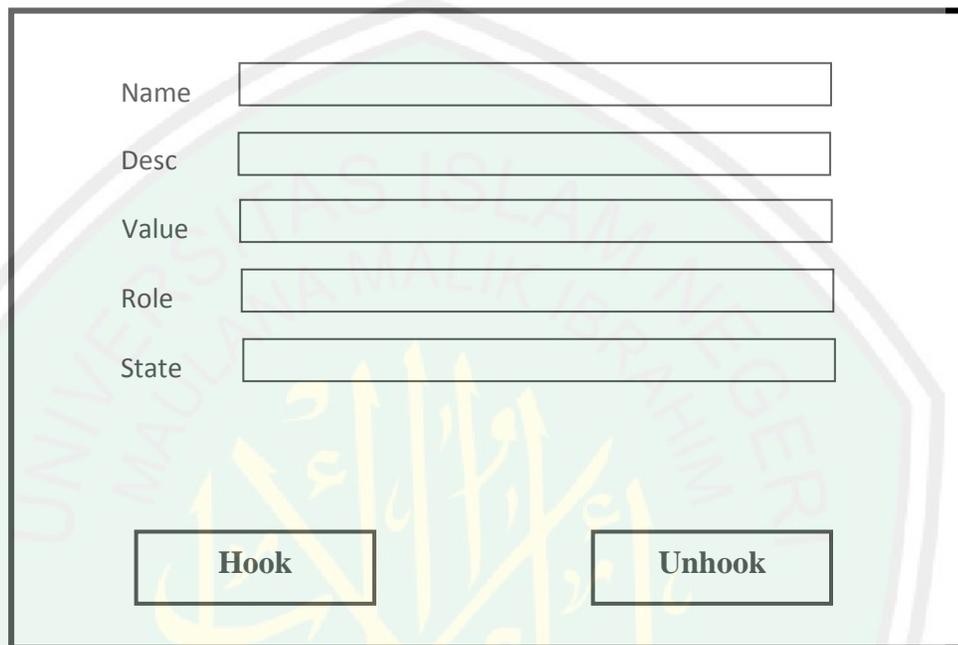
State pada activity diagram merepresentasikan state dari komputasi yang dieksekusi, bukan state dari suatu obyek biasa. Berikut activity diagram screen reader :



Gambar 3.6 Activity Diagram Screen Reader

3.6 Rancangan Antar Muka

Desain Interface merupakan rancangan antarmuka (interface) program yang akan diimplementasikan. Rancangan yang dibuat antara lain :



The image shows a form for menu design. It contains five text input fields labeled 'Name', 'Desc', 'Value', 'Role', and 'State' stacked vertically. Below these fields are two buttons: 'Hook' on the left and 'Unhook' on the right. The entire form is enclosed in a black rectangular border. A large, faint watermark of the Maulana Malik Ibrahim State Islamic University of Malang logo is visible in the background.

Gambar 3.7 Desain Menu Utama

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai hasil uji coba terhadap aplikasi *screen reader* yang telah dibuat. Uji coba ini bertujuan untuk mengetahui apakah aplikasi dapat berjalan sebagaimana mestinya sesuai dengan perancangan yang dibuat pada bab 3.

4.1 Implementasi

Implementasi sistem merupakan realisasi sistem berdasarkan pada desain yang telah ditentukan. Implementasi sistem juga merupakan proses pembuatan dan penerapan sistem secara utuh baik dari sisi perangkat keras maupun perangkat lunak.

4.2 Ruang Lingkup Perangkat keras

Perangkat keras yang dibutuhkan dalam pengembangan aplikasi *screen reader* adalah sebagai berikut :

1. Laptop Toshiba Intel® Pentium ® CPU P600 @ 1.87 GHz
2. Memory 1 GB DDR2
3. Hardisk 320 GB, Serial ATA
4. Intel® Graphics 256 MB

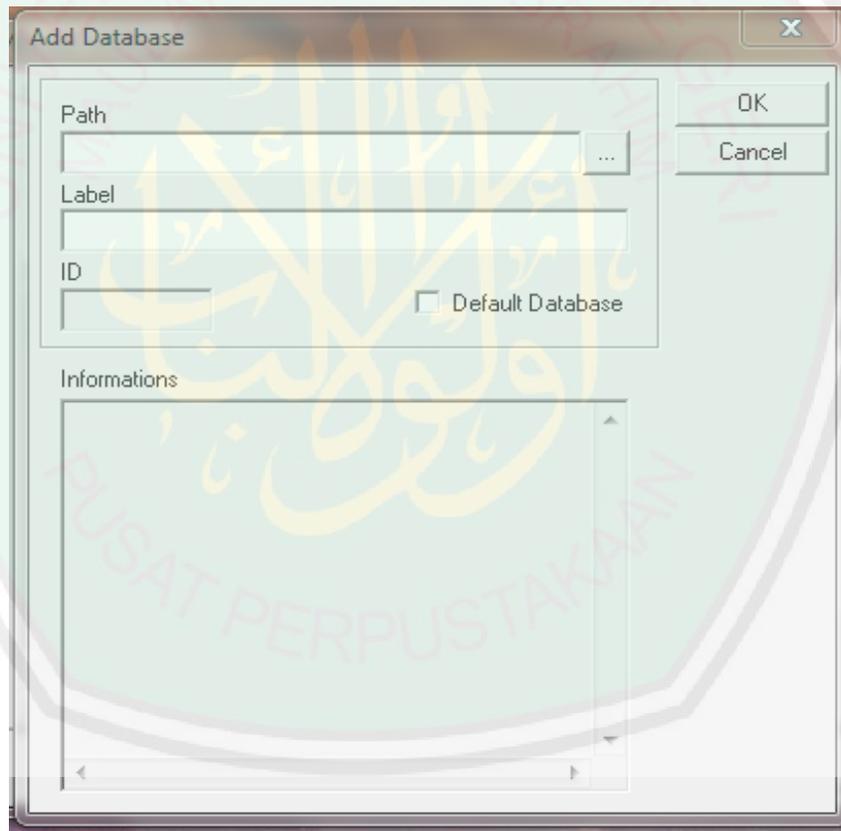
4.3 Ruang Lingkup Perangkat Lunak

Perangkat lunak yang dibutuhkan agar mampu menjalankan aplikasi *screen reader* dengan baik adalah sebagai berikut :

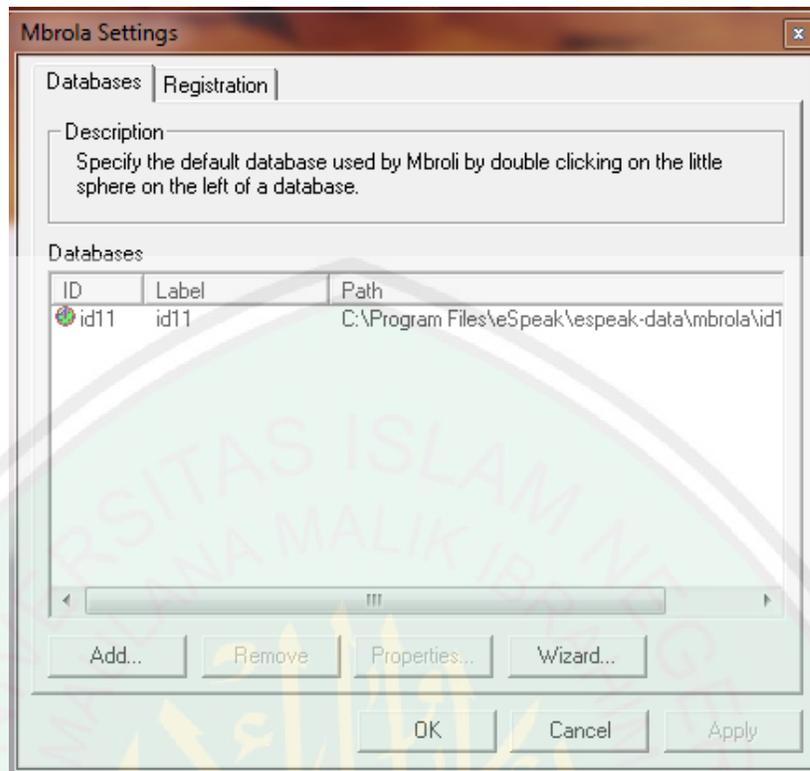
1. Microsoft Windows 7
2. Microsoft Visual Basic 6
3. MBROLA Tools 3.5
4. Database Diphone Bahasa Indonesia

4.4 Sistem kerja Software

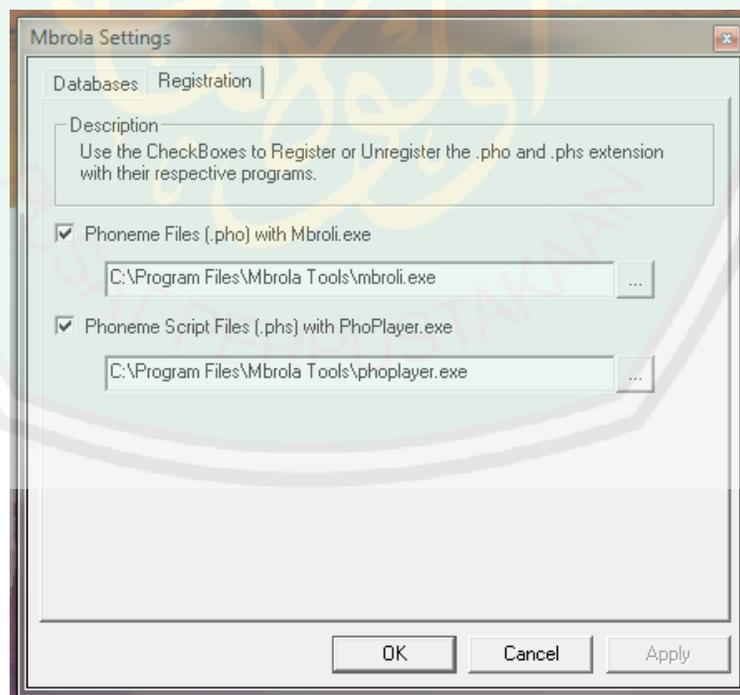
Sebelum menjalankan program terlebih dahulu dilakukan seting database Mbrola bahasa Indonesia (id1). Dimulai dengan menambahkan database id1 ke dalam program Mbrola pada *window add database*.



Gambar 4.1 form *Add Database*



Gambar 4.2 Window Mbrola settings



Gambar 4.3 Registrasi .pho dan .phs

Setelah database id1 dimasukkan, selanjutnya pada gambar 4.3 dilakukan registrasi untuk menentukan *path* program yang akan dipakai untuk membaca phoneme file dengan ekstensi .pho yang akan dikonversi lagi menjadi ucapan.

Pertama kali adalah mengaktifkan fungsi pembaca layar pada screen reader.

Kode program untuk mengaktifkan pembaca layar adalah sebagai berikut :

```
LHook = SetWinEventHook(SYS_ALERT, OB_ACCELERATORCHANGE, 0&, AddressOf
WinEventFunc, 0, 0, WINEVENT_SKIPOWNPROCESS)
```

Selanjutnya dilakukan pengambilan *event properties* dari posisi kursor *mouse* pada layar komputer. Kode program untuk mengambil nilai event adalah sebagai berikut :

```
Select Case LEvent
Case SYS_FOREGROUND, SYS_ALERT, OB_FOCUS, OB_SELECTION,
OB_VALUECHANGE
LRet = AccessibleObjectFromEvent(HWnd, idObject, idChild, ObA, V)
If LRet = 0 Then
With F1
sName = ObA.accName(V)
.Text1(0).Text = sName
.Text1(1).Text = ObA.accDescription(V)
s1 = ObA.accValue(V)
.Text1(2).Text = s1
.Text1(3).Text = ObA.accRole(V)
```

```

.Text1(4).Text = ObA.accState(V)

End With

If (ObA.accState(V) And STATE_UNAVAILABLE) Then s = " disabled"

If LEvent = OB_VALUECHANGE Then

    s = s1 & s

Else

    s = sName & s

End If

SpeakIt s

End If

End Select

WinEventFunc = 0

End Function

```

Setelah Name event didapatkan. Selanjutnya dilakukan pemanggilan fungsi

Mod SpeakIt :

```

Public Sub SpeakIt(sText As String)

    On Error Resume Next

    If (Len(sText) = 0) Then Exit Sub

    Voice.Speak sText, SPEAK_FLAGS_1

End Sub

```

Mod SpeakIt berfungsi untuk mengucapkan teks masukan dari Nama event, yang mana teks masukan tersebut di konversi menjadi ucapan oleh Mbrola.

```

Public Sub UnHook()

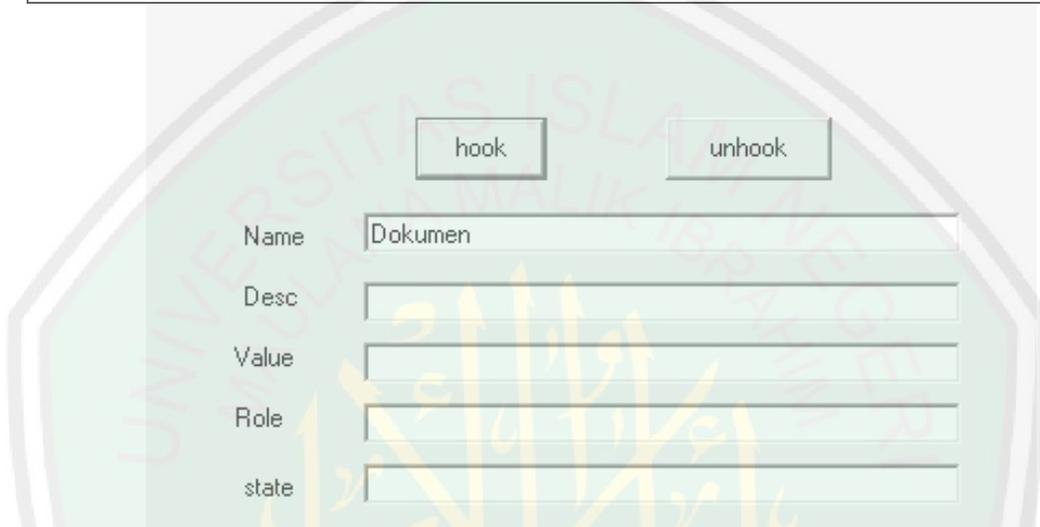
Dim LRet As Long

If LHook = 0 Then Exit Sub

LRet = UnhookWinEvent(LHook)

End Sub

```



Gambar 4.4 Tampilan screen reader ketika membaca pergerakan mouse pada posisi menu “Dokumen”

Gambar 4.4 menunjukkan bahwa teks masukan yang diperoleh dari pergerakan mouse berupa teks “Dokumen”, teks inilah yang nantinya akan di konversikan oleh *engine* Mbrola menjadi suara

Penggalan kode program diatas berfungsi untuk menonaktifkan fungsi pembaca layar ketika tombol *unhook* ditekan.

4.5 Hasil Uji coba

Pada sub bab ini akan dilakukan pengujian kecakapan (*versatility*) pengucapan kata pada aplikasi *screen reader*, salah satu kriteria yang perlu diperhatikan dalam TTS (*Text To Speech*) adalah kecakapan (*versatility*), yakni

berkenaan dengan seberapa luas kata yang dapat diucapkan. Dalam bahasa Indonesia kita mengenal vocal, konsonan, dan juga alofon, dimana kombinasi posisi dari masing-masing komponen tersebut akan sangat mungkin membedakan pengucapannya.

Tabel 4.1 Data Masukan yang diproses oleh Mbrola

KATA	DIPHONE- DIPHONE	KODE SAMPA
alquran	a_ al qu ur ra an _n	_ VL q U R V N
Dokumen	d_ do ok ku um me en _n	_ d Q K U M @ N
kualitas	k_ ku ua al li it ta as _s	_ K U V L I t V s
gambar	g_ ga am mb ba ar _r	_ G V M b V R
buat	b_ bu ua at _t	b U V T
Program	p_ pr ro og gr ra am _m	p R Q G R V M
strata	s_ st tr ra at ta _a	_ S t R V t V R
struktur	s_ st tr ru uk kt tu ur _r	_ S t R U K t U R
transportasi	t_ tr ra an ns sp po or rt ta as si _i	t R V N S P Q R t V S I

Tabel 4.2 Data hasil pengujian kecakapan pengucapan

NO	POLA	CONTOH SUKU KATA /KATA	KEMAMPUAN PENGUCAPAN
1	VK	<u>A</u> l-quran	Baik
2	KV	<u>D</u> o-kumen	Baik
3	KVV	<u>Kua</u> -litas	Baik
4	KVK	<u>Gam</u> -bar	Baik
5	KVVK	<u>Buat</u>	Baik
6	KKV	<u>Pro</u> -gram	Baik
7	KKKV	<u>Stra</u> -ta	Baik
8	KKKVK	<u>Struk</u> -tur	Baik
9	KKVKK	<u>Trans</u> -portasi	Baik

Berdasarkan pengamatan terhadap beberapa kata dalam bahasa Indonesia oleh sistem TTS yang ada pada *screen reader* ini, maka dapat dikatakan bahwa sistem TTS yang ada pada *screen reader* ini mampu mengucapkan pengucapan vokal /a/, /i/, /u/, /e/, /o/ dengan benar, baik vokal pada posisi awal, tengah maupun akhir kata, dan juga masih dapat mengucapkan dengan baik untuk vokal serupa yang beriringan.

Demikian pula dengan pengucapan *grafem* konsonan, sebagian besar dapat diucapkan dengan baik. Dalam hal ini system diuji dengan pola variasi posisi vokal-konsonan dalam Bahasa Indonesia seperti pada table 4.2. Pengujian dilakukan dengan sejumlah kata yang berpola seperti diatas. Berdasarkan data pengujian, maka sistem TTS hasil realisasi dapat mengucapkan seluruh kata baku dalam Bahasa Indonesia.



BAB V

PENUTUP

5.1 Kesimpulan

Kesimpulan yang diperoleh setelah mengerjakan tugas akhir tentang implimentasi *text to speech* pada *screen reader* berbasis *Mbrola* sebagai berikut :

1. Sistem *text to speech* dapat di terapkan dalam aplikasi *screen reader*.
2. Teknik *Diphone Concatenation* pada *MBROLA* dapat digunakan pada aplikasi *screen reader* guna menghasilkan suara dalam bahasa Indonesia yang memiliki interpretasi yang baik.
3. Berdasarkan pengamatan terhadap beberapa kata dalam bahasa Indonesia oleh sistem TTS yang ada pada *screen reader* ini, maka dapat dikatakan bahwa sistem TTS yang ada pada *screen reader* ini mampu mengucapkan pengucapan vokal /a/, /i/, /u/, /e/, /o/ dengan benar, baik vokal pada posisi awal, tengah maupun akhir kata, dan juga masih dapat mengucapkan dengan baik untuk vokal serupa yang beriringan. Demikian pula dengan pengucapan *grafem* konsonan, sebagian besar dapat diucapkan dengan baik. Dalam hal ini system diuji dengan pola variasi posisi vokal-konsonan dalam Bahasa Indonesia seperti pada table pengujian dilakukan dengan sejumlah kata yang berpola seperti VK, KV, KVV, KVK, KVVK, KKV, KKKV, KKKVK, KKVKK. Dari hasil yang didapatkan Aplikasi *screen reader* memiliki kemampuan pengucapan yang baik.

5.2 Saran

Untuk pengembangan lebih lanjut, penulis memberikan saran sebagai berikut:

1. Program ini dapat dikembangkan dengan menambahkan database bahasa selain bahasa indonesia
2. Menambahkan fungsi deteksi bahasa sehingga apabila ditemukan bahasa selain bahasa indonesia maka sistem akan otomatis mengganti database bahasa sesuai bahasa yang di deteksi.



DAFTAR PUSTAKA

Al-Qur'an Al-Karim

Al-Qurthubi, Syaikh Imam. 2008. *Al jami' li Ahkam Al-Quran*. Jakarta: Pustaka Azzam.

Ath-Thabari, Abu ja'far Muhammad bin Jarir. 2008. *Jami' Al-bayan An Ta'wil Ayi Al-Qur'an*. Jakarta: Pustaka Azzam.

Arman, Arry Akhmad. 2002. *Konversi dari Teks ke Fonem*. ITB : Bandung

Tritoasmoro, Iwan Iwut. 2006. *Text-To-Speech Bahasa Indonesia Menggunakan Concatenation Synthesizer Berbasis Fonem*. Institut Teknologi Sepuluh November (ITS). Surabaya.

Rachma, Handi Dwi, Zonda Rumiaga dan Miftahul Huda. 2011. *Pembuatan Text-To-Speech Synthesis System Untuk Penutur Berbahasa Indonesia*. Politeknik Elektronika Negeri Institut Teknologi Sepuluh November (ITS). Surabaya.

Rommel, Edwin. *Aplikasi SMS dengan Text to Speech Bahasa Indonesia Pada Sistem Operasi Symbian*. Departemen Teknik Elektro Institut Teknologi Bandung (ITB). Bandung.

Stephanus, Priyowidodo. *Speech Synthesizer Bahasa Indonesia Berbasis Diphone MBROLA*. Universitas Sumatra Utara. Medan.

indotts.melsa.net.id/perkembangan_TTS.pdf. Diakses pada 30 November 2011

<http://membacasebagaiketerampilanbahasa.blogspot.com/2012/11/pengertian-membaca.html>. Diakses pada tanggal 21 Desember 2012

<http://adynesia2.blogspot.com/2011/10/makalah-peran-bahasa-indonesia-dalam.html>. Diakses pada tanggal 21 Desember 2012

http://carapedia.com/pengertian_definisi_bahasa_menurut_para_ahli_info494.html. Diakses pada tanggal 21 Desember 2012

<http://wawan-junaidi.blogspot.com/2010/09/sejarah-bahasa-indonesia.html>. Diakses pada tanggal 21 Desember 2012

<http://teknologibahasa.wordpress.com/>
Diakses pada tanggal 21 Desember 2012