

**PERANCANGAN APLIKASI SPEECH RECOGNITION
SEBAGAI PERINTAH, MENGGUNAKAN MICROSOFT SAPI
UNTUK MENJALANKAN APLIKASI BERBASIS WINDOWS
DENGAN MENGGUNAKAN VISUAL BASIC 6.0**

SKRIPSI



**Disusun Oleh:
Ahmad Muzakky Syarif
09.18.190**

**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNIK INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013**

LEMBAR PERSETUJUAN DAN PENGESAHAN
PERANCANGAN APLIKASI SPEECH RECOGNITION
SEBAGAI PERINTAH, MENGGUNAKAN MICROSOFT SAPI
UNTUK MENJALANKAN APLIKASI BERBASIS WINDOWS
DENGAN MENGGUNAKAN VISUAL BASIC 6.0

SKRIPSI

*Disusun dan Diajukan untuk melengkapi dan memenuhi persyaratan guna
mencapai Gelar Sarjana Teknik Informatika Strata Satu (S-1)*


Disusun Oleh :
Ahmad Muzakky Syarif
09.18.190

Diperiksa dan Disetujui,

Dosen Pembimbing I

Dosen Pembimbing II


Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002


Febriana Santi Wahyuni, S.Kom, M.Kom
NIP.P. 1031000425

Ketua Jurusan Teknik Informatika S-1


Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002

PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013



**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL
MALANG**

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini :

Nama : Ahmad Muzakky Syarif
Nim : 09.18.190
Program Studi : Teknik Informatika S-1
Fakultas : Teknologi Industri

Menyatakan dengan sesungguhnya bahwa Skripsi saya yang berjudul :
“PERANCANGAN APLIKASI SPEECH RECOGNITION SEBAGAI PERINTAH, MENGGUNAKAN MICROSOFT SAPI UNTUK MENJALANKAN APLIKASI BERBASIS WINDOWS DENGAN MENGGUNAKAN VISUAL BASIC 6.0” Adalah Skripsi saya sendiri bukan duplikat serta mengutip atau menyadur seluruhnya karya orang lain kecuali dari sumber aslinya.

Malang, 13 Agustus 2013

Yang membuat pernyataan

METERAI
TEMAPEL
Tgl
EEA0C BFM 19062
KEMENTERIAN

6000 DJP

Ahmad Muzakky Syarif

KATA PENGANTAR

Dengan menyatakan syukur kepada Allah Subhanahu Wata'ala yang telah melimpahkan rahmat dan hidayah serta inayahNya kepada penulis sehingga dapat menyelesaikan skripsi ini. Shalawat dan Salam mudah-mudahan kita selaku umatnya bisa selalu berusaha mencontoh tauladan nabi Muhammad SAW.

Tugas akhir ini dapat penulis selesaikan atas bantuan dari berbagai pihak yang selalu membantu baik moril maupun materi. Untuk itu penulis menyampaikan banyak terima kasih kepada :

1. Kedua Orangtua dan keluarga yang senantiasa tak luput mengucapkan doa dan selalu mengulur dukungan sehingga penulis lebih kuat dan tenang dalam menyelesaikan skripsi ini.
2. Ir. Soeparno Djiwo, MT, selaku Rektor ITN Malang.
3. Ir. Anang Subardi, MT, selaku Dekan Fakultas Teknologi Industri ITN Malang.
4. Joseph Dedy Irawan, ST, MT, selaku Ketua Jurusan Teknik Informatika ITN Malang dan dosen pembimbing I yang selalu memberikan dorongan dan motivasi kepada penulis.
5. Febriana Santi Wahyuni, S.Kom, M.Kom, selaku dosen pembimbing II yang telah memberikan arahan serta bimbingan dalam penulisan skripsi ini.
6. Segenap Dosen Program Studi Teknik Informatika Institut Teknologi Nasional Malang, yang telah membimbing dan mencurahkan ilmunya kepada penulis.
7. Orang-orang terdekat yang selalu menyertaiku dengan tulus, semoga keindahan menyertai kalian semua.
8. Sahabat-sahabatku segenap angkatan 2009 ITN Malang, khususnya Jurusan Teknik Informatika yang selalu bersama dalam suka maupun duka.
9. Semua pihak yang telah mendukung dan membantu terselesaikannya skripsi ini.

Penyusun menyadari bahwa skripsi ini masih jauh dari sempurna, oleh karena itu dengan segala kerendahan hati, penulis mengharapkan kritik dan saran yang membangun untuk perbaikan kedepannya. Semoga karya kecil ini bermanfaat bagi peningkatan serta perkembangan ilmu pengetahuan dan teknologi mahasiswa Institut Teknologi Nasional Malang khususnya dan mahasiswa di Indonesia umumnya.

Malang, 13 Agustus 2013

Penyusun

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN DAN PENGESAHAN SKRIPSI	ii
SURAT PERNYATAAN ORISINALITAS	iii
ABSTRAK	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	viii
DAFTAR TABEL	x
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Penulisan	3
BAB II LANDASAN TEORI	
2.1 Konsep Digitalisasi Audio.....	4
2.1.1 <i>Suara</i>	4
2.1.2 <i>Analog to Digital</i>	5
2.2 Speech Recognition.....	7
2.2.1 <i>Mode Sistem Pengenalan Pembicaraan</i>	8
2.2.2 <i>Pemisahan Kata</i>	8
2.2.3 <i>Sifat Pengenalan Pembicaraan</i>	10
2.2.4 <i>Pencocokan Kata</i>	10
2.2.5 <i>Perbendaharaan Kata</i>	10
2.3 Microsoft Speech Application Programming Interface	11
2.3.1 <i>Application Programming Interface (API)</i>	12
2.3.2 <i>Device Driver Interface(DDI)</i>	15
2.4 Microsoft Developer Network (MSDN)	17
2.5 Microsoft Visual Basic	17
2.5.1 <i>Interface Visual Basic 6.0</i>	18
2.5.2 <i>Konsep Dasar Pemograman Dalam Visual Basic 6.0</i>	20
2.5.3 <i>Membuat Project Baru</i>	21

BAB III PERANCANGAN	
3.1	Analisa 25
3.1.1	<i>Analisa Speech Application Programming Interface</i> 25
3.1.2	<i>Analisa Kebutuhan Perangkat</i> 28
3.2	Perancangan Blok Diagram 29
3.3	Perancangan Diagram Alir (Flow Chart) 29
3.4	Perancangan Tampilan Aplikasi 30
BAB IV IMPLEMENTASI DAN PENGUJIAN	
4.1	Implementasi Perangkat Lunak 33
4.1.1	<i>Implementasi Aplikasi Perintah Suara</i> 35
4.2	Pengujian 38
4.2.1	<i>Pengujian Fungsional</i> 39
4.2.2	<i>Pengujian User</i> 40
BAB V PENUTUP	
5.1	Kesimpulan 42
5.2	Saran 42
DAFTAR PUSTAKA	
LAMPIRAN	

DAFTAR GAMBAR

2.1.	Karakteristik Gelombang Suara	4
2.2.	Pengkodean PCM	5
2.3.	Ilustrasi Modulasi Delta	6
2.4.	Proses Speech Recognition	8
2.5.	Arsitektur Microsoft SAPI	12
2.6.	Proses Yang terjadi Pada <i>Event</i> API	14
2.7.	Diagram Alir DDI	16
2.8.	Interface Visual Basic 6.0.....	19
2.9.	Komponen Standart Dalam Toolbox	20
2.10.	Layar Pemilihan Jenis Project	21
2.11.	Jendela Form	23
2.12.	Jendela Kode	24
3.1.	Diagram Alir Speech Application Programming Interface	27
3.2.	Blok Diagram Sistem	29
3.3.	Flow Chart Aplikasi Perintah Suara	30
3.4.	Rancangan Tampilan Awal	30
3.5.	Rancangan Menambah Perintah	31
3.6.	Rancangan Tampilan Menu Help	32
4.1.	Struktur Menu.....	33
4.2.	Components Pada Visual Basic 6.0 (1)	34
4.3.	Components Pada Visual Basic 6.0 (2)	34
4.4.	Implementasi Tampilan Menu Utama	35
4.5.	Implementasi Tampilan Help_Manual	35
4.6.	Implementasi Tampilan Menu Manual	36
4.7.	Implementasi Tampilan Help_Microsoft SAPI	36
4.8.	Implementasi Tampilan Microsoft SAPI	37
4.9.	Implementasi Tampilan Menu About Me	37
4.10.	Implementasi Tampilan Abot Me	37
4.11.	Implementasi Tampilan Add (Tambah Perintah)	38

DAFTAR TABEL

Tabel 2.1. Metode API	14
Tabel 2.2. Metode DDI	17
Tabel 3.1. Method Speech Application Programming Interface	26
Tabel 4.1. Data Spesifikasi Perangkat	38
Tabel 4.2. Data Pengujian Fungsional	39
Tabel 4.3. Hasil Pengujian User	40

BAB I

PENDAHULUAN

1.1 Latar Belakang

Komunikasi merupakan kebutuhan manusia yang sangat penting. Untuk mendapatkan informasi, manusia membutuhkan proses komunikasi sebagai salahsatu media interaksi antara satu dengan yang lainnya, dan kebanyakan komunikasi dilakukan dengan suara (bicara) dibandingkan dengan tulisan atau gambar. Karena komunikasi dengan suara adalah hal yang lebih alami, selain itu, dengan berbicara komunikasi lebih efisien dan efektif, manusia bisa berbicara sepuluh kali lebih cepat daripada menulis atau mengetik.

Salahsatu interaksi dalam penggunaan komputer adalah "*Shortcut*" yang merupakan jalan pintas untuk menjalankan sebuah aplikasi atau program dibandingkan dengan menjalankan dari direktori tempat program disimpan yang biasanya terletak dalam sistem, contoh untuk menjalan "*Microsoft Office Word*" kita harus membuka direktori (C:\Program Files\Office12\WINWORD.exe), cara ini membutuhkan waktu yang lebih lama dan membingungkan. Oleh karena itu dibuatkalah "*Shortcut*" untuk mempersingkat dalam menjalankannya.

Dengan *suara* yang diimplementasikan sebagai *perintah (voice command)* pengguna bisa lebih cepat dalam menjalankan suatu program dibandingkan dengan *Shortcut*. Sebagai contoh ketika kita sedang mengetik dokumen dan ingin membuka aplikasi lain yang bertipe (*.exe) seperti *Windows Media Player (C:\Program Files\Windows Media Player.exe)*. Kita hanya menggunakan perintah suara untuk menjalankannya dengan ucapan "*Player*" *perintah suara inilah yang akan menggantikan path yang ada pada Program Files yang bertipe *.exe.*

Dalam memudahkan dan melayani User dalam menggunakan komputer berbasis Windows, banyak penelitian yang telah dan sedang dilakukan untuk mendapatkan pengenalan suara, salahsatu yang terkenal adalah penelitian yang dilakukan *Microsoft Corporation* yang dikembangkan untuk sistem operasi Windows, Selain mengembangkan mesin pengenalan suara, Microsoft juga mengembangkan standart untuk pengenalan suara yaitu *Speech Application*

Programming Interface (SAPI). (Abdusy Syarif, paper SNATI 2011).

Oleh karena itu dengan adanya perancang ini diharapkan lebih mempermudah User dalam menggunakan komputer khususnya dalam menjalankan program yang bertipe (*.exe.)

1.2 Rumusan Masalah

Permasalahan yang dibahas dalam penulisan Skripsi ini adalah :

- a Bagaimana langkah-langkah pembuatan *Aplikasi Speech Recognition Sebagai Perintah, Menggunakan Microsoft SAPI Untuk Menjalankan Aplikasi Berbasis Windows dengan menggunakan Visual Basic 6.0.*
- b Bagaimana cara sistem mengenali suara yang masuk melalui microphone dan merubahnya menjadi sebuah perintah untuk menjalankan sebuah program atau aplikasi bertipe *.exe.

1.3 Tujuan

Adapun tujuan penulisan ini adalah merancang dan membuat *Aplikasi Speech Recognition Sebagai Perintah, Menggunakan Microsoft SAPI Untuk Menjalankan Aplikasi Berbasis Windows dengan menggunakan Visual Basic 6.0.*

1.4 Batasan Masalah

Batasan masalah pembahasan yang di ambil agar sesuai dengan tujuan dan tidak terjadi penyimpangan tujuan utama, maka ditentukan ruang lingkup pembahasan sebagai berikut :

- a Pembuatan Aplikasi ini menggunakan bahasa pemograman *Visual Basic 6.0*
- b File yang dieksekusi pada Aplikasi ini hanya bertipe *.Exe.
- c Aplikasi yang dirancanag berbasis Windows

1.5 Metodologi

Metodologi yang digunakan dalam penulisan skripsi ini menggunakan beberapa metode antara lain :

- a Pengumpulan Data

Mengumpulkan informasi dengan cara study pustaka dan study literatur yang berkaitan dengan perencanaan dan pembuatan aplikasi.

- b **Analisa dan Perancangan**
Sebelum pembuatan aplikasi, dilakukan analisa kebutuhan perangkat lunak dan perangkat keras kemudian perancangan aplikasi sebagai sandaran pembuatan aplikasi.
- c **Implementasi dan Pengujian**
Membuat aplikasi sesuai perancangan yang dilanjutkan dengan pengujian aplikasi.

1.6 Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan penulisan skripsi ini, maka sistematika penulisan disusun sebagai berikut :

Bab I : Pendahuluan

Berisi Latar Belakang, Rumusan Masalah, Tujuan Penulisan, Batasan Permasalahan, Metode Penulisan dan Sistematika Penulisan.

Bab II : Landasan Teori

Berisi teori-teori yang menunjang dalam proses penulisan skripsi ini.

Bab III : Analisa dan Perancangan

Berisi tentang analisa dan perancangan sistem yang menggambarkan mekanisme dari sistem yang akan dibuat.

Bab IV : Implementasi dan Pengujian

Berisi tentang implementasi dan pengujian struktur serta tampilan aplikasi.

Bab V : Penutup

Merupakan bab terakhir yang berisikan kesimpulan dan saran yang dapat digunakan untuk pengembangan lebih lanjut tentang topik terkait.

BAB II

LANDASAN TEORI

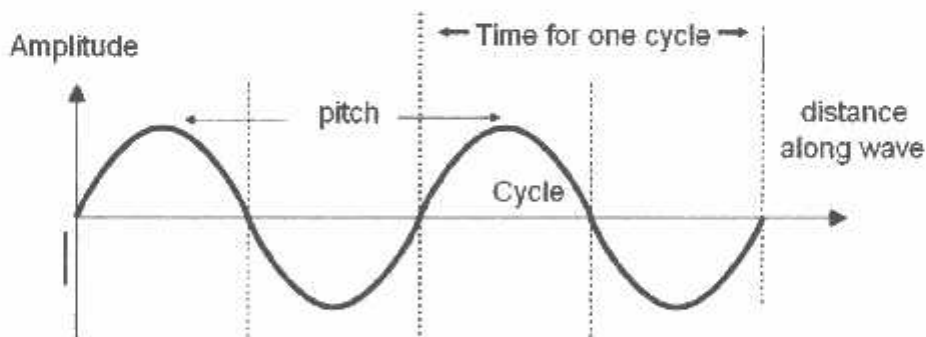
2.1 Konsep Digitalisasi Audio

2.1.1 Suara

Suara atau bunyi biasanya merambat melalui udara dan akan memantul apabila ada dinding, suara atau bunyi tidak dapat merambat melalui ruang hampa. Suara dihasilkan oleh getaran suatu benda. Selama bergetar, perbedaan tekanan terjadi di udara sekitarnya. Pola *osilasi* yang terjadi disebut sebagai gelombang, lihat pada *Gambar 2.1*, Gelombang mempunyai pola sama dan berulang pada interval tertentu, yang disebut periode.

Suara dapat dibagi menjadi dua yaitu :

- *Frekuensi* adalah merupakan standar pengukuran suara jumlah satu siklus per-detik (Jumlah pitch atau amplitudo terbesar yang terdengar oleh telinga).
- *Amplitudo* merupakan intensitas atau keras lemahnya suara, Semakin keras suara semakin besar amplitudonya. Dimana semua bunyi mempunyai jangka waktu dan irama. (*ittelkom, 2012*).



Gambar 2.1 Karakteristik Gelombang Suara

2.1.2 Analog to Digital

Di sekitar kita, dalam kehidupan sehari-hari, sebenarnya lebih banyak sinyal yang direpresentasikan dalam bentuk analog dari pada sinyal dalam bentuk digital. Misalnya, suara, cahaya, suhu, bau dan sebagainya. Namun sinyal-sinyal analog semacam itu akan lebih mudah disimpan, diolah,

direproduksi kembali apabila disimpan dalam bentuk data digital. Sebagai contoh, *Compact Disc* yang dijual di pasaran dapat menampung sejumlah besar lagu adalah hasil konversi sinyal suara analog ke dalam bentuk digital. Film-film yang dapat dinikmati melalui DVD juga merupakan hasil dari rekayasa digital. Dan masih banyak lagi manfaat yang dapat kita rasakan saat ini dengan adanya teknologi digital.

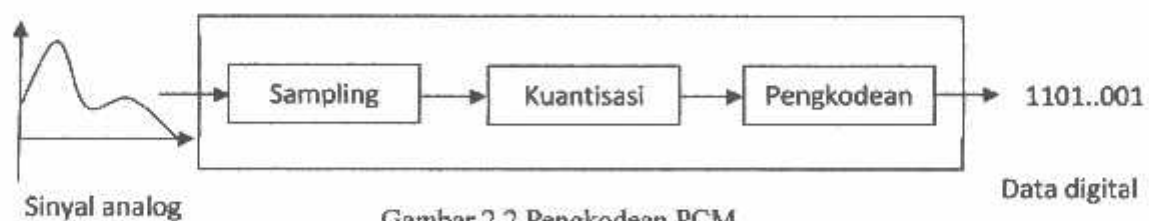
Untuk memperoleh data digital dibutuhkan suatu proses untuk mengubah sinyal analog menjadi data digital. Ada beberapa metode yang dapat digunakan. Namun dalam sub-bab ini kita hanya akan membahas dua metode yang paling banyak digunakan, yaitu *Pulse Code Modulation (PCM)* dan Modulasi Delta (Delta modulation).

- ***Pulse Code Modulation (PCM)***

PCM merubah sinyal analog menjadi data digital melalui proses awal yang disebut dengan *sampling*. Sampling adalah proses mencacah sinyal analog menjadi potongan-potongan sinyal dengan amplitudo sesuai dengan sinyal asli. Setelah didapatkan sinyal hasil sampling, sinyal tersebut selanjutnya dikuantisasi. Kuantisasi adalah proses pembulatan amplitudo sinyal terkuantisasi ke bilangan integer terdekat. Proses terakhir adalah melakukan pengkodean digital terhadap kode hasil kuantisasi. Jadi dapat disimpulkan bahwa PCM menggunakan tiga langkah utama di dalam mengubah sinyal analog menjadi data digital, yaitu:

- Proses pencacahan (*sampling*),
- Proses kuantisasi, dan
- Pengkodean data digital.

Untuk lebih jelas lihat pada *Gambar 2.2*.

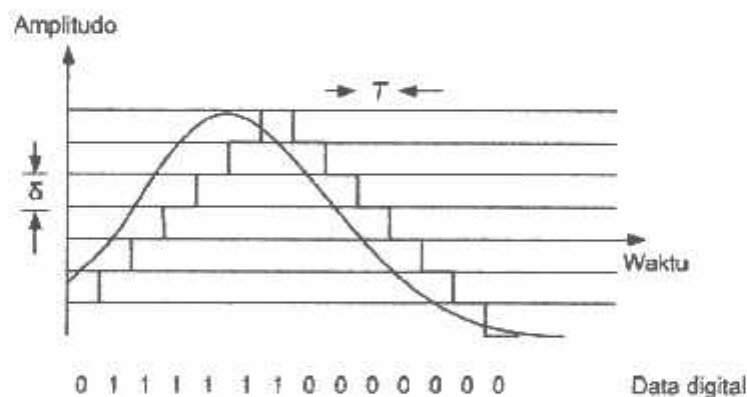


Gambar 2.2 Pengkodean PCM

• *Modulasi Delta*

Teknik konversi dari sinyal analog menjadi data digital akan menjadi lebih sederhana apabila diimplementasikan dengan menggunakan Modulasi Delta daripada menggunakan PCM. Modulasi Delta tidak mendeteksi amplitudo sebagaimana halnya pada PCM, melainkan mendeteksi perubahan amplitudo antara cacahan saat ini dengan cacahan sebelumnya. Perbedaan antara amplitudo saat ini dengan amplitudo sebelumnya disebut dengan δ . Perhatikan ilustrasi Modulasi Delta dalam Gambar 2.3. Apabila δ bernilai positif, maka Modulasi Delta akan membangkitkan bit 1, sebaliknya apabila δ bernilai negatif maka Modulasi Delta akan membangkitkan bit 0. Dengan demikian keluaran dari Modulasi Delta merupakan deretan bit yang menggambarkan perubahan amplitudo dari sinyal analog.

Untuk dapat menghasilkan unjuk kerja Modulasi Delta yang lebih baik, δ dapat dibuat menjadi adaptif. Dengan menggunakan Modulasi Delta Adaptif nilai δ akan berubah-ubah mengikuti amplitudo dari sinyal analog.



Gambar 2.3 Ilustrasi Modulasi Delta

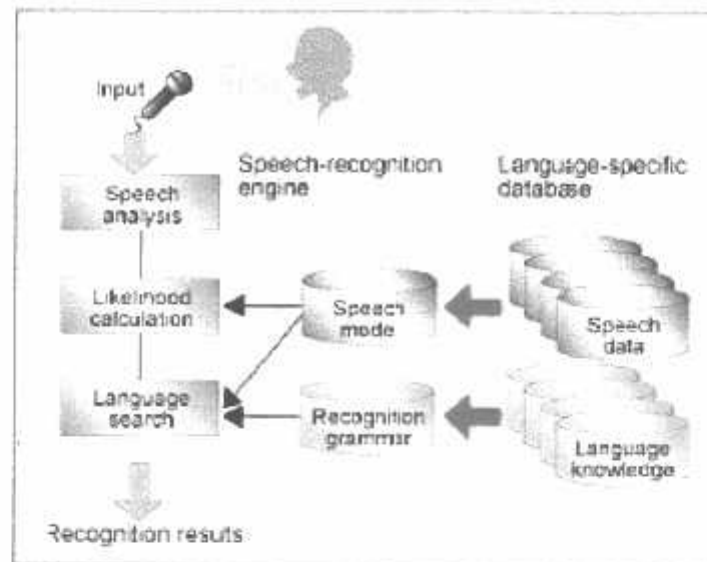
Sebagaimana halnya pada PCM, kesalahan akibat kuantisasi juga terjadi pada Modulasi Delta. Namun secara umum dapat dikatakan bahwa kesalahan kuantisasi dari Modulasi Delta lebih kecil daripada PCM. (*Teknik Jaringan Multimedia, umg, 2010*).

2.2 Speech Recognition

Teknologi pengenalan suara adalah teknologi yang menggunakan peralatan dengan sumber masukannya adalah suara, seperti *microphone* untuk menginterpretasikan suara manusia untuk transkripsi atau sebagai metode alternatif interaksi dengan komputer. Teknologi pengenalan suara tidak sama dengan teknologi *voice recognition* yang hanya mengenali suara sebagai identifikasi keamanan. Sistem komersial untuk pengenalan suara telah ada sejak 1990. Walaupun teknologi ini sukses, masih sedikit orang yang menggunakan sistem pengenalan pengenalan suara pada komputer. Kebanyakan pengguna komputer dalam berinteraksi dengan komputernya merasa lebih nyaman dengan menggunakan peralatan-peralatan masukan konvensional yaitu keyboard dan mouse, walaupun kenyataannya dengan menggunakan teknologi pengenalan suara memungkinkan pengguna untuk berbicara secara langsung dan cepat serta efisien.

Pada Speech Recognition, masukan sistem adalah ucapan atau suara manusia, selanjutnya sistem akan mengidentifikasi kata atau kalimat yang diucapkan dan menghasilkan teks yang sesuai dengan apa yang diucapkan. Sinyal ucapan pertama kali akan dilewatkan pada bagian penganalisis ucapan untuk mendapatkan besaran-besaran atau ciri-ciri yang mudah diolah pada tahap berikutnya. Untuk setiap ucapan yang berbeda akan dihasilkan pola ciri yang berbeda.

Penganalisis sintaks akan melakukan transformasi sinyal ucapan dari domain waktu ke domain frekuensi. Pada domain frekuensi, untuk kurun waktu yang singkat, setiap sinyal dapat terlihat memiliki ciri-ciri yang unik. Namun demikian, pengucapan suatu unit bunyi ucapan (*fonem*) seringkali bervariasi antar orang yang berbeda, juga terpengaruh oleh fonem-fonem disekitarnya, kondisi emosi, *noise*, dan faktor-faktor lainnya. Sistem *Speech Recognition* akan melakukan pengenalan untuk setiap unit bunyi pembentuk ucapan (*fonem*), selanjutnya mencoba mencari kemungkinan kombinasi hasil ucapan yang paling dapat diterima. Sistem yang lebih sederhana adalah sistem yang hanya dapat mengenal sejumlah kata yang jumlahnya terbatas. Sistem ini biasanya lebih akurat dan lebih mudah dilatih. Lihat pada *Gambar 2.4*, sebagai proses kerja Speech Recognition (*Endy, Anugrah, Tesis, UKM, 2007*)



Gambar 2.4 Proses Speech Recognition

2.2.1 Mode Sistem Pengenalan Pembicaraan

Terdapat dua mode pada sistem pengenalan pembicaraan yaitu:

- a. *Mode diktasi*. Pada mode ini pengguna komputer mengucapkan kata atau kalimat yang selanjutnya akan dikenali oleh komputer dan diubah menjadi data teks. Kemungkinan jumlah kata yang dapat dikenali dibatasi oleh jumlah kata yang telah terdapat pada database. Pengenalan mode diktasi merupakan speaker dependent. Keakuratan mode ini bergantung pada pola suara dan aksen pembicara serta pelatihan yang telah dilakukan.
- b. *Mode command and control*. Pada mode ini pengguna komputer mengucapkan kata atau kalimat yang sudah terdefinisi terlebih dahulu pada database dan selanjutnya digunakan untuk menjalankan perintah tertentu pada aplikasi komputer. Jumlah perintah yang dikenali tergantung dari aplikasi yang telah didefinisikan terlebih dahulu pada database. Mode ini merupakan speaker independent karena jumlah kata yang dikenali biasanya terbatas sekali dan ada kemungkinan pembicara tidak perlu melakukan pelatihan pada sistem sebelumnya.

2.2.2 Pemisahan Kata

Pemisahan kata adalah proses untuk memisahkan suara yang diucapkan oleh pengguna menjadi beberapa bagian. Masing-masing bagian dapat berupa kalimat atau satu kata. Terdapat tiga macam metode yang digunakan pada proses pemisahan kata ini yaitu : *discrete speech*, *word spotting* dan *continuous speech*.

- a Pada *discrete speech*, pengguna diharuskan mengucapkan kalimat secara terpenggal dengan adanya jeda sejenak diantara kata. Jeda tersebut digunakan oleh sistem untuk mendeteksi awal dan akhir sebuah kata. *Discrete speech* ini mempunyai kelebihan yaitu sedikit *resource* (memori komputer, waktu proses) yang digunakan oleh sistem untuk mendeteksi suara, tetapi mempunyai kelemahan yaitu ketidaknyamanan pengguna dalam mengucapkan sebuah kalimat.
 - b Pada *word spotting*, dalam sebuah kalimat yang diucapkan pengguna, sistem hanya mendeteksi kata yang terdapat di dalam perbendaharaan yang dimilikinya, dan mengabaikan kata lain yang tidak dimilikinya. Sehingga walau pengguna mengucapkan kalimat yang berbeda, bila di dalam kalimat tersebut terdapat sebuah kata yang sama dan terdapat di perbendaharaan sistem, maka hasil pengenalan akan sama. Kelemahan metode ini adalah besar kemungkinan sistem melakukan kesalahan arti pengenalan dalam bentuk kalimat. Tetapi metode ini mempunyai kelebihan yaitu pengguna dapat mengucapkan kalimat secara normal tanpa harus berhenti diantara kata.
 - c Pada metode *continuous speech*, sistem akan mengenali dan memproses setiap kata yang diucapkan. Metode ini cukup akurat dalam mengenali ucapan pengguna. Tetapi di samping itu metode ini memerlukan *resource* yang besar dalam prosesnya. Pada metode ini, sistem harus dapat mendeteksi awal dan akhir setiap kata dalam kalimat tanpa adanya jeda diantara kata tersebut, dan setelah berhasil memisahkan kata, langkahselanjutnya adalah mencocokkan dengan perbendaharaan kata yang dimilikinya.
-

2.2.3 Sifat Pengenalan Pembicaraan

Sistem pengenalan pembicaraan mempunyai beberapa sifat dilihat dari ketergantungan terhadap pengguna yaitu *speaker dependent*, *speaker independent* dan *speaker adaptive*.

- a Pada *speaker dependent*, sistem membutuhkan pelatihan untuk setiap pengguna yang akan menggunakan sistem tersebut. Sistem tidak akan dapat mengenali pengguna yang belum pernah melakukan pelatihan.
- b Pada *speaker independent*, pengguna tidak perlu melakukan pelatihan sebelum dapat menggunakan sistem, karena sistem mampu mengenali suara semua pengguna tidak tergantung warna suara dan dialek yang digunakan.
- c *Speaker adaptive* merupakan perpaduan dari *speaker dependent* dan *speaker independent*, dimana pengguna tidak perlu melakukan pelatihan dan keakuratan pengenalan sistem akan makin meningkat jika pengguna yang sama bekerja terus menerus selama beberapawaktu tertentu

2.2.4 Pencocokan kata

Pencocokan kata adalah proses untuk mencocokkan kata ucapan yang berhasil diidentifikasi dengan basis data yang dipunyai oleh sistem. Terdapat dua metode yang dapat dipakai pada proses pencocokan kata ini, yaitu : *whole-word matching* dan *phoneme matching*.

- a. Pada *whole-word matching*, sistem akan mencari di dalam basis data kata yang tepat persis dengan kata hasil ucapan pengguna.
- b. Sedangkan pada *phoneme matching*, sistem memiliki kamus *fonem*. (*fonem* adalah bagian terkecil dan unik dari suara yang membentuk sebuah kata.)

2.2.5 Perbendaharaan kata

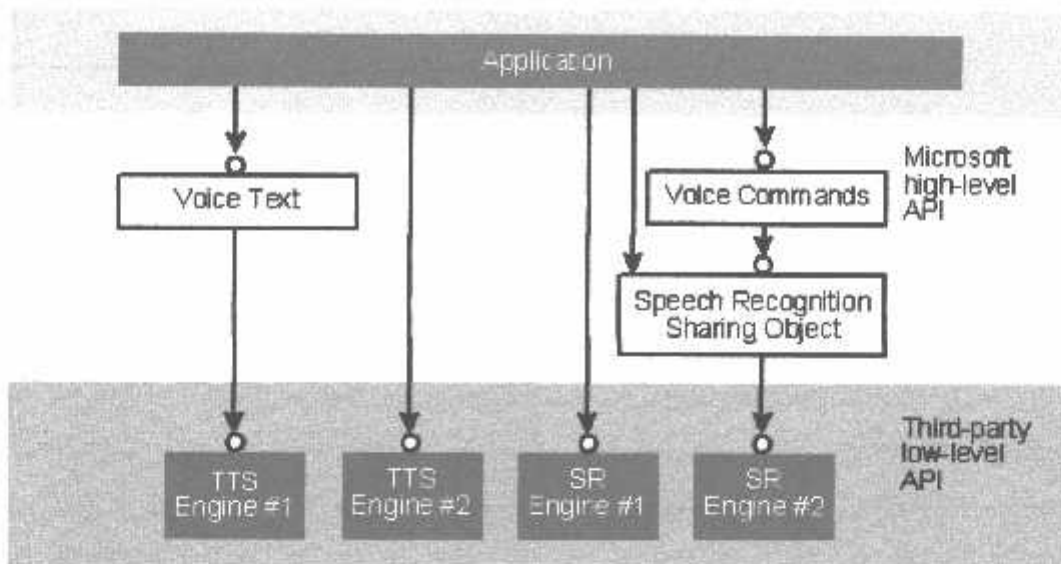
Perbendaharaan kata ialah bagian terakhir dalam sebuah sistem pengenalan pembicaraan. Terdapat dua hal yang perlu diperhatikan pada perbendaharaan kata, yaitu ukuran dan keakuratan. Jika perbendaharaan kata

berjumlah banyak maka sebuah sistem akan mudah dalam melakukan pencocokan kata, tetapi dengan makin meningkatnya jumlah perbendaharaan kata, maka jumlah kata yang mempunyai ucapan hampir sama juga meningkat, hal ini menurunkan keakuratan pengenalan. Dan sebaliknya, jika sebuah sistem mempunyai perbendaharaan kata sedikit, maka keakuratan pengenalan akan tinggi karena sedikitnya kata yang hampir sama, tetapi akan semakin banyak kata yang tidak dikenali. Untuk sistem pengenalan pembicaraan dengan mode *command and control*, akan lebih baik jika menggunakan jumlah perbendaharaan kata sedikit (kurang dari 100 kata), tetapi untuk mode *diktasi* akan membutuhkan jumlah perbendaharaan kata yang banyak.

2.3 Microsoft Speech Application Programming Interface

Speech Application Programming Interface (SAPI) merupakan *Application Programming Interface (API)* yang dikembangkan oleh Microsoft untuk digunakan pada *speech recognition* dan *speech synthesis* dengan sistem operasi berbasis Windows. Dengan menggunakan (API) tersebut dimungkinkan bagi pihak ketiga *3 party company* atau individu untuk mengembangkan perangkat *Speech Recognition* yang dapat bekerjasama dengan SAPI tersebut. Lihat Gambar 2.5.

Diperkenalkan oleh Microsoft pada tahun 1995, *Speech Application Programming Interface (SAPI)* memungkinkan sistem untuk mengenali suara manusia sebagai masukan dan keluaran berupa suara manusia dari text. Fasilitas ini menyediakan hubungan interaksi baru antara manusia dengan computer, *Speech Application Programming Interface (SAPI)* merupakan bagian dari *Windows Open Services Architecture model*. Pada *Speech Recognition (SR)* dan *Text to Speech (TTS)* terdapat beberapa *modules* yang disebut *engines*, yang telah tersedia pada semua sistem operasi Windows. Berikut merupakan arsitektur *Speech Application Programming Interface (SAPI)* : SAPI terdiri dari 2 antar muka yaitu *Application Programming Interface (API)* dan *Device Driver Interface (DDI)*. (Abdusy Syarif, *Seminar Nasional Aplikasi Teknologi, 2011*).



Gambar 2.5 Arsitektur Microsoft SAPI

Komponen utama di dalam SAPI 5 adalah sebagai berikut:

- a. *Voice Command*, sebuah obyek level tinggi untuk perintah dan kontrol menggunakan pengenalan suara.
- b. *Voice Dictation*, sebuah obyek level tinggi untuk *continous dictation speech recognition*.
- c. *Voice Talk*, sebuah obyek level tinggi untuk *speech synthesis*.
- d. *Voice Telephony*, sebuah obyek untuk menulis aplikasi telepon berbasis pengenalan suara.
- e. *Direct Speech Recognition*, sebuah obyek sebagai mesin untuk mengontrol pengenalan suara (*direct control of recognition engine*)
- f. *Direct Text to Speech*, sebuah obyek sebagai mesin yang mengontrol *synthesis*.
- g. *Audio Object*, untuk membaca dari audio device atau sebuah *file audio Option Explicit*

2.3.1 Application Programming Interface (API)

Pada sistem pengenalan pembicaraan, aplikasi akan menerima *event* pada saat suara yang diterima telah dikenali oleh *engine*.

Komponen SAPI yang menghasilkan *event* ini (API) diimplementasikan oleh antar muka *ISpNotifySource*. Lebih spesifik, SAPI menggunakan

SetNotifySink, yaitu aplikasi akan meneruskan *pointer* *IspNotifySink* ke *IspNotifySource::SetNotifySink*. *IspNotifySource::SetNotifySink* ini akan menerima pemanggilan melalui *IspNotifySink::Notify* ketika terdapat satu atau lebih *event* yang menyatakan bahwa aplikasi dapat mengambil data. Biasanya aplikasi tidak mengimplementasikan *IspNotifySink* secara langsung tetapi menggunakan *CoCreate Instance* untuk membuat obyek *IspNotifySink*, yang diimplementasikan oleh komponen *CLSID_SpNotify*. Obyek ini menyediakan antar muka *IspNotifyControl*.

Tetapi antar muka *IspNotifySource* dan *IspNotifySink* hanya menyediakan mekanisme untuk notifikasi dan tidak ada *event* yang ditimbulkan oleh notifikasi tersebut. Dengan menggunakan obyek *IspEventSource*, aplikasi dapat menerima informasi tentang *event* yang ditimbulkan oleh notifikasi. *IspEventSource* juga menyediakan mekanisme untuk menyaring dan membuat antrian *event*. Biasanya aplikasi tidak menerima notifikasi dari *IspEventSource* sampai terjadi pemanggilan terhadap *IspEventSource::SetInterest* untuk menentukan *event* mana yang akan menghasilkan notifikasi dan *event* mana yang berulang sehingga harus dimasukkan ke daftar antrian. *Event* tersebut diidentifikasi dengan menggunakan tanda *SPEVENTENUM*.

Ketika aplikasi menerima notifikasi, ada kemungkinan terdapat informasi yang sama pada beberapa *event*. Dengan memanggil *IspEventSource::GetInfo*, maka variabel anggota *ulCount* akan mengembalikan nilai yang berupa struktur *SPEVENTSOURCEINFO* yang didalamnya terdapat jumlah *event* yang mempunyai informasi yang sama. Dengan menggunakan *IspEventSource::GetEvents*, aplikasi akan mengeluarkan sejumlah struktur *SPEVENT*, dimana masing-masing mempunyai informasi tentang *event* tertentu. Untuk lebih jelas lihat *Gambar 2.6* dan *Tabel 2.1*.

Objects (RDO), atau ActiveX Data Object (ADO), serta menawarkan pembuatan kontrol ActiveX dan objek ActiveX. Beberapa bahasa skrip seperti Visual Basic for Applications (VBA) dan Visual Basic Scripting Edition (VBScript), mirip seperti halnya Visual Basic, tetapi cara kerjanya yang berbeda.

Para programmer dapat membangun aplikasi dengan menggunakan komponen-komponen yang disediakan oleh Microsoft Visual Basic Programprogram yang ditulis dengan Visual Basic juga dapat menggunakan Windows API, tapi membutuhkan deklarasi fungsi eksternal tambahan.

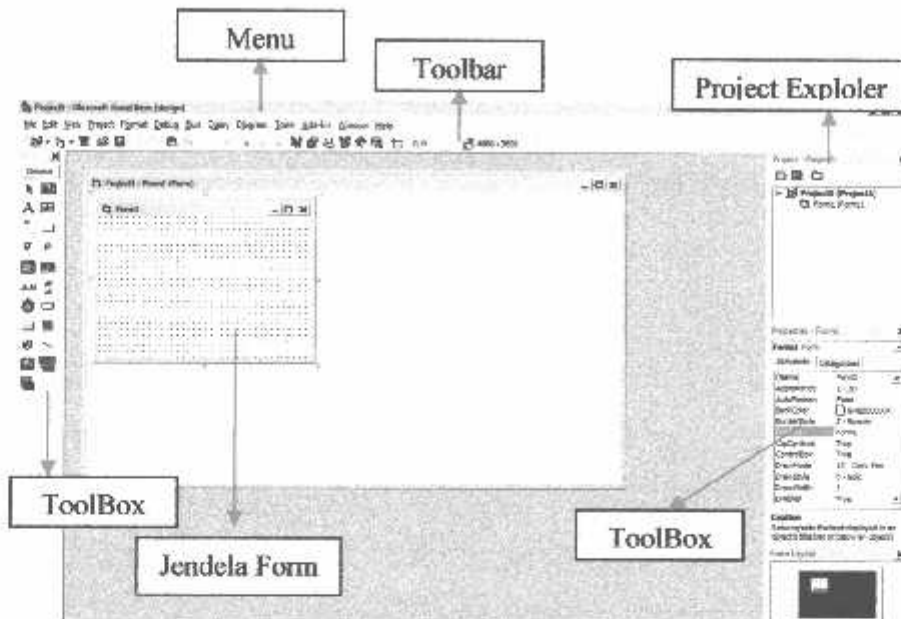
Dalam pemrograman untuk bisnis, Visual Basic memiliki pangsa pasar yang sangat luas. Dalam sebuah survey yang dilakukan pada tahun 2005, 62% pengembang perangkat lunak dilaporkan menggunakan berbagai bentuk Visual Basic, yang diikuti oleh C++, JavaScript, C#, dan Java.

Visual Basic merupakan bahasa yang mendukung OOP, namun tidak sepenuhnya. Beberapa karakteristik obyek tidak dapat dilakukan pada Visual Basic, seperti *Inheritance* tidak dapat dilakukan pada class module. *Polymorphism* secara terbatas bisa dilakukan dengan mendeklarasikan *class module* yang memiliki *Interface* tertentu. Visual Basic (VB) tidak bersifat case sensitif.

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 pilih 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.

2.5.1 Interface Visual Basic 6.0

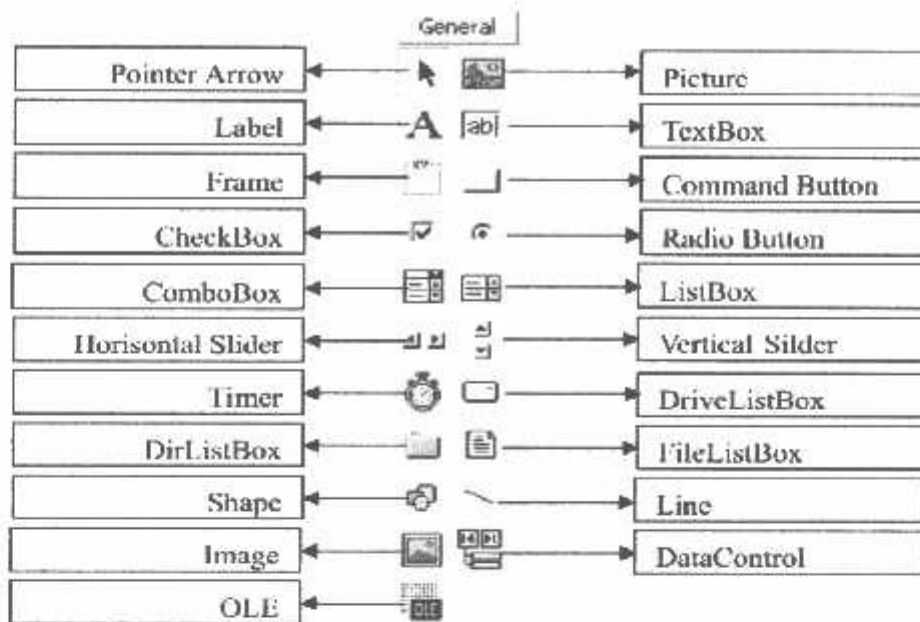
Interface Visual Basic 6.0, berisi menu, toolbar, toolbox, form, project explorer dan property seperti terlihat pada *Gambar 2.8*.



Gambar 2.8 Interface Visual Basic 6.0

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.

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 2.9.



Gambar 2.9. Komponen Standar Dalam Toolbox

2.5.2 Konsep Dasar Pemrograman Dalam Visual Basic 6.0

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

- 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 variable (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

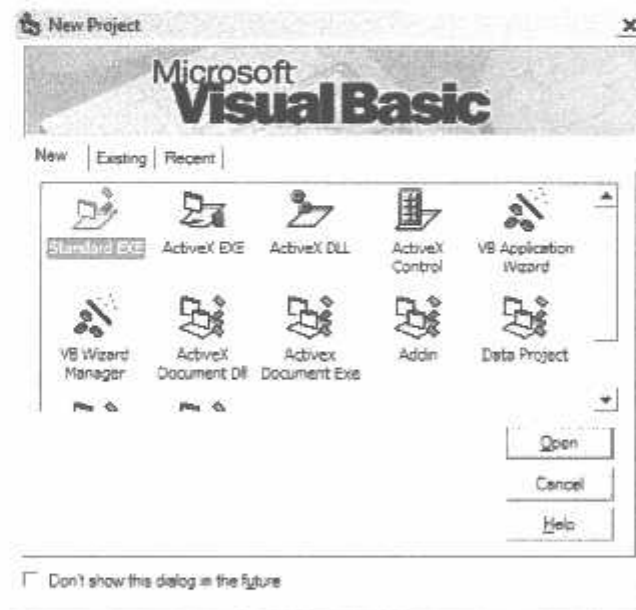
- Metode: bahwa jalannya program dapat diatur sesuai aplikasi dengan menggunakan metode pemrograman yang diatur sebagai aksi dari

setiap komponen. Metode inilah tempat untuk mengekspresikan logika pemrograman dari pembuatan suatu program aplikasi.

- c. 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.

2.5.3 Membuat Project Baru

Untuk memulai pembuatan program aplikasi di dalam Visual Basic, yang dilakukan adalah membuat project baru. Project adalah sekumpulan form, modul, fungsi, data dan laporan yang digunakan dalam suatu aplikasi. Membuat project baru dapat dilakukan dengan memilih menu `[File] >> [New Project]` atau dengan menekan ikon `[new project]` pada Toolbar yang terletak di pojok kiri atas. Setelah itu akan muncul konfirmasi untuk jenis project dari program aplikasi yang akan dibuat seperti terlihat pada *Gambar 2.10*.



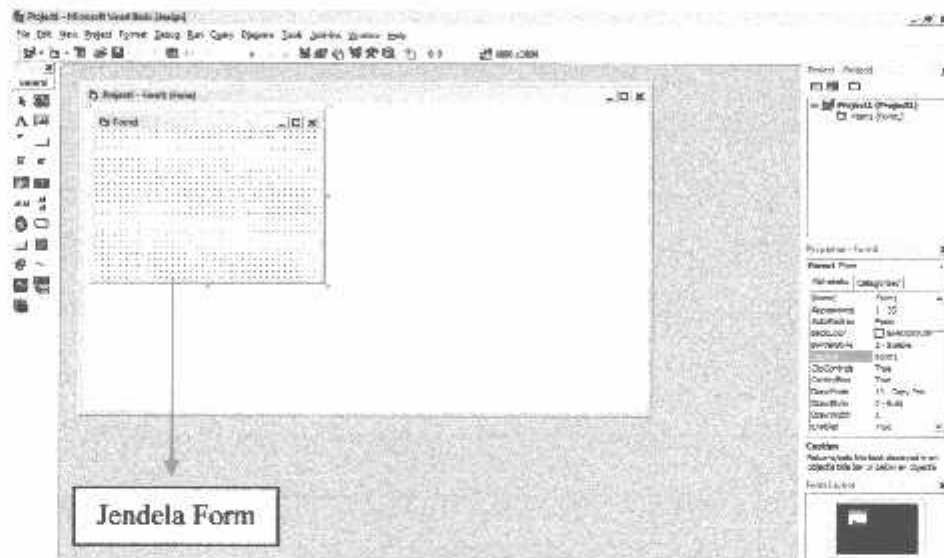
Gambar 2.10. Layar Pemilihan Jenis Project.

Visual Basic 6.0 menyediakan 13 jenis project yang bisa dibuat seperti terlihat pada *Gambar 2.10*. Ada beberapa project yang biasa digunakan oleh banyak pengguna Visual Basic, antara lain:

- a. *Standard EXE*: Project standar dalam Visual Basic dengan komponen-komponen standar. Jenis project ini sangat sederhana, tetapi memiliki keunggulan bahwa semua komponennya dapat diakui oleh semua unit komputer dan semua user meskipun bukan administrator. Pada buku ini akan digunakan project Standard EXE ini, sebagai konsep pemrograman visualnya.
- b. *ActiveX EXE*: Project ini adalah *project ActiveX* berisi komponen-komponen kemampuan untuk berinteraksi dengan semua aplikasi di sistem operasi windows.
- c. *ActiveX DLL*: Project ini menghasilkan sebuah aplikasi library yang selanjutnya dapat digunakan oleh semua aplikasi di sistem operasi windows.
- d. *ActiveX Control*: Project ini menghasilkan komponen-komponen baru untuk aplikasi Visual Basic yang lain e. *VB Application Wizard*: Project ini memandu pengguna untuk membuat aplikasi secara mudah tanpa harus pusing-pusing dengan perintah-perintah pemrograman.
- f. *Addin*: Project seperti *Standard EXE* tetapi dengan berbagai macam komponen tambahan yang memungkinkan kebebasan kreasi dari pengguna.
- g. *Data project*: Project ini melengkapi komponennya dengan komponen komponen database. Sehingga bisa dikatakan project ini memang disediakan untuk keperluan pembuatan aplikasi database.
- h. *DHTML Application*: Project ini digunakan untuk membuat aplikasi internet pada sisi client (*client side*) dengan fungsi-fungsi DHTML.
- i. *IIS Application*: Project ini menghasilkan aplikasi internet pada sisi server (*server side*) dengan komponen-komponen CGI (*Common Gateway Interface*).

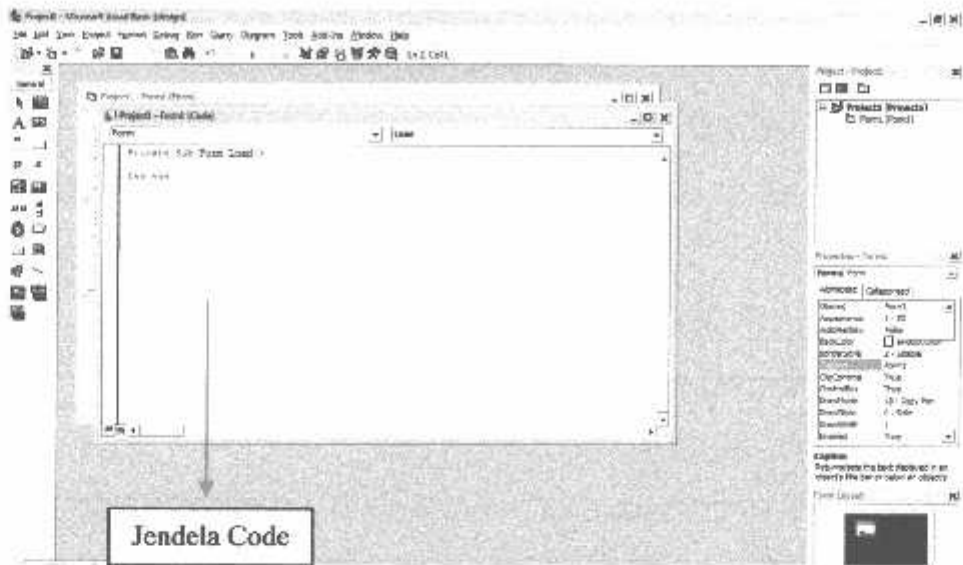
Selanjutnya pilih *Standard EXE* dan tekan [Ok]. Lalu muncul tampilan dari *Standard Exe*. Dengan demikian project sudah siap dibuat.

Dalam pembuatan project sebelumnya *double click* pada form yang terbuat maka akan terlihat jendela tersembunyi (*hidden windows*) yang berupa jendela untuk pembuatan program atau jendela kode (*code windows*). Hal ini dapat dilakukan dengan cara memilih ikon jendela form atau jendela kode yang ada di [Project Explorer]. Hal ini dapat dilihat pada *Gambar 2.11* dan *Gambar 2.12*.



Gambar 2.11. Jendela Form

Pada jendela form, digunakan dalam membangun tampilan dari program aplikasi yang akan dibuat dengan mengatur komponen-komponen baik letak, properti dan eventnya. Untuk mengambil suatu komponen dari [Toolbox] dapat Jendela Form dilakukan dengan klik komponen tersebut, kemudian klik atau tarik pada posisi yang benar pada form.



Gambar 2.12. Jendela Kode

Sedangkan, pada jendela kode, digunakan untuk menuliskan program dari komponen-komponen yang sudah diletakkan pada jendela form sesuai dengan aplikasi yang akan dibuat.

BAB III

ANALISA DAN PERANCANGAN

3.1 Analisa

Untuk merancang aplikasi ini, diperlukan analisa sistem dan perangkat sebagai identifikasi kebutuhan sistem pada aplikasi.

3.1.1 Analisa Speech Application Programming Interface

Langkah pertama pembuatan sistem pengenalan pembicaraan adalah pembuatan antar muka utama bagi aplikasi. Antar muka utama yang digunakan oleh aplikasi ialah ISpRecoContext. Antar muka ini merupakan sebuah ISpEventSource, yang berarti ISpRecoContext tersebut adalah antar muka aplikasi untuk menerima notifikasi dari sebuah *event* pengenalan pembicaraan. Aplikasi menggunakan *engine* pengenalan pembicaraan (ISpRecognizer). Dengan menggunakan *shared recognizer* yang memungkinkan untuk berbagi *resource* dengan aplikasi pengenalan pembicaraan yang lain. Untuk membuat ISpRecoContext untuk *shared* ISpRecognizer, aplikasi hanya memerlukan pemanggilan terhadap CoCreateInstance (COM) pada komponen CLSID_SpSharedRecoContext. Dimana SAPI akan mengatur *audio input stream* sesuai dengan pengaturan standard SAPI. Shared ISpRecognizer ini paling banyak digunakan pada aplikasi pengenalan pembicaraan secara umum. Langkah berikut pembuatan sistem pengenalan pembicaraan adalah mengatur notifikasi untuk *event* yang dibutuhkan oleh aplikasi. Karena ISpRecognizer adalah ISpEventSource yang juga merupakan ISpNotifySource, maka aplikasi dapat memanggil salah satu metode IspNotifySource dari ISpRecoContext untuk memberitahukan dimana terdapat sebuah *event* dari ISpRecoContext.. Kemudian aplikasi juga harus memanggil ISpEventSource::SetInterest untuk menyatakan *event* mana yang perlu untuk dinotifikasikan. *Event* yang paling penting adalah *SPEI_RECOGNITION*, yang berfungsi untuk menyatakan bahwa ISpRecognizer telah mengenali suara tertentu dari ISpRecoContext.

Langkah terakhir pembuatan sistem pengenalan pembicaraan ini, aplikasi harus membuat, meletakkan di memori (*load*) dan mengaktifkan

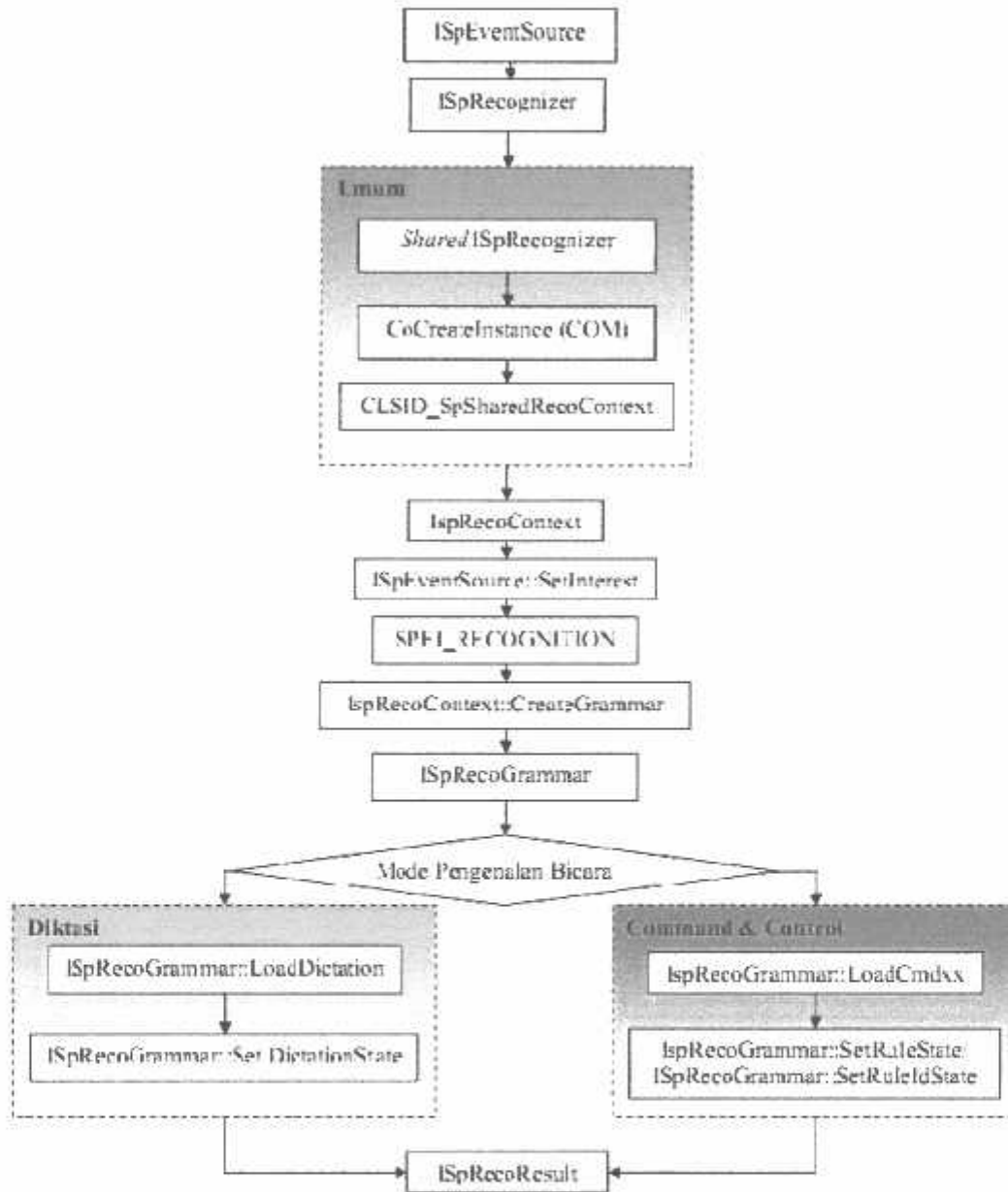
IspRecoGrammar sesuai dengan mode pengenalan pembicaraan, misal : *diktasi* atau *command and control*. Pertama kali, aplikasi harus membuat IspRecoGrammar menggunakan IspRecoContext::CreateGrammar. Kemudian aplikasi harus meletakkan *grammar* yang telah dibuat tersebut pada memori (*load*) yaitu dengan memanggil IspRecoGrammar::LoadDictation untuk mode *diktasi* atau IspRecoGrammar::LoadCmdxxx untuk mode *command and control*. Untuk mengaktifkan *grammar* sehingga sistem pengenalan pembicaraan dapat bekerja, aplikasi harus memanggil IspRecoGrammar::Set DictationState untuk mode *diktasi* atau IspRecoGrammar::SetRuleState atau IspRecoGrammar::SetRuleIdState untuk mode *command and control*.

Ketika terjadi notifikasi pada saat pengenalan pembicaraan bekerja, maka IParam yang merupakan variabel anggota dari struktur *SPEVENT* akan menjadi IspRecoResult yang kemudian digunakan oleh aplikasi untuk dapat menentukan apa yang telah terkenali dan sekaligus menentukan IspRecoGrammar mana yang harus digunakan. IspRecognizer, baik shared ataupun InProc, dapat mempunyai IspRecoContext lebih dari satu dan masing-masing IspRecoContext dapat menerima notifikasi sesuai dengan *event* yang telah didefinisikan. Sebuah IspRecoContext dapat mempunyai lebih dari satu IspRecoGrammars di mana masing-masing IspRecoGrammar tersebut digunakan untuk mengenali tipe grammar yang berbeda.

Tabel 3.1 Method Speech Application Programming Interface

Method	Penjelasan
IspEventSource	Turunan dari IspEventSource dan seluruh <i>method</i> -nya dapat diakses dari IspRecoContext.
SpRecoContext::CreateGrammar	Membuat objek SpGrammar.
IspRecognizer	Digunakan untuk mengkonfigurasi atau mengambil <i>attributes</i> dari SR engine
IspRecoGrammar	Digunakan untuk mengatur kata-kata dan <i>phrase</i> yang akan dikenali SR engine.
IspRecoGrammar::LoadDictation	Mengambil dan menginisialisasi <i>Dictation topic</i> .
IspRecoGrammar::SetDictationState	Mengatur <i>Dictation state</i> menjadi aktif atau tidak aktif.

SpRecoResult	Digunakan oleh aplikasi untuk mengambil informasi tentang hasil dari SR engine (<i>hypotheses, recognitions, dan false recognitions</i>)
--------------	--



Gambar 3.1 Diagram Alir Speech Application Programming Interface

3.1.2 Analisa Kebutuhan Perangkat

Kebutuhan perangkat yang digunakan dalam membuat Aplikasi ini meliputi perangkat lunak dan perangkat keras yang dihubungkan dengan sistem operasi.

Adapun perangkat lunak yang dibutuhkan dalam pembuatan aplikasi meliputi :

a. *Sistem Operasi*

Untuk membuat aplikasi perintah suara ini penulis menggunakan sistem operasi Microsoft Windows 7 Ultimate.

b. *Microsoft Visual Basic 6.0,*

Merupakan perangkat utama dalam membuat Aplikasi, dengan menggunakan *components* pendukung yang disediakan antara lain Microsoft Voice Commands, Voice Dictation, Voice Text dan Direct Speech Recognition (DirectSR).

c. *Microsoft Speech Recognition*

Perangkat lunak ini berguna agar komputer dapat mengenali perintah yang kita ucapkan.

d. *Microsoft Speech Application Programming Interface*

Merupakan perangkat lunak yang akan mengolah input suara dan akan mencocokkan dengan data *library* yang terdapat dalam *engine*.

Dan adapun perangkat keras untuk pembuatan Aplikasi ini adalah :

a. *Mainboard Intel Pentium Core 2 Duo.*

b. *Processor Intel(R) Core 2 Duo @ 2.00GHz*

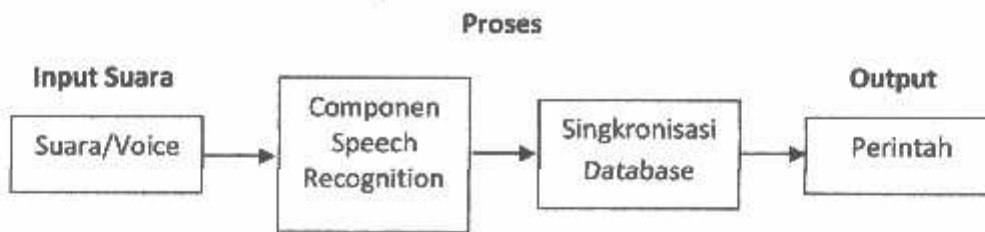
c. *Memory 2 GB.*

d. *Hardisk Toshiba 320 GB.*

e. *VGA NVIDIA GeForce G 105M*

f. *Sound Realtek High Definition Audio.*

3.2 Perancangan Blok Diagram



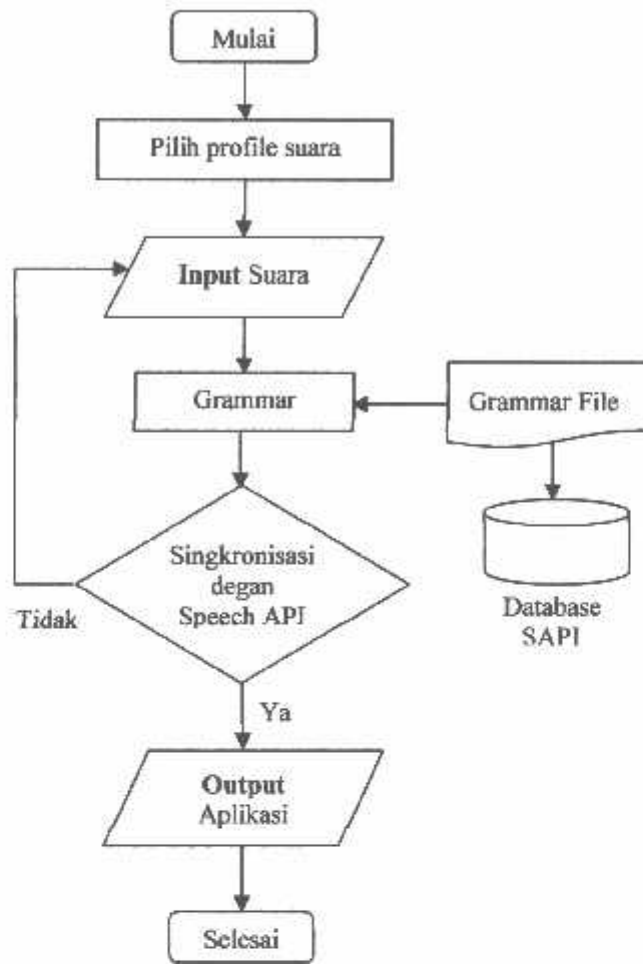
Gambar 3.2 Blok Diagram Sistem

Pada *Gambar 3.2* menunjukkan ilustrasi dari Input yang kemudian menghasilkan Output. Secara umum prinsip kerja dari Aplikasi ini adalah ketika seseorang berbicara kepada komputer, program menangkap suara tersebut melalui *microphone* dan mengubahnya menjadi sinyal digital, kemudian program menganalisa sinyal digital tersebut dengan membandingkan dan mencocokkannya dengan *digital pattern* yang ada dalam databasenya. Setelah itu akan diambil *digital pattern* yang paling besar prosentase kemiripannya, kemudian dari *digital pattern* tersebut diubah menjadi teks dan mengubahnya menjadi sebuah perintah untuk menjalankan suatu aplikasi atau program.

3.3 Perancangan Diagram Alir (Flow Chart)

Pada *Gambar 3.3*, merupakan Diagram Alir (Flow Chart) Aplikasi Perintah Suara

Pada saat aplikasi diaktifkan, *input* yang akan didapatkan merupakan sinyal suara pengguna. Sinyal suara tersebut akan diubah menjadi *Text (speech-to-text)* yang akan diproses berdasarkan struktur *grammar* yang terhubung dengan database *Speech API*. Kemudian terjadilah proses *Singkronisasi suara* untuk membedakan proses yang akan dilakukan untuk setiap kata tersebut, dan selanjutnya menjadi *output* sebagai aplikasi. Seperti pada *Gambar 3.3*.

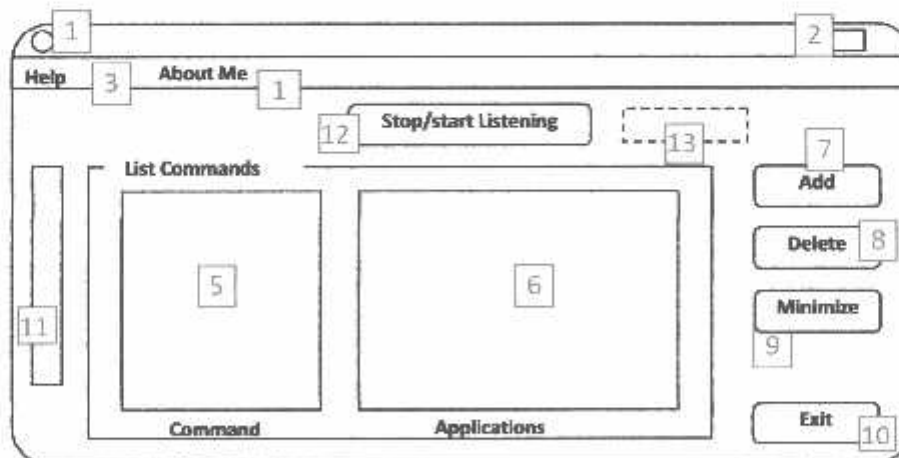


Gambar 3.3 Flow Chart Aplikasi Perintah Suara

3.4 Perancangan Tampilan Aplikasi

Berikut ini rancangan tampilan aplikasi Perintah Suara :

a. Tampilan Awal

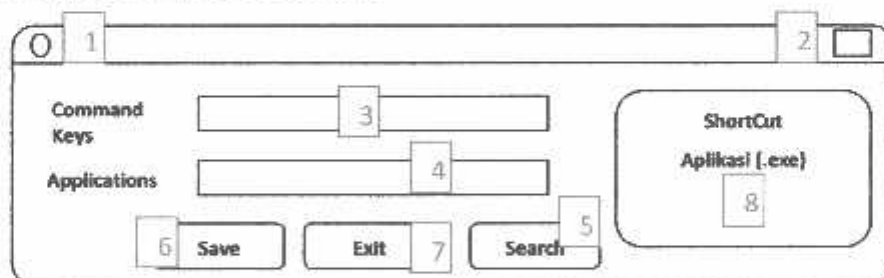


Gambar 3.4 Rancangan Tampilan Awal

Keterangan Tampilan Awal :

1. Icon aplikasi perintah suara
2. *Shortcut* untuk menutup aplikasi
3. Menu *Help* sebagai bantuan untuk pengguna (*MenuEditor*)
4. Menu *About Me* menampilkan Profile Pembuat (*MenuEditor*)
5. List perintah yang berhasil dimasukan oleh pengguna (*ListBox*)
6. List letak aplikasi yang akan dipanggil melalui suara (*ListBox*)
7. Tombol untuk memasukan perintah baru (*Command Button*)
8. Tombol untuk menghapus perintah yang berada di List perintah dan List letak aplikasi berada (*Command Button*)
9. Tombol untuk meminimalisasi Aplikasi sebagai *Icon Tray* (*Command Button*)
10. Tombol untuk keluar dari aplikasi (*Command Button*)
11. Monitoring sebagai respon masukan suara (*Progress Bar*)
12. Tombol untuk memulai dan mengakhiri perintah suara (*Command Button*).
13. Monitoring Dictation (pengucapan) suara (*Label*)

b. Tampilan Menambah Perintah



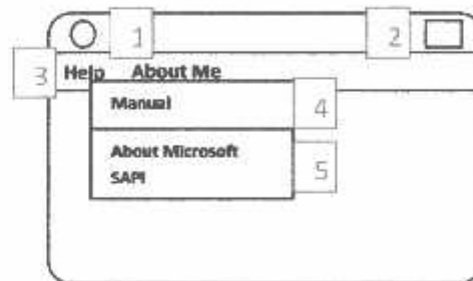
Gambar 3.5 Rancangan Menambah Perintah

Keterangan Tampilan Menambah Perintah :

1. Icon aplikasi
2. *Shortcut* untuk menutup “tampilan menambah perintah”
3. Tempat pengguna mengetikan perintah yang akan digunakan ketika akan melakukan suatu pemanggilan (*TextBox*)
4. Tempat pengguna memasukan tempat file yang akan dipanggil berada (*TextBox*)

5. Tombol bantuan untuk mencari file yang diinginkan (*CommandButton*)
6. Tombol untuk menyimpan input aplikasi perintah suara (*CommandButton*)
7. Tombol untuk kembali ke tampilan awal (*CommandButton*).
8. Area Shortcut dari Aplikasi yang berformat (.exe).

c. Tampilan Menu Help/Bantuan



Gambar 3.6 Rancangan Tampilan Menu Help

Keterangan Menu Help/Bantuan :

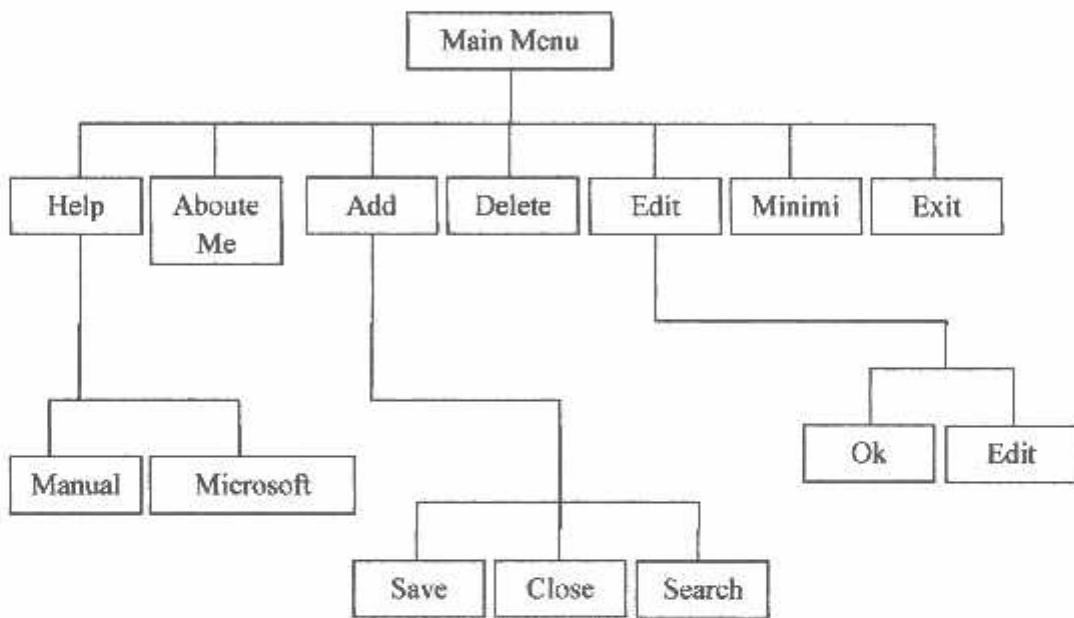
1. Icon aplikasi
2. *Shortcut* untuk menutup tampilan menu bantuan
3. Menu *Help* (*MenuEditor*)
4. Sub Menu *Manual*, berisi petunjuk penggunaan aplikasi perintah suara
5. Sub Menu *Microsoft SAPI*, berisi informasi mengenai Microsoft SAPI

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Perangkat Lunak

Aplikasi ini merupakan aplikasi yang hanya digunakan untuk memanggil atau mengaktifkan file bertipe *.exe dan direktori aplikasi yang akan dipanggil diketahui tempatnya dan telah tersimpan dalam aplikasi perintah suara.

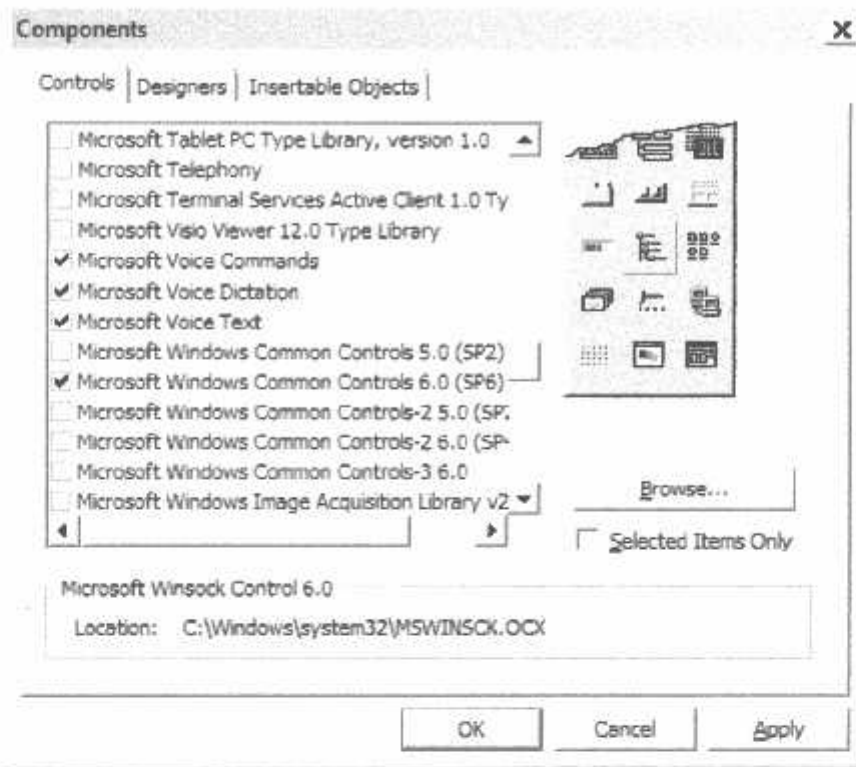
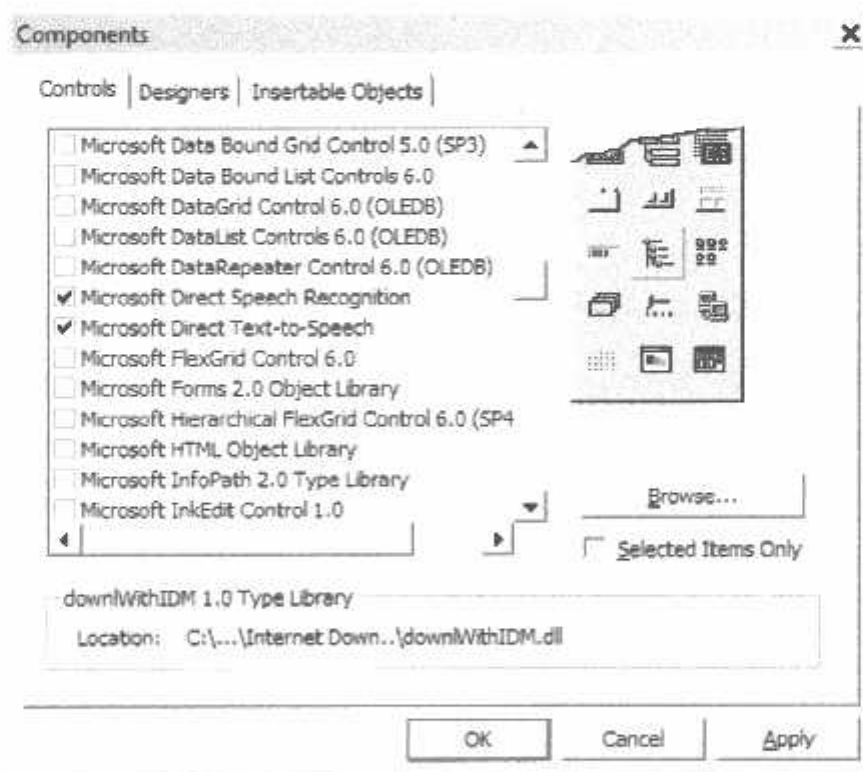
Berikut merupakan struktur menu (*structure chart*) dari aplikasi ini.



Gambar 4.1 Struktur Menu

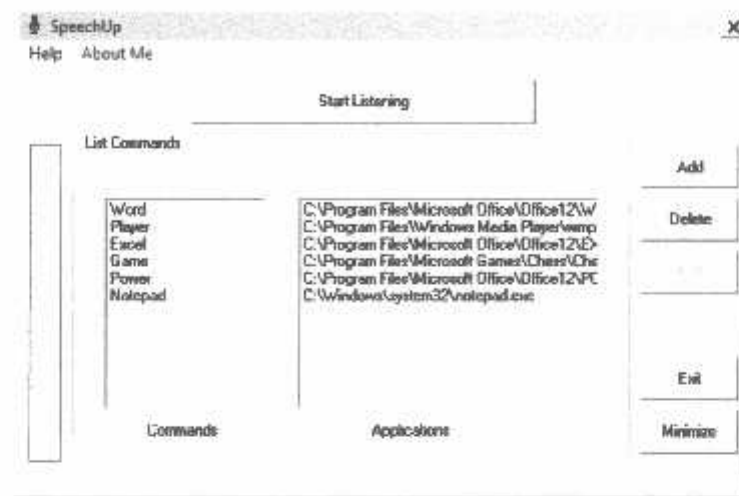
Berikut ini adalah keterangan dan alur kerja dari masing-masing menu yang terdapat pada aplikasi ini, sebagai berikut :

Dengan menggunakan *components* yang terdapat pada Visual Basic 6.0, yakni Speech Recognition, Voice Command, Voice Dictation, dan Voice Text. Aplikasi ini akan tercipta dan digabungkan dengan *Microsoft Speech Application Programming Interface*.

Gambar 4.2 *Components* pada Visual Basic 6.0 (1)Gambar 4.3 *Components* pada Visual Basic 6.0 (2)

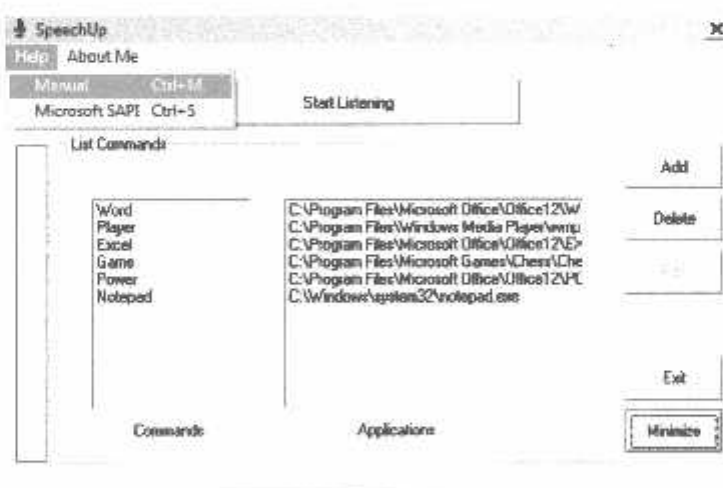
4.1.1 Implementasi Aplikasi Perintah Suara

Berikut pada Gambar 4.4 merupakan tampilan utama yang akan pengguna lihat adalah tampilan awal aplikasi



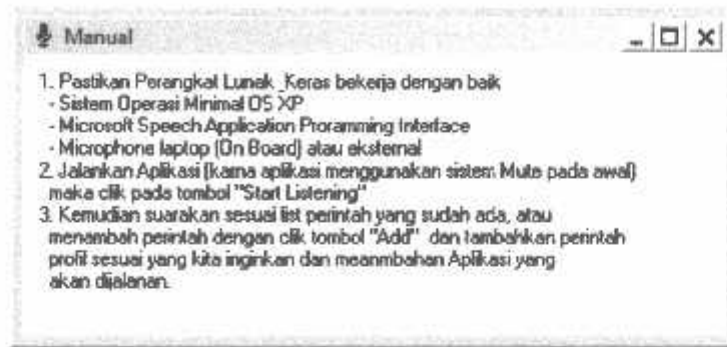
Gambar 4.4 Implementasi Tampilan Menu Utama

Pengguna harus memastikan microphone yang digunakan berjalan dengan baik, agar aplikasi dapat bekerja secara maksimal, dengan parameter yang tersedia pada menu editor *Help*, kemudian Sub Menu *Manual* yang terletak di pojok kiri atas tampilan awal, pengguna diingatkan agar microphone dicek terlebih dahulu agar bekerja dengan baik.



Gambar 4.5 Implementasi Tampilan *Help Manual*

Setelah *Menu Manual* di klik maka akan keluar tampilan seperti pada *Gambar 4.6*



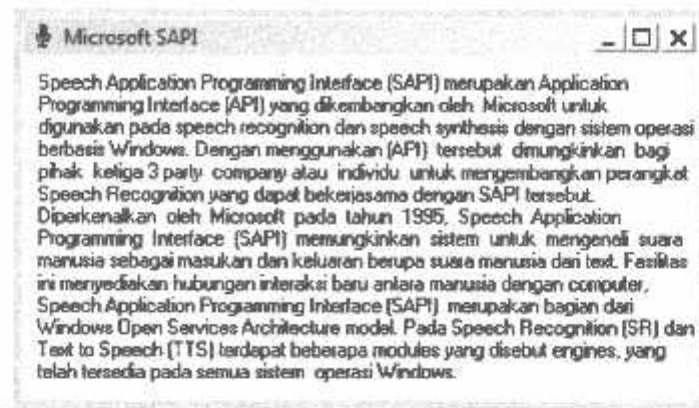
Gambar 4.6 Implementasi Tampilan *Manual*

Selain Sub Menu *Manual* untuk membantu pengguna dalam penggunaan aplikasi ini, terdapat *Sub Menu Microsoft SAPI* sebagai keterangan sederhana tentang Microsoft SAPI yang masih belum banyak diketahui oleh orang banyak.



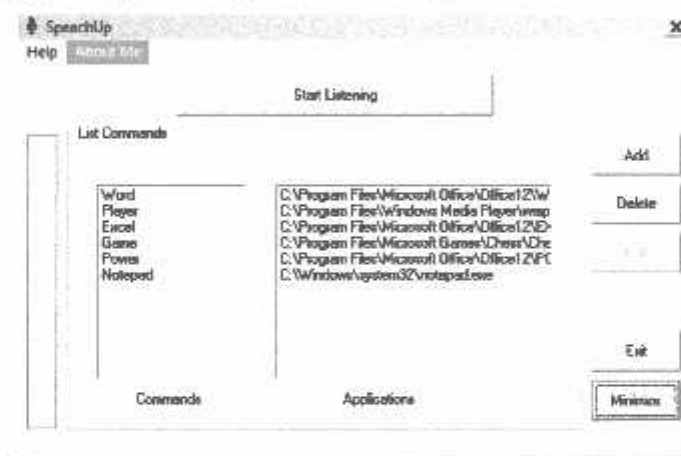
Gambar 4.7 Implementasi Tampilan *Help_Microsoft SAPI*

Maka, setelah Sub Menu *Microsoft SAPI* di klik, akan keluar tampilan seperti pada *Gambar 4.8*



Gambar 4.8 Implementasi Tampilan *Microsoft SAPI*

Kemudian pada menu utama terdapat menu editor *About Me* yang menampilkan *Profile Pembuat Aplikasi*.



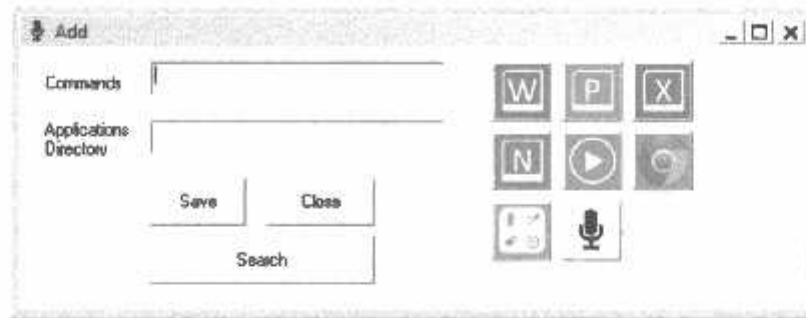
Gambar 4.9 Implementasi Tampilan Menu *About Me*

Setelah *Menu About Me* di klik, maka akan keluar tampilan seperti pada *Gambar 4.10*



Gambar 4.10 Implementasi Tampilan *About Me*

Pengguna juga bisa membuat kata kunci perintah sendiri dan memilih aplikasi yang akan dijalankan yang sudah tersedia di menu *Add* atau mencari file yang bertipe *.EXE yang ada di direktori Program Files pada komputer, seperti terlihat pada *Gambar 4.11*



Gambar 4.11 Implementasi Tampilan Add (*Tambah Perintah*)

Setelah kata kunci dimasukkan dan lokasi file yang akan dieksekusi telah diketahui pengguna akan menekan tombol *Save*, maka pengguna akan kembali pada tampilan awal aplikasi, dan pada list perintah dan list file yang akan dieksekusi terdapat data yang telah diinputkan berhasil tersimpan.

4.2 Pengujian

Data yang digunakan dalam pengujian Aplikasi ini adalah suara manusia (pengguna) secara langsung dengan media perantara sebuah *microphone* sebagai media pengujian dalam ruangan dan notebook sebagai media pengujian luar ruang.

Data masukan tersebut akan diuji dengan spesifikasi komputer sebagai berikut :

Tabel 4.1 Data Spesifikasi Perangkat

Perangkat	Personal Computer	Notebook
Processor	Intel Dual Core 2.00GHz	Intel Core 2 Duo 2.00GHz
Ram	1 Gb	2Gb
Adapter suara (<i>Soundcard</i>)	Realtek High Definition Audio	Realtek High Definition Audio
Media input	Headset + Microphone	Headset Bluetooth
Sistem Operasi	Windows 7 Ultimate	Windows 7 Ultimate

4.1.1 Pengujian Fungsional

Pengujian dilakukan pada tiga kondisi lingkungan sekitar (tempat pengujian) :

- A. Ruang keluarga dengan keadaan tingkat noise rendah (malam hari).
- B. Ruang keluarga dengan keadaan tingkat noise sedang (pagi hari).
- C. Tempat umum (halaman kampus) dengan keadaan tingkat noise tinggi (siang hari).

Proses pengujian ini akan mengeksekusi atau menjalankan beberapa program pada direktori komputer (*C:\Program Files*), seperti terlihat pada *Tabel 4.2* dengan tingkat frekuensi, jarak dan volume yang sama, dengan media input berupa *microphone* dalam kondisi yang sama

Tabel 4.2 Hasil Pengujian Fungsional

No	Input yang diucapkan	Pelafalan input	Respon berdasarkan kondisi lingkungan sekitar (tempat pengujian)		
			A	B	C
1	Word	Word	Berhasil	Berhasil	Gagal
2	Notepad	Notepad	Berhasil	Berhasil	Gagal
3	Player	Player	Berhasil	Berhasil	Gagal
4	Games	Games	Berhasil	Berhasil	Berhasil
5	Excel	Excel	Berhasil	Berhasil	Gagal
6	Point	Point	Berhasil	Berhasil	Gagal

Dari hasil pengujian dengan inputan yang diucapkan “Word, Notepad, Player, Games, Excel dan Point”, maka didapatkan hasil sebagai berikut :

- a. Penggunaan masukan kata kunci dengan tingkatan *noise* “A” (*noise* yang rendah) dapat dikenali dengan baik oleh aplikasi, hal ini sama
- b. ketika aplikasi dijalankan dilingkungan dengan tingkatan *noise* “B” (*noise* yang sedang) aplikasi masih dapat dengan mudah mengenalinya.
- c. Berbeda ketika pengujian dilakukan pada lingkungan yang mempunyai tingkatan *noise* “C” (*noise* yang tinggi), aplikasi tidak dapat mengenali

beberapa kata kunci dengan baik karena terlalu banyak noise yang dianggap sebagai inputan oleh Aplikasi, kecuali dengan pelafalan pada kata kunci “Games”, Aplikasi masih bisa mengenali dengan baik.

4.2.1 Pengujian User

Pengujian ini dilakukan kepada 10 orang Responden dengan menggunakan media *Kuesioner*, seperti pada *Tabel 4.3*

Tabel 4.3 Hasil Pengujian User

Prosedur	Baik	Cukup	Kurang
Tampilan Aplikasi	10	0	0
Melafalkan Kata Kunci	10	0	0
Menambah Kata Kunci & File Aplikasi	10	0	0
Menu Editor	8	2	0
Eksekusi Aplikasi	8	1	1

Dari 10 (Sepuluh) Responden dengan hasil data diatas dapat disimpulkan bahwa :

1. Responden mengatakan “**Baik**” adalah
 - a. Tampilan Aplikasi : 10 orang
 - b. Melafalkan Kata Kunci : 10 orang
 - c. Menambah Kata Kunci & File Aplikasi : 10 orang
 - d. Menu Editor : 8 orang
 - e. Eksekusi Aplikasi : 8 orang
2. Responden mengatakan “**Cukup**” adalah
 - a. Tampilan Aplikasi : 0 orang
 - b. Melafalkan Kata Kunci : 0 orang
 - c. Menambah Kata Kunci & File Aplikasi : 0 orang
 - d. Menu Editor : 2 orang
 - e. Eksekusi Aplikasi : 1 orang
3. Responden mengatakan “**Kurang**” adalah
 - a. Tampilan Aplikasi : 0 orang
 - b. Melafalkan Kata Kunci : 0 orang

- c. Menambah Kata Kunci & File Aplikasi : 0 orang
 - d. Menu Editor : 0 orang
 - e. Eksekusi Aplikasi : 1 orang
-

BAB V PENUTUP

5.1 Kesimpulan

Dari pengujian yang telah dilakukan, maka dapat disimpulkan bahwa aplikasi perintah suara ini dapat mengenali dengan baik masukan yang diinginkan oleh pengguna. Dan *Visual Basic 6.0* dengan *Speech Recognition* serta *Speech Application Programming Interface* dapat dimanfaatkan untuk merancang dan menciptakan aplikasi perintah suara.

Berikut ini hasil pengujian Aplikasi antara lain :

1. Pada kondisi noise rendah dan sedang, Aplikasi dapat berjalan baik dengan semua penggunaan kata kunci pada pengujian. Sedangkan pada kondisi noise tinggi, Aplikasi berjalan kurang baik, hal ini terlihat pada hasil pengujian yang hanya beberapa kata kunci yang bisa di kenali oleh Aplikasi.
2. Semua Responden yang berjumlah 10 Orang menilai bahwa Tampilan Aplikasi, Pelafalan Kata Kunci dan Menambah Kata Kunci & File Aplikasi File Aplikasi berjalan dengan ***Baik***.
3. Dari 10 Responden, 2 orang menilai Menu Editor ***Cukup***, dan 1 Responden yang menilai Eksekusi Aplikasi ***Kurang***

5.2 Saran

Dari Hasil diatas maka penulis mengharapkan kepada pengembang yang akan datang khususnya pada Aplikasi yang berkaitan dengan yang penulis rancang antara lain :

1. Aplikasi bisa menggunakan pelafalan bahasa yang diinginkan tidak hanya dalam bahasa Inggris.
2. Ada sistem filter suara agar tidak terlalu banyak noise sehingga Aplikasi berjalan dengan lebih baik.

DAFTAR PUSTAKA

- Didik Nurcahyono, *Pembuatan Speech Recognition Dan Database Wicara Untuk Kontrol Peralatan Rumah Tangga Jarak Jauh*, PENS-ITS, Surabaya, 2007.
- EEPIS ITS EDU, *Pembuatan Speech Recognition dan Database Wicara*. (<http://www.eepis-its.edu>).
- EEPIS ITS EDU, *Konsep Dasar Suara*. (<http://www.lecturer.eepisits.edu/Bab1KonsepDasar.pdf>).
- Google.2012.*Teori dasar pengenalan suara*. (<http://www.google.com>)
- Google.2012. *Microsoft visual Basic 6.0* (<http://hamdansalam.blogspot.com>).
- ittelkom, "*Konsep Digitalisasi Audio*", (<http://www.ittelkom.ac.id/staf/miq/Subject/Teknik>). 2012.
- Kurniadi Adi, 2001, Mei, "*Pemrogramman Microsoft Visual Basic 6.0*" Elex Media Komputindo.
- Microsoft Speech Application Interface*. (<http://en.wikipedia.org/wiki.com>).
- Modulasi Kode Pulsa* (http://su.wikipedia.org/wiki/Modulasi_kode_pulsa).
- Rinaldi, *Speech Recognition Menggunakan Algoritma Fast Fourier Transform Dan Divide And Conquer* (<http://blog.uinmalang.ac.id>).
- Teknik Jaringan Multimedia umg*, ([http://www.portal.umg.ac.id/Pemrosesan Sinyal Digital](http://www.portal.umg.ac.id/Pemrosesan_Sinyal_Digital)). 2010.

LAMPIRAN





PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting). Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Nomor : ITN-78/T.INF/TA/2013
Lampiran : -
Perihal : Bimbingan Skripsi

11 Mei 2013

Kepada : Yth. Bpk/Ibu Joseph Dedy Irawan, ST, MT
Dosen Pembimbing Program Studi Teknik Informatika S1
Institut Teknologi Nasional
M a l a n g

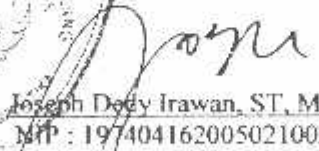
Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :


Nama : AHMAD MUZAKKY SYARIF
Nim : 0918190
Prodi : Teknik Informatika S1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Bpk/Ibu selama masa waktu 6 (enam) bulan, terhitung mulai tanggal :

11 Mei 2013 – 11 Nopember 2013

Sebagai satu syarat untuk menempuh Ujian Sarjana Teknik, Program Studi Teknik Informatika S1.
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S1
Ketua,

Joseph Dedy Irawan, ST, MT
MIP : 197404162005021002



Form S-4a



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km. 2 Telp. (0341) 417635 Fax. (0341) 417634 Malang

Nomor : ITN-78/T.LINE/TA/2013
Lampiran : -
Perihal : Bimbingan Skripsi

11 Mei 2013

Kepada : Yth. Bpk/Ibu Febriani Santi Wahyuni, Skom, Mkom,
Dosen Pembimbing Program Studi Teknik Informatika S1
Institut Teknologi Nasional
M a l a n g

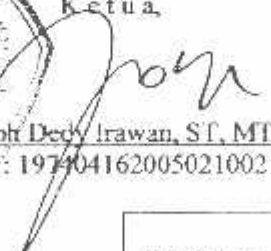
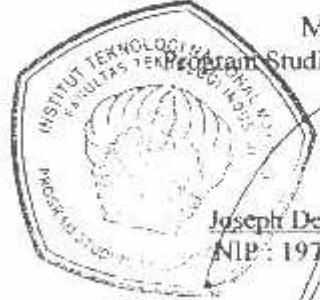
Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : AHMAD MUZAKKY SYARIF
Nim : 0918190
Prodi : Teknik Informatika S1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Bpk/Ibu selama masa waktu 6 (enam) bulan, terhitung mulai tanggal ;

11 Mei 2013 – 11 Nopember 2013

Sebagai satu syarat untuk menempuh Ujian Sarjana Teknik, Program Studi Teknik Informatika S1.
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S1
Ketua,

Joseph Dedy Irawan, S.T, MT
NIP. : 197404162005021002

Form S-4a




INTSITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Karanglo Km. 2 Malang

FORMULIR BIMBINGAN SKRIPSI

Nama : Ahmad Muzakky Syarif
NIM : 0918190
Masa Bimbingan : 11 Mei 2013 ⁹/_d 11 November 2013
Judul Skripsi : PERANCANGAN APLIKASI SPEECH RECOGNITION
SEBAGAI PERINTAH, MENGGUNAKAN MICROSOFT SAPI
UNTUK MENJALANKAN APLIKASI BERBASIS WINDOWS
DENGAN MENGGUNAKAN VISUAL BASIC 6.0

NO	TANGGAL	URAIAN	PARAF PEMBIMBING
1.	11-06-2013	Bab I & II	
2.	17-06-2013	Bab III	
3.	20-06-2013	Revisi Bab III & Demo Program	
4.	23-06-2013	Penambahan IconTray	
5.	26-06-2013	Demo Program IconTray	
6.	10-06-2013	Makalah SEMHNS	
7.	24-07-2013	Bab IV & V	
8.	02-08-2013	Revisi Bab IV & V	

Malang, 13 Agustus 2013
Dosen Pembimbing I


Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Karanglo Km. 2 Malang

FORMULIR BIMBINGAN SKRIPSI

Nama : Ahmad Muzakky Syarif
NIM : 0918190
Masa Bimbingan : 11 Mei 2013 s/d 11 November 2013
Judul Skripsi : PERANCANGAN APLIKASI SPEECH RECOGNITION
SEBAGAI PERINTAH, MENGGUNAKAN MICROSOFT SAPI
UNTUK MENJALANKAN APLIKASI BERBASIS WINDOWS
DENGAN MENGGUNAKAN VISUAL BASIC 6.0

NO	TANGGAL	URAIAN	PARAF PEMBIMBING
1.	17-06-2013	Bab I, II & III	
2.	20-06-2013	Makalah SEMHAS	
3.	23-06-2013	Bab IV & V	
4.	26-06-2013	Revisi Bab IV & V	
5.			
6.			
7.			
8.			

Malang, 13 Agustus 2013
Dosen Pembimbing II

Febriana Santi Wahyuni, S.Kom, M.Kom
NIP.P. 1031000425



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Karanglo, KM 2 Malang


**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Ahmad Muzakky Syarif
NIM : 0918190
Masa Bimbingan : 11 Mei 2013 s/d 11 November 2013
Judul Skripsi : PERANCANGAN APLIKASI SPEECH RECOGNITION
SEBAGAI PERINTAH, MENGGUNAKAN MICROSOFT SAPI
UNTUK MENJALANKAN APLIKASI BERBASIS WINDOWS
DENGAN MENGGUNAKAN VISUAL BASIC 6.0

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :
Hari : Senin
Tanggal : 19 Agustus 2013
Nilai : 80.15 (A)

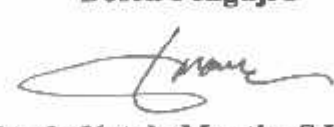
Panitia Ujian Skripsi :

Ketua Majelis Penguji


Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002

Anggota Penguji :

Dosen Penguji I


Sandy Nataly Mantja, S.Kom
NIP.P. 1030800418

Dosen Penguji II


Nurlaily Vendyansyah, ST



FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata I Program Studi Teknik Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Ahmad Muzakky Syarif
NIM : 0918190
JURUSAN : Teknik Informatika S-1
JUDUL : PERANCANGAN APLIKASI SPEECH RECOGNITION SEBAGAI PERINTAH, MENGGUNAKAN MICROSOFT SAPI UNTUK MENJALANKAN APLIKASI BERBASIS WINDOWS DENGAN MENGGUNAKAN VISUAL BASIC 6.0

Tanggal	Penguji	Uraian	Paraf
19 Agustus 2013	I	- Hasil pengujian disimpulkan yang berkaitan dengan noise dsb. Gagal pun disertakan dikesimpulan.	
	II	- Demo ulang penggunaan komponen	

Dosen Penguji I

Sandy Nataly Mantja, S.Kom
NIP.P. 1030800418

Dosen Penguji II

Nurlaily Vendyansyah, ST

Dosen Pembimbing I

Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002

Dosen Pembimbing II

Febriana Santi Wahyuni, S.Kom, M.Kom
NIP.P. 1031000425

LEMBAR PENGUJIAN APLIKASI

Perancangan Aplikasi Speech Recognition Sebagai Perintah,
Menggunakan Microsoft SAPI Untuk Menjalankan Aplikasi
Berbasis Windows Dengan Menggunakan Visual Basic 6.0

Nama Penguji : Abu Bakar

Keterangan Penguji :

.....

.....

- *Ceklist (✓) Sesuai point Prosedur pada kolom yang telah disediakan.*

No	Prosedur	Baik	Cukup	Kurang
1	Tampilan Aplikasi	✓		
2	Melafalkan Kata Kunci	✓		
3	Menambah Kata Kunci & File Aplikasi	✓		
4	Menu Editor	✓		
5	Eksekusi Aplikasi	✓		
Saran :				

Malang, Agustus 2013

Penguji,


Abu Bakar

LEMBAR PENGUJIAN APLIKASI

Perancangan Aplikasi Speech Recognition Sebagai Perintah,
Menggunakan Microsoft SAPI Untuk Menjalankan Aplikasi
Berbasis Windows Dengan Menggunakan Visual Basic 6.0

Nama Penguji : Anselmus Y. Yopi

Keterangan Penguji :

.....


.....

- *Ceklist (✓) Sesuai point Prosedur pada kolom yang telah disediakan.*

No	Prosedur	Baik	Cukup	Kurang
1	Tampilan Aplikasi	✓		
2	Melafalkan Kata Kunci	✓		
3	Menambah Kata Kunci & File Aplikasi	✓		
4	Menu Editor	✓		
5	Eksekusi Aplikasi	✓		
Saran :				

Malang, Agustus 2013

Penguji,


Anselmus Y. Yopi

LEMBAR PENGUJIAN APLIKASI

Perancangan Aplikasi Speech Recognition Sebagai Perintah,
Menggunakan Microsoft SAPI Untuk Menjalankan Aplikasi
Berbasis Windows Dengan Menggunakan Visual Basic 6.0

Nama Penguji : WIJAYA HADI KUSUMA

Keterangan Penguji :

.....

.....

- *Ceklist (✓) Sesuai point Prosedur pada kolom yang telah disediakan.*

No	Prosedur	Baik	Cukup	Kurang
1	Tampilan Aplikasi	✓		
2	Melafalkan Kata Kunci	✓		
3	Menambah Kata Kunci & File Aplikasi	✓		
4	Menu Editor	✓		
5	Eksekusi Aplikasi	✓		
Saran :				

Malang, Agustus 2013

Penguji,



WIJAYA H.K

LEMBAR PENGUJIAN APLIKASI

Perancangan Aplikasi Speech Recognition Sebagai Perintah,
Menggunakan Microsoft SAPI Untuk Menjalankan Aplikasi
Berbasis Windows Dengan Menggunakan Visual Basic 6.0

Nama Penguji : AHMAD ROZIGU ZAIN

Keterangan Penguji :

.....

.....

- Ceklist (✓) Sesuai point Prosedur pada kolom yang telah disediakan.

No	Prosedur	Baik	Cukup	Kurang
1	Tampilan Aplikasi	✓		
2	Melafalkan Kata Kunci	✓		
3	Menambah Kata Kunci & File Aplikasi	✓		
4	Menu Editor	✓		
5	Eksekusi Aplikasi	✓		
Saran :				

Malang, 14 Agustus 2013

Penguji,

AHMAD ROZIGU ZAIN

LEMBAR PENGUJIAN APLIKASI

Perancangan Aplikasi Speech Recognition Sebagai Perintah,
Menggunakan Microsoft SAPI Untuk Menjalankan Aplikasi
Berbasis Windows Dengan Menggunakan Visual Basic 6.0

Nama Penguji : Nur Muhammad Al Gams

Keterangan Penguji :

.....

.....

- Ceklist (✓) Sesuai point Prosedur pada kolom yang telah disediakan.

No	Prosedur	Baik	Cukup	Kurang
1	Tampilan Aplikasi	✓		
2	Melafalkan Kata Kunci	✓		
3	Menambah Kata Kunci & File Aplikasi	✓		
4	Menu Editor	✓		
5	Eksekusi Aplikasi	✓		
Saran :				

Malang, Agustus 2013

Penguji,



NUR M. GAMS

LEMBAR PENGUJIAN APLIKASI

Perancangan Aplikasi Speech Recognition Sebagai Perintah,
Menggunakan Microsoft SAPI Untuk Menjalankan Aplikasi
Berbasis Windows Dengan Menggunakan Visual Basic 6.0

Nama Penguji : GALIH IBNU LABAL

Keterangan Penguji :

.....

.....

- Ceklist (✓) Sesuai point Prosedur pada kolom yang telah disediakan.

No	Prosedur	Baik	Cukup	Kurang
1	Tampilan Aplikasi	✓		
2	Melafalkan Kata Kunci	✓		
3	Menambah Kata Kunci & File Aplikasi	✓		
4	Menu Editor	✓		
5	Eksekusi Aplikasi	✓		
Saran :				

Malang, 12 Agustus 2013

Penguji,


GALIH IBNU LABAL

LEMBAR PENGUJIAN APLIKASI

Perancangan Aplikasi Speech Recognition Sebagai Perintah,
Menggunakan Microsoft SAPI Untuk Menjalankan Aplikasi
Berbasis Windows Dengan Menggunakan Visual Basic 6.0

Nama Penguji : NYILO ENGGAP WARDANA

Keterangan Penguji : Bagus untuk inspirasi

- Ceklist (✓) Sesuai point Prosedur pada kolom yang telah disediakan.

No	Prosedur	Baik	Cukup	Kurang
1	Tampilan Aplikasi	✓		
2	Melafalkan Kata Kunci	✓		
3	Menambah Kata Kunci & File Aplikasi	✓		
4	Menu Editor	✓		
5	Eksekusi Aplikasi	✓		
Saran : <u>Tampilan Lebih dikembangkan</u>				

Malang, Agustus 2013

Penguji,



LEMBAR PENGUJIAN APLIKASI

Perancangan Aplikasi Speech Recognition Sebagai Perintah,
Menggunakan Microsoft SAPI Untuk Menjalankan Aplikasi
Berbasis Windows Dengan Menggunakan Visual Basic 6.0

Nama Penguji : Mudakir A.

Keterangan Penguji : Menarik

- Ceklist (✓) Sesuai point Prosedur pada kolom yang telah disediakan.

No	Prosedur	Baik	Cukup	Kurang
1	Tampilan Aplikasi	✓		
2	Melafalkan Kata Kunci	✓		
3	Menambah Kata Kunci & File Aplikasi	✓		
4	Menu Editor	✓		
5	Eksekusi Aplikasi	✓		
Saran : <u>Ditambah lebih banyak menu.</u>				

Malang, Agustus 2013

Penguji,


Mudakir A.

LEMBAR PENGUJIAN APLIKASI

Perancangan Aplikasi Speech Recognition Sebagai Perintah,
Menggunakan Microsoft SAPI Untuk Menjalankan Aplikasi
Berbasis Windows Dengan Menggunakan Visual Basic 6.0

Nama Penguji : Khafie Andi Rusali

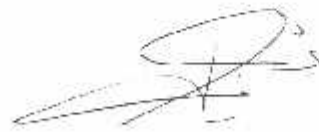
Keterangan Penguji : Salah satu referensi untuk tugas akhir saya

- Ceklist (✓) Sesuai point Prosedur pada kolom yang telah disediakan.

No	Prosedur	Baik	Cukup	Kurang
1	Tampilan Aplikasi	✓		
2	Melafalkan Kata Kunci	✓		
3	Menambah Kata Kunci & File Aplikasi	✓		
4	Menu Editor		✓	
5	Eksekusi Aplikasi		✓	
Saran : <u>Bisa ditambahkan bahasa Indonesia</u>				

Malang, Agustus 2013

Penguji,



LEMBAR PENGUJIAN APLIKASI

**Perancangan Aplikasi Speech Recognition Sebagai Perintah,
Menggunakan Microsoft SAPI Untuk Menjalankan Aplikasi
Berbasis Windows Dengan Menggunakan Visual Basic 6.0**

Nama Penguji : YUHANIS WALIAMA

Keterangan Penguji :

.....

.....

- *Ceklist (✓) Sesuai point Prosedur pada kolom yang telah disediakan.*

No	Prosedur	Baik	Cukup	Kurang
1	Tampilan Aplikasi	✓		
2	Melafalkan Kata Kunci	✓		
3	Menambah Kata Kunci & File Aplikasi	✓		
4	Menu Editor		✓	
5	Eksekusi Aplikasi			✓
Saran :				

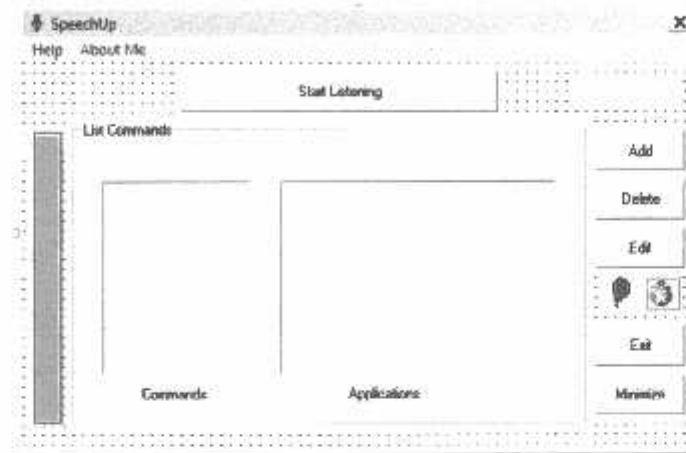
Malang, Agustus 2013

Penguji,



LISTING APLIKASI

1. Listing Halaman Utama



```
Option Explicit
Dim nid As NOTIFYICONDATA
'deklarasi
'manggil library windows user32
Dim Command As String, Description As String, Category As String, Flags As Long, Action
As String
Private Declare Function SetWindowPos Lib "user32" (ByVal hwnd As Long, ByVal
hwndInsertAfter As Long, ByVal X As Long, ByVal Y As Long, ByVal cx As Long, ByVal cy
As Long, ByVal wFlags As Long) As Long
Private Const SWP_NOMOVE = &H2
Private Const SWP_NOSIZE = &H1
Private Const HWND_TOPMOST = -1
Private Const HWND_NOTOPMOST = -2

'-----

' SetWindowPos Flags
Const SWP_NOZORDER = &H4
Const SWP_NOREDRAW = &H8
Const SWP_NOACTIVATE = &H10

'The frame changed: send WM_NCCALCSIZE
Const SWP_FRAMECHANGED = &H20

Const SWP_SHOWWINDOW = &H40
Const SWP_HIDEWINDOW = &H80
Const SWP_NOCOPYBITS = &H100

'Don't do owner Z ordering
```

```

Const SWP_NOOWNERZORDER = &H200
Const SWP_DRAWFRAME = SWP_FRAMECHANGED
Const SWP_NOREPOSITION = SWP_NOOWNERZORDER
' SetWindowPos() hwndInsertAfter values
Const HWND_TOP = 0
Const HWND_BOTTOM = 1

Private Sub Manual_Click()
Form3.Show
End Sub

Private Sub Tentangku_Click()
Form4.Show
End Sub

Private Sub Cmd_Start_Click()
If Cmd_Start.Caption = "Start Listening" Then
' mengaktifkan Direct SR1,Timer,
DirectSR1.Activate
Timer1.Enabled = True
Timer1.Interval = 2000
Cmd_Start.Caption = "Stop Listening" 'merubah tilisan caption
Else
' menonaktifkan Direct SR1,Timer,
DirectSR1.Deactivate
detek.Caption = ""
Cmd_Start.Caption = "Start Listening" 'merubah tilisan caption
End If
Form1.Command6.Enabled = True
End Sub

Private Sub Command4_Click()
minimize_to_tray
Form1.Command6.Enabled = False
End Sub

Private Sub Command5_Click()
Form1.List1.Clear
Form1.List2.Clear
Form_Load
End Sub

Private Sub Command6_Click()
Form6.Show
Form6.Text1.Text = List1.Text
Form6.Text2.Text = List2.Text

Dim index As Integer

```

```
index = List1.ListIndex
On Error Resume Next
Form1.List1.RemoveItem (index)
Form1.List2.RemoveItem (index)
Me.Hide
End Sub
```

```
Private Sub DirectSR1_VUMeter(ByVal beginhi As Long, ByVal beginlo As Long, ByVal  
level As Long)
```

```
    If VU_Meter.Max < level Then VU_Meter.Max = level  
    VU_Meter.Value = level
```

```
End Sub
```

```
Private Sub List1_Click()
```

```
    Dim noth
```

```
    List2.ListIndex = List1.ListIndex
```

```
    Command6.Enabled = True
```

```
End Sub
```

```
Private Sub List1_DbClick()
```

```
    MsgBox Form1.List2.List(List2.ListIndex)
```

```
End Sub
```

```
Private Sub List2_Click()
```

```
    List1.ListIndex = List2.ListIndex
```

```
End Sub
```

```
Private Sub List2_DbClick()
```

```
    MsgBox Form1.List2.List(List2.ListIndex)
```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
    Form1.List1.Clear
```

```
    Form1.List2.Clear
```

```
    Call loadfiles
```

```
    Form2.Show
```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
    End
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
    Dim index As Integer
```

```
    index = List1.ListIndex
```

```
    On Error Resume Next
```

```
    Form1.List1.RemoveItem (index)
```

```
    Form1.List2.RemoveItem (Index)
```

```
    Call savefiles
```

```
    Form1.List1.Clear
```

```
    Form1.List2.Clear
```

```
    Call loadfiles
```

```
    Form1.DirectSR1.Activate
```

```
    Form1.Command6.Enabled = False
```

```
End Sub
```

```

Private Sub Form_Load()
Dim junk, windir$
windir = Space(144)
junk = getwindir(windir, 144)
windir = Trim(windir)
i = InStr(windir$, vbNullChar)
windir$ = Mid$(windir$, 1, i - 1)
words = windir$ & "\words.txt"
dirs = windir$ & "\dirs.txt"
test = Dir(words)
If test = "" Then
Open words For Output As #1
Close #1
End If
test = Dir(dirs)
If test = "" Then
Open dirs For Output As #1
Close #1
End If
Call loadfiles
Command6.Enabled = False
Dim v
v = SetWindowPos(hwnd, HWND_TOPMOST, 0, 0, 0, 0, SWP_NOSIZE Or SWP_NOMOVE)
End Sub

```

```

Private Sub DirectSR1_PhraseFinish(ByVal Flags As Long, ByVal beginhi As Long, ByVal
beginlo As Long, ByVal endhi As Long, ByVal endlo As Long, ByVal Phrase As String, ByVal
parsed As String, ByVal results As Long)
Dim noth
List1.ListIndex = -1
For i = 0 To List1.ListCount
If Phrase = "" Then
List1.ListIndex = -1
List2.ListIndex = -1
Exit Sub
End If
' jika phraseFinish(yg diucapkan) = yg ad pada list, maka akan memanggil shell pada list2
(alamat *.exe)
If Phrase = List1.List(i) Then
List1.ListIndex = i
List2.ListIndex = i
noth = Shell(List2, vbNormalNoFocus)
End If
Next i
detek.Caption = Phrase
Timer1.Enabled = True
Timer1.Interval = 30000
End Sub

```

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Dim msg As Long
Dim sFilter As String
msg = X / Screen.TwipsPerPixelX
Select Case msg
Case WM_LBUTTONDOWN
Me.Show
Shell_NotifyIcon NIM_DELETE, nid
Case WM_LBUTTONUP
Case WM_LBUTTONDOWNBLCLK
Case WM_RBUTTONDOWN
Case WM_RBUTTONUP
Me.Show
Shell_NotifyIcon NIM_DELETE, nid
Case WM_RBUTTONDOWNBLCLK
End Select
End Sub
Private Sub Form_Unload(Cancel As Integer)
Shell_NotifyIcon NIM_DELETE, nid
End Sub

Sub minimize_to_tray()
Me.Hide
nid.cbSize = Len(nid)
nid.hwnd = Me.hwnd
nid.ulid = vbNull
nid.uFlags = NIF_ICON Or NIF_TIP Or NIF_MESSAGE
nid.uCallbackMessage = WM_MOUSEMOVE
nid.hIcon = Me.Icon
nid.szTip = "SpeechUp" & vbNullChar
Shell_NotifyIcon NIM_ADD, nid
End Sub

Private Sub menu_Click()

End Sub

Private Sub Me_Click()
Form4.Show
End Sub

Private Sub MicrosoftSAPI_Click()
Form5.Show
End Sub

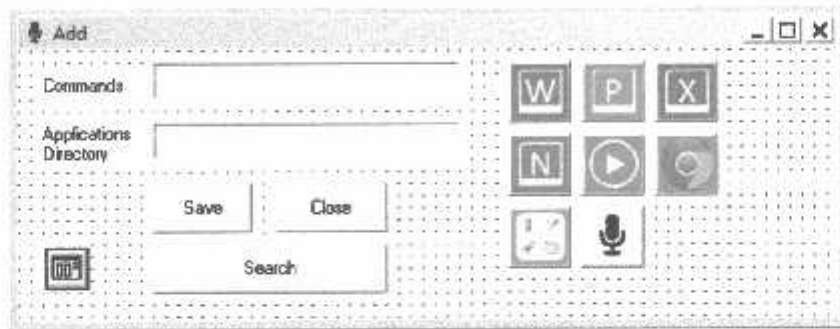
Private Sub Panduan_Click()
Form3.Show
End Sub
```

```

Private Sub Timer1_Timer()
detek.Caption = "<Nothing>"
End Sub

```

2. Listing Halaman Add



Option Explicit

```

Private Declare Function SetWindowPos Lib "user32" (ByVal hwnd As Long, ByVal
hwndInsertAfter As Long, ByVal X As Long, ByVal Y As Long, ByVal cx As Long, ByVal cy
As Long, ByVal wFlags As Long) As Long
Private Const SWP_NOMOVE = &H2
Private Const SWP_NOSIZE = &H1
Private Const HWND_TOPMOST = -1
Private Const HWND_NOTOPMOST = -2

```

```

Private Sub Command10_Click()
Text2.Text = "C:\Program Files\Microsoft Games\Chess\Chess.exe"
End Sub

```

```

Private Sub Command11_Click()
Text2.Text = "C:\Program Files\Microsoft Visual Studio\VB98\VB6.EXE"
End Sub

```

```

Private Sub Command4_Click()
Text2.Text = "C:\Program Files\Microsoft Office\Office12\WINWORD.exe"

```

End Sub

```

Private Sub Command5_Click()
Text2.Text = "C:\Program Files\Microsoft Office\Office12\POWERPNT.exe"
End Sub

```

```

Private Sub Command6_Click()
Text2.Text = "C:\Program Files\Microsoft Office\Office12\EXCEL.exe"
End Sub

```

```

Private Sub Command7_Click()
Text2.Text = "C:\Windows\system32\notepad.exe"
End Sub

Private Sub Command8_Click()
Text2.Text = "C:\Program Files\Windows Media Player\wmplayer.exe"
End Sub

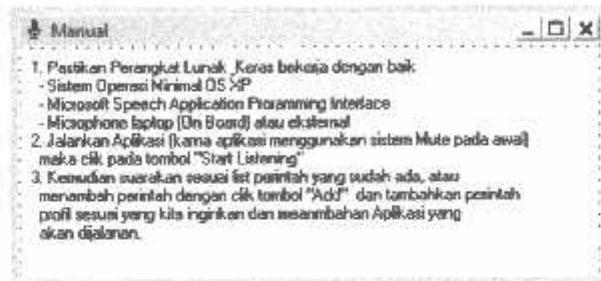
Private Sub Command9_Click()
Text2.Text = "C:\Users\AzKya\AppData\Local\Google\Chrome\Application\chrome.exe"
End Sub

Private Sub Form_Activate()
Text1.SetFocus
End Sub
Private Sub Command1_Click()
'tombol save
Dim count, pos, i, length As Integer
Dim line, test As Variant
If Text1.Text = "" Then
MsgBox "Masukan perintah"
Exit Sub
End If
If Text2.Text = "" Then
MsgBox "Masukan file yang dieksekusi"
Exit Sub
End If
Form1.List1.AddItem (Text1.Text)
Form1.List2.AddItem (Text2.Text)
Call savefiles
Form1.List1.Clear
Form1.List2.Clear
Call loadfiles
Text1.Text = ""
Text2.Text = ""
Text1.SetFocus
Unload Me
Form1.Command6.Enabled = False
End Sub
Private Sub Command2_Click()
'tombol close
Unload Me
Form1.Show
Form1.Command6.Enabled = False
End Sub
Private Sub Command3_Click()
'tombol search
On Error Resume Next
CommonDialog1.Filter = "Executable Files (*.exe)|*.EXE"

```

```
CommonDialog1.ShowOpen
If Err.Number = cdlCancel Then Exit Sub
Text2 = CommonDialog1.FileName
Form1.Command6.Enabled = False
End Sub
```

3. Listing Halaman Manual



```
Private Sub Panduan_Click()
Form3.Show
End Sub
```

4. Listing Halaman About Me



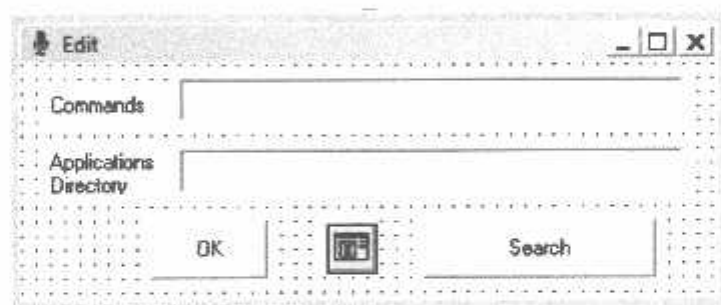
```
Private Sub Me_Click()
Form4.Show
End Sub
```

5. Listing Halaman Microsoft SAPI



```
Private Sub MicrosoftSAPI_Click()  
Form5.Show  
End Sub
```

6. Listing Halaman Edit



```
Private Sub Command1_Click()  
'control edit  
Dim count, pos, l, length As Integer  
Dim line, test As Variant  
If Text1.Text = "" Then  
Form1.Show  
Unload Me  
Exit Sub  
End If  
If Text2.Text = "" Then  
Form1.Show  
Unload Me  
Exit Sub  
End If  
Form1.List1.AddItem (Text1.Text)  
Form1.List2.AddItem (Text2.Text)  
Call savefiles
```

```

Form1.List1.Clear
Form1.List2.Clear
Call loadfiles
Text1.Text = ""
Text2.Text = ""
Text1.SetFocus
Form1.Show
Unload Me
Form1.Command6.Enabled = False
End Sub

```

```

Private Sub Command3_Click()
On Error Resume Next
CommonDialog1.Filter = "Executable Files (*.exe)|*.EXE"
CommonDialog1.ShowOpen
If Err.Number = cdICancel Then Exit Sub
Text2 = CommonDialog1.FileName
Form1.Command6.Enabled = False
End Sub

```

7. Fungsi Minimize

```

Public Type NOTIFYICONDATA
cbSize As Long
hwnd As Long
uid As Long
uFlags As Long
uCallbackMessage As Long
hIcon As Long
szTip As String * 64
End Type
Public Const NIM_ADD = &H0
Public Const NIM_MODIFY = &H1
Public Const NIM_DELETE = &H2
Public Const WM_MOUSEMOVE = &H200
Public Const NIF_MESSAGE = &H1
Public Const NIF_ICON = &H2
Public Const NIF_TIP = &H4
Public Const WM_LBUTTONDOWN = &H203
Public Const WM_LBUTTONUP = &H202
Public Const WM_RBUTTONDOWN = &H204
Public Const WM_RBUTTONUP = &H205
Public Declare Function Shell_NotifyIcon Lib "shell32" Alias "Shell_NotifyIconA" (ByVal dwMessage As Long, pnid As NOTIFYICONDATA) As Boolean

```