

**IMPLEMENTASI SPEECH RECOGNITION PADA KEAMANAN FOLDER
DENGAN SPEECH APPLICATION PROGRAMMING INTERFACE**

(SPEECH -API)

SKRIPSI

SYARIF HUSAIN HASIBUAN

NIM. 101402070



**PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA**

2014

**IMPLEMENTASI SPEECH RECOGNITION PADA KEAMANAN FOLDER
DENGAN SPEECH APPLICATION PROGRAMMING INTERFACE**

(SPEECH -API)

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh izajah sarjana
Teknologi Informasi

SYARIF HUSAIN HASIBUAN

NIM. 101402070



**PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA**

2014

PERSETUJUAN

Judul : IMPLEMENTASI SPEECH RECOGNITION PADA
KEAMANAN FOLDER DENGAN SPEECH
APPLICATION PROGRAMMING INTERFACE
(SPEECH – API)

Kategori : SKRIPSI

Nama : SYARIF HUSAIN HASIBUAN

Nomor Induk Mahasiswa : 101402070

Program Studi : S1 TEKNOLOGI INFORMASI

Departemen : TEKNOLOGI INFORMASI

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI
INFORMASI

UNIVERSITAS SUMATERA UTARA

Komisi Pembimbing :

Pembimbing 2

Pembimbing 1

Romi Fadillah Rahmat B.Comp. Sc., M.Sc
NIP. 19860303 201012 1 004

Baihaqi Siregar S.Si., M.T
NIP. 19790108 201202 1 002

Diketahui/disetujui oleh
Program Studi S1 Teknologi Informasi
Ketua,

Muhammad Anggia Muchtar, ST., MM.IT
NIP. 19800110 200801 1 010

PERNYATAAN**IMPLEMENTASI SPEECH RECOGNITION PADA KEAMANAN FOLDER
DENGAN SPEECH APPLICATION PROGRAMMING INTERFACE
(SPEECH-API)****SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, Mei 2015

SYARIF HUSAIN HASIBUAN

101402070

UCAPAN TERIMA KASIH



Alhamdulillah, puji syukur kita hantarkan kehadiran Allah SWT, yang telah memberikan Rahmat, kesehatan dan kesempatan sehingga ananda dapat menyelesaikan penulisan skripsi ini hingga selesai. Dan untuk yang paling Teristimewah kepada Ayahanda Drs. H. Hasyim Hasibuan dan Ibunda Dra. Hj. Asnah Siregar, serta keluarga yang memberikan dukungan tak henti-hentinya kepada penulis, sehingga penulis selalu bersemangat untuk menyelesaikan skripsi ini, ucapan beribu terimakasih dari penulis tak akan cukup membalas semua yang telah kalian berikan kepada penulis. penulis tidak akan bisa membalas budi kalian sampai kapanpun, kasih sayang dan perhatian yang sangat luar biasa yang selalu kalian berikan kepada penulis. Dan tak akan bosan-bosannya ucapan terimakasih itu akan penulis persembahkan hanya untuk kalian yaitu Ayahanda dan Ibunda tercinta.

Skripsi ini berjudul : **Implementasi Speech Recognition pada Keamanan Folder Dengan Speech Application Programming Interface (SPEECH – API).**

Penelitian ini bertujuan untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknologi Informasi dan juga tentunya sebagai bahan masukan khususnya untuk penulis dan kepada siapa saja yang memerlukannya.

Penulis menyadari sepenuhnya bahwa hasil penelitian ini belum sempurna, terutama dalam penulisan bahasa dan materi, sehingga penulis tetap menerima kritik dan saran yang bersifat membangun.

Terlepas dari kekurangan yang ada, penulis tetap menyampaikan penghargaan dan terimakasih yang sebesar-besarnya kepada :

1. Penulis ingin mengucapkan terima kasih kepada Bapak Romi Fadillah Rahmat B. Comp. Sc., M.Sc selaku pembimbing pertama dan Bapak Baihaqi Siregar S. Si., M.T selaku pembimbing kedua yang telah membimbing penulis dalam penelitian serta penulisan skripsi ini. Tanpa inspirasi serta motivasi yang diberikan dari kedua pembimbing, tentunya penulis tidak akan dapat menyelesaikan skripsi ini. Penulis juga mengucapkan terima kasih kepada [Muhammad Anggia Muchtar ST., MM., IT] sebagai dosen pembimbing pertama dan [Dani Gunawan ST., MT] sebagai dosen pembimbing kedua yang

telah memberikan masukan serta kritik yang bermanfaat dalam penulisan skripsi ini. Ucapan terima kasih juga ditujukan kepada semua dosen serta semua pegawai pada program studi S1 Teknologi Informasi, yang telah membantu serta membimbing penulis selama proses perkuliahan.

2. Terima kasih juga penulis ucapkan kepada seluruh teman-teman angkatan 2010 yang telah bersama-sama penulis melewati perkuliahan pada program studi S1 Teknologi Informasi terutama teman-teman dekat penulis yang telah berbagi ilmu dan memberikan motivasi sehingga penulis dapat menyelesaikan skripsi ini.
3. Terima kasih juga penulis ucapkan kepada seluruh anak-anak kantin PSI baik angkatan 2008 sampai 2013 yang telah bersama-sama penulis melewati perkuliahan pada program studi S1 Teknologi Informasi terutama teman-teman dekat penulis yaitu kak Ika selaku penjaga kantin sekaligus kakak bagi semua anak kantin, (2008) Zainul F., (2009) Fadli R, Fernando A, Alex W, M. Kurniawan, Dezi, Rudyanto, Mahatir F, Dedek A., dkk. (2010) Defri Ajung, Teddy A, Thomas S, Aser H, Fandy S., Dkk (2011) Noel A, dkk. (2013) Sammy, Bedi, Joseph, william, jodi., Dkk. yang telah bersama-sama memberikan motivasi serta semangat bagi penulis.
4. Terima kasih juga penulis ucapkan kepada seluruh Staff dan Pegawai Tata Usaha yaitu Ibu Delima, kak Naumi, abang Faisal Hamid dan Manaf yang telah banyak membantu dalam menyelesaikan studi di Program studi Teknologi Informasi.
5. Terima kasih juga penulis ucapkan kepada seorang terkasih yang sangat spesial dan berarti buat penulis yaitu Kartika Sari Dewi A.P., S. Sos yang telah membantu, memotivasi dan mendukung serta meluangkan waktu sehingga penulis bisa menyelesaikan penulisan ini.

ABSTRAK

Seiring dengan kemajuan teknologi, cara komunikasi antara manusia dan komputer telah mengalami perubahan dari waktu ke waktu. Perubahan tersebut dengan tujuan supaya komunikasi antara manusia dan komputer dapat berlangsung secara lebih efisien. Metode komunikasi yang cukup reliable dan user friendly yang masih digunakan sampai sekarang adalah GUI (Graphical User Interface). Metode baru yang baru diperkenalkan untuk menggantikan GUI adalah VUI (VoiceUser Interface). Perkembangan VUI ini ditunjang oleh teknologi ASR (Automatic Speech Recognizer). Dengan adanya teknologi ASR tersebut memungkinkan komputer untuk mengenali suara manusia sesuai dengan bahasa yang digunakan. Microsoft menyertakan teknologi ASR tersebut pada sistem operasi yang dibuatnya. Oleh Microsoft, teknologi ASR digabungkan dengan teknologi Speech Synthesis, yaitu teknologi yang dapat mensintesis suara manusia, menjadi suatu API (Application Programming Interface) yang disebut Microsoft Speech API. Pada penelitian ini Speech-API diimplemetasikan pada aplikasi pengamanan folder komputer dengan menggunakan bahasa pemrograman VB. NET 2010 dengan hasil yang baik.

Kata kunci: Speech Application Programming Interface (Speech - API), keamanan folder.

**IMPLEMENTATION OF SPEECH RECOGNITION IN SECURITY FOLDER
WITH SPEECH APPLICATION PROGRAMMING INTERFACE
(SPEECH -API)**

ABSTRACT

Along with advances in technology, means of communication between humans and computers have changed from time to time. Such changes with the goal of communication between humans and computers can take place more efficiently. Methods of communication are quite reliable and user friendly which is still used today is a GUI (Graphical User Interface). The new method was introduced to replace the GUI is VUI (VoiceUser Interface). The development is supported by technology VUI ASR (Automatic Speech Recognizer). With the ASR technology that allows computers to recognize human voice according to the language used. Microsoft includes the ASR technology on operating systems made. By Microsoft, ASR technology combined with Speech Synthesis technology, a technology that can synthesize human voice, became an API (Application Programming Interface) which is called the Microsoft Speech API. In this study Speech-API implemented in a computer folder security application using VB programming language. .NET 2010 with good results.

Keywords: Speech Application Programming Interface (Speech - API), the security folder.

DAFTAR ISI

	Hal.
Persetujuan	i
Pernyataan	ii
Ucapan Terima Kasih	iii
Abstrak	v
Abstract	vi
Daftar Isi	vii
Daftar Gambar	x
Daftar Tabel	xii
 BAB 1 Pendahuluan	
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah	2
 1.4. Tujuan Penelitian	2
1.5. Manfaat Penelitian	2
1.6. Metode Penelitian	3
1.7. Sistematika Penulisan	4
 BAB 2 Landasan Teori	
2.1. Suara	5
2.2. Representasi Suara	8
2.3. <i>Speech Recognition</i>	9

2.4. Microsoft <i>Speech Recognition</i>	10
2.4.1. Kemampuan Sistem <i>Speech Recognition Library</i>	10
2.4.2. Ruang Lingkup Sistem <i>Speech Recognition Library</i>	11
2.4.3. Arsitektur <i>Speech Application Programming Interface (S-API)</i>	12
2.4.4. Application Programming Interface (API)	12
2.4.5. <i>Device Driver Interface (DDI)</i>	13
2.4.6. Menjalankan Microsoft <i>Speech Recognition</i>	15
2.5. Penelitian Terdahulu	16

BAB 3 Analisis dan Perancangan

3.1. Analisis	23
3.1.1. Analisis Masalah	23
3.1.2. Usulan Pemecahan Masalah	23
3.1.3. <i>Speech Application Programming Interface (Speech-API)</i>	24
3.1.4. Analisis Kebutuhan Perangkat Lunak	24
3.1.5. Analisis Kebutuhan Perangkat Keras	24
3.2. Perancangan Sistem	25
3.2.1. General Arsitektur	25
3.2.2. Flow Chart Sistem	27
3.2.3. Flow Chart Proteksi Folder	28
3.2.4. Perancangan Sistem	29
3.2.4.1. Rancangan Antar Muka	29
3.2.4.2. Rancangan Menu Utama	30
3.2.4.3. Rancangan <i>Speech</i>	31
3.2.4.4. Rancangan Data User	31
3.2.4.5. Rancangan Login	32

3.2.4.6. Rancangan Browse	33
3.2.4.7. Rancangan Listing Proses	33
 BAB 4 Implementasi dan Pengujian	 34
4.1. Implementasi	34
4.1.1. Tampilan Menu Utama	34
4.1.2. Tampilan Browse	35
4.1.3. Tampilan Speech Lock Folder	36
4.1.4. Tampilan Speech Unlock Folder	38
4.1.5. Tampilan Data User	39
4.1.6. Tampilan Login	39
4.1.7. Tampilan Listing	40
4.2. Hasil Pengujian	40
 BAB 5 Kesimpulan dan Saran	
5.1. Kesimpulan	42
5.2. Saran	42
 Daftar Pustaka	 43
Lampiran : <i>Source Code</i>	

DAFTAR GAMBAR

	Hal.
Gambar 2.1. Alur Gelombang Suara	5
Gambar 2.2. Gelombang Suara	6
Gambar 2.3. Proses Sampling Audio Analog ke Digital	8
Gambar 2.4. Blok Diagram Arsitektur Speech - API	12
Gambar 2.5. Tampilan Setup Microsoft Speech Recognition	15
Gambar 2.6. Tampilan Speech Recognition	16
Gambar 3.1. General Arsitektur	25
Gambar 3.2. Flow Chart Sistem	27
Gambar 3.3. Flow Chart Proteksi Folder	28
Gambar 3.4. Rancangan Menu Utama	30
Gambar 3.5. Rancangan Speech	31
Gambar 3.6. Rancangan User	32
Gambar 3.7. Rancangan Login	32
Gambar 3.8. Rancangan Browse	33
Gambar 3.9. Rancangan Listing Proses	33
Gambar 4.1. Tampilan Menu Utama	34
Gambar 4.2. Tampilan Browse	35
Gambar 4.3. Pemilihan Folder Yang Akan Diproteksi	35
Gambar 4.4. Tampilan Browse	36
Gambar 4.5. Tampilan Konfirmasi Pemilihan Folder	36
Gambar 4.6. Tampilan Speech	37
Gambar 4.7. Tampilan Speech Dalam Memproteksi Folder	37
Gambar 4.8. Tampilan Hasil Proteksi Folder	38

Gambar 4.9. Pemilihan Folder Yang Akan di-Unlock	38
Gambar 4.10. Tampilan Data User	39
Gambar 4.11. Tampilan Login	39
Gambar 4.12. Tampilan Listing	39

DAFTAR TABEL

	Hal
Tabel 4.1. Hasil Pengujian Proteksi Folder	40
Tabel 4.2. Hasil Pengujian Un-Proteksi Folder	41

BAB I

PENDAHULUAN

1.1.Latar Belakang

Dalam kehidupan sehari-hari komputer mempunyai peranan sangat penting dalam membantu mempermudah dan mempercepat pekerjaan. Pada zaman dahulu seseorang yang belum mempunyai komputer harus menggunakan mesin ketik untuk mengetik. Namun dengan komputer kita dapat mengeditnya berulang-ulang tanpa harus memboroskan kertas. Dan dokumen/hasil ketikan tersebut dapat disimpan dalam komputer. Seiring dengan perkembangan zaman, kebutuhan pengguna komputer akan keamanan suatu data yang disimpan dalam *file* menjadi salah satu kepentingan yang tidak bisa dianggap sebelah mata. Sebagai pengguna komputer seseorang berhak untuk menjaga kerahasiaan dan keamanan *file* yang dimiliki. Sebab sifat *file* dan dokumen penting sangat sensitif maka dari itu perlu adanya keamanan *file* dan dokumen-dokumen penting dengan menggunakan *password*. Tetapi tidak sedikit *password* menjadi masalah baru bagi pengguna komputer. Seperti mudah ditebak oleh orang lain dan lupa *password*. Pengenalan dan penggunaan suara bisa menjadi aplikasi dalam sistem *security* dalam menjaga *file* atau dokumen penting.

Pengamanan *file* atau dokumen penting menggunakan *password* masih bersifat konvensional. Karena itu dapat dilakukan pengamanan yang lebih detail untuk mengurangi tingkat kesalahan atau penyalagunaan *password* bagi pengguna komputer yaitu dengan mengidentifikasi manusia berdasarkan beberapa karakteristik fisiologis seseorang sebagai kode unik untuk melakukan pengenalan seperti DNA, telinga, sidik jari, geometri tangan, gaya berjalan, wajah, selaput jala, gigi, hentakan tombol, tanda tangan, dan suara. (Darma Putra, 2009).

Suara merupakan bunyi yang dikeluarkan dari mulut manusia seperti pada waktu berbicara, menyanyi, tertawa, dan menangis (KBBI, 2009). Hal tersebut yang menjadi pembeda suara seseorang karena bersifat unik. Selain itu suara tidak dapat

lupa, tidak mudah hilang, dan tidak mudah dipalsukan karena keberadaanya melekat pada diri manusia sehingga keunikannya lebih terjamin. (Syah, 2009).

1.2 Rumusan Masalah

Minimnya pengamanan dalam mengakses *folder* menjadikan penyalahgunaan dan pengambilan *file* rentan terjadi. Maka dari itu, diperlukan suatu pendekatan keamanan *folder* untuk mengenali identitas seseorang melalui suara yang tidak mudah dimanipulasi agar keamanan *folder* dapat terjaga.

1.3 Batasan Masalah

Batasan masalah dari penelitian ini adalah sebagai berikut :

1. Perekaman suara dilakukan dari Microphone.
2. Untuk perekaman suara *training* awal harus dilakukan dalam 3 kali pengambilan.
3. Objek penelitian adalah suara yang dikeluarkan.
4. Hasil berupa keakuratan pengenalan suara untuk membuka *folder*.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah menerapkan *speech recognition* dengan menggunakan teknik S-API (*Speech Application Programming Interface*) di dalam keamanan *folder* guna menjaga keamanan dalam pengaksesan *folder*.

1.5 Manfaat Penelitian

Penulis berkeinginan agar penelitian ini dapat memberikan manfaat kepada penulis dan para pembaca. Manfaat dari penelitian ini adalah sebagai berikut:

1. Membuka keamanan *folder* dengan menggunakan *voice recognition* agar menjaga keamanan dalam pengaksesan *folder*.
2. Menambah pengetahuan baru mengenai teknik S-API (*Speech Application Programming Interface*).
3. Penelitian dapat menjadi bahan rujukan untuk pengembangan penelitian lebih lanjut.

1.6 Metode Penelitian

Tahapan – tahapan yang akan dilakukan pada penulisan skripsi ini adalah sebagai berikut:

1. Studi Literatur

Studi literature dilakukan dengan cara mengumpulkan bahan referensi yaitu dari buku, paper, artikel, makalah, jurnal, dan situs – situs dari internet. Studi literature yang dilakukan berkaitan dengan *Voice Recognition*, Keamanan *folder*.

2 Identifikasi Masalah

Pada tahapan ini dilakukan identifikasi masalah yang akan diselesaikan pada aplikasi ini.

3 Analisis

Pada tahap ini dilakukan analisis terhadap hasil studi literatur untuk mengetahui dan memahami mengenai penerapan keamanan pada folder dengan *voice recognition*.

4 Perancangan Sistem

Pada tahap ini dilakukan perancangan arsitektur, perancangan penyimpanan suara dari sistem yang akan dikembangkan.

5 Implementasi Sistem

Pada tahap ini dilakukan proses implementasi pengkodean program dalam aplikasi komputer menggunakan bahasa pemrograman yang telah dipilih sesuai dengan analisis dan perancangan yang sudah dilakukan.

6 Pengujian

Pada tahap ini dilakukan proses dan percobaan terhadap sistem sesuai dengan kebutuhan yang ditentukan sebelumnya serta memastikan program dibuat berjalan seperti yang diharapkan.

7 Dokumentasi

Pada tahap ini dilakukan pembuatan dokumentasi sistem, lengkap dengan analisis yang diperoleh, hasil analisis dan implementasi dari aplikasi yang dibangun dapat dihasilkan sesuai yang diharapkan.

1.7. Sistematika Penulisan

Penulisan skripsi ini terdiri dari lima bab dengan masing – masing bab akan dijelaskan secara singkat sebagai berikut :

Bab 1 : Pendahuluan

Pada bab ini berisikan latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

Bab 2 : Landasan Teori

Pada bab ini dibahas mengenai teknik yang digunakan untuk penelitian skripsi teknik *SAPI (Speech Application Programming Interface)*, disini dijelaskan bagaimana teknik tersebut digabungkan dengan keamanan data (*folder*).

Bab 3 : Analisis dan Perancangan Sistem

Pada bab ini berisikan paparan analisis terhadap permasalahan dan penyelesaian persoalan terhadap penerapan *voice recognition* pada Keamanan *Folder* menggunakan *SAPI (Speech Application Programming Interface)*.

Bab 4 : Implementasi dan Pengujian Sistem

Pada bab ini berisi implementasi perancangan sistem dari hasil analisis dan perancangan yang sudah dibuat, serta menguji sistem untuk menemukan kelebihan dan kekurangan pada penerapan *voice recognition* pada Keamanan *Folder* menggunakan *SAPI (Speech Application Programming Interface)*

Bab 5 : Kesimpulan dan Saran

Pada bab ini berisikan kesimpulan yang didapatkan terhadap hasil penelitian skripsi dan saran untuk pengembangan lebih lanjut tentang topik terkait yang dibahas pada skripsi.

BAB 2

LANDASAN TEORI

2.1 Suara

Audio (suara) adalah fenomena fisik yang dihasilkan oleh getaran suatu benda yang berupa sinyal analog dengan amplitudo yang berubah secara kontinyu terhadap waktu yang disebut frekuensi (Binanto, 2010).

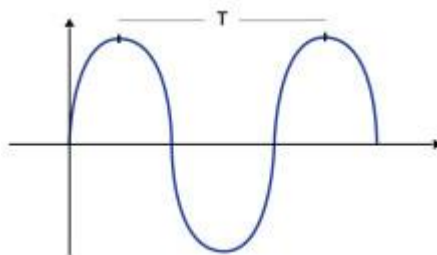


Gambar 2.1 Alur Gelombang Suara (Binanto, 2010)

Selama bergetar, perbedaan tekanan terjadi di udara sekitarnya. Pola osilasi yang terjadi dinamakan sebagai gelombang. Gelombang mempunyai pola sama yang berulang pada interval tertentu, yang disebut sebagai periode. Contoh suara periodik adalah instrumen musik, nyanyian burung sedangkan contoh suara non periodik adalah batuk, percikan ombak dan lain-lain.

Gelombang suara terjadi sebagai variasi tekanan dalam sebuah media, seperti udara. Ia tercipta dari bergetarnya sebuah benda, yang menyebabkan udara sekitarnya ikut bergetar. Udara yang bergetar kemudian diterima oleh telinga, menyebabkan gendang telinga manusia bergetar, kemudian otak menafsirkannya sebagai suara. Gelombang suara berjalan melalui udara, sama seperti gelombang yang terjadi di air. Bahkan, gelombang air lebih mudah untuk dilihat dan dimengerti, hal ini sering digunakan sebagai analogi untuk menggambarkan bagaimana gelombang suara

berperilaku. Gelombang suara juga dapat ditampilkan dalam grafik XY (Binanto, 2010).



Gambar 2.2 Gelombang Suara (Binanto, 2010)

Pada Gambar 2.2 di atas, suatu grafik gelombang berbentuk dua dimensi, tetapi di dunia nyata gelombang suara berbentuk tiga dimensi. Grafik menunjukkan gelombang bergerak sepanjang jalan dari kiri ke kanan, tapi kenyataannya perjalanan gelombang suara bergerak ke segala arah menjauhi sumber. Kira-kira sama seperti riak air yang terjadi ketika kita menjatuhkan sebuah batu ke dalam kolam. Namun model dua dimensi ini, cukup dapat menjelaskan tentang bagaimana suara bergerak dari satu tempat ke tempat lain.

Bagian-bagian Gelombang Suara

Semua gelombang memiliki sifat-sifat tertentu. Ada tiga bagian yang paling penting untuk audio (Binanto, 2010) :

Panjang gelombang: Jarak antara titik manapun pada gelombang (pada gambar ditunjukkan sebagai titik tertinggi) dan titik setara pada fase berikutnya. Secara harfiah, panjang gelombang adalah jarak yang digambarkan dgn huruf “T”.

Amplitudo: atau kekuatan sinyal gelombang (*intensity*). Titik tertinggi dari gelombang bila dilihat pada grafik. Amplitudo tinggi biasa disebut sebagai volume yang lebih tinggi, diukur dalam dB. Nama perangkat untuk meningkatkan amplitudo disebut amplifier.

Frequency: Frekuensi waktu yang dibutuhkan oleh gelombang bergerak dari satu pase ke pase berikutnya dalam satu detik. Diukur dalam *herz* atau *cycles per second*. Semakin cepat sumber suara bergetar, semakin tinggi frekuensi.

Frekuensi yang lebih tinggi ditafsirkan sebagai *pitch* yang lebih tinggi. Sebagai contoh, ketika seseorang menyanyi dengan suara bernada tinggi dengan memaksakan pita suara bergetar lebih cepat.

Hal berikutnya yang perlu diperhatikan adalah apa artinya ketika gelombang mencapai titik tertinggi atau titik rendah. Pada sinyal elektronik, nilai tinggi menunjukkan tegangan positif yang tinggi. Ketika sinyal ini dikonversi menjadi gelombang suara, kita dapat membayangkan nilai-nilai tinggi tersebut sebagai daerah yang mewakili peningkatan tekanan udara. Ketika gelombang menyentuh titik tertinggi, hal ini berhubungan dengan molekul udara yang menyebar bersama-sama secara padat. Ketika gelombang menyentuh titik rendah, molekul udara menyebar lebih tipis (renggang).

Audio diproduksi oleh sebuah objek yang bergetar, contohnya loud speaker, *musical instrument*, ataupun pita suara manusia. Getaran mekanik dari sebuah loud speaker membuat pergerakan udara terdorong dan tertarik dari kondisi stabil, adanya gerakan mendorong dan menarik yang terus menerus dari sebuah speaker membuat tekanan udara berubah yang pada akhirnya menyebabkan terjadinya sebuah gelombang suara.

Sebuah gelombang suara dapat dideskripsikan oleh frekuensi dan amplitudo. Frekuensi 1 Hz berarti 1 cycle gelombang lengkap setiap satu detik. Satuan sebuah frekuensi adalah Hertz (Hz). Frekuensi *audible (human hearing rang)* adalah 20 Hz sampai 20000 Hz. Dalam kenyataan praktis sebuah sumber suara selalu diproduksi pada banyak frekuensi secara simultan. Amplitudo sebuah gelombang mengacu pada besarnya perubahan tekanan dan tingkat kerasnya (*loudness*) gelombang suara.

Sebuah sinyal suara diproduksi dan ditransmisikan melalui udara, akhirnya diterima pada telinga manusia. Telinga manusia memiliki gendang pendengaran (*eardrum*) yang dapat bergetar pada saat menerima gerakan gelombang udara (*push and pull*). Pengelompokan sound dapat dilakukan berdasarkan *acoustic behavior*-nya. Berdasarkan *acoustic behavior*-nya *sound* dibedakan menjadi dua jenis yaitu *direct sound* dan *indirect sound (ambient)*.

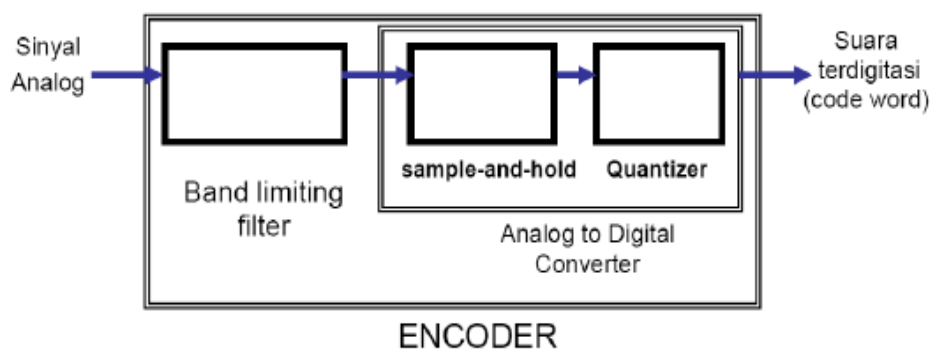
Dikatakan *direct sound* apabila sumber suara berjalan dari sumber suara langsung menuju ke pendengar dalam aliran garis lurus. *Indirect sound* bila sumber

suara dipantulkan terlebih dahulu pada satu atau lebih permukaan bidang sebelum sampai pada pendengar, karena adanya proses pemantulan sinyal suara pada *indirect sound* maka ditemukan adanya *delay time* untuk tiba kepada pendengar.

Terdapat beberapa macam *indirect sound*, tergantung pada *room acoustic*, sebagai contoh pengaturan pada *car audio* lebih susah apabila dibandingkan *room audio* karena bentuk ruang dan material yang sangat memungkinkan terjadinya banyak pantulan sumber suara sebelum sampai ke pendengar. Hal ini sangat menyulitkan penempatan sumber suara dalam mobil. *Echo* atau gema terjadi ketika sebuah *indirect sound* tertunda dalam waktu yang cukup lama untuk dapat didengar pendengar sebagai perulangan sinyal suara sebuah *direct sound*.

2.2 Representasi Suara

Gelombang suara analog tidak dapat langsung direpresentasikan pada komputer. Komputer mengukur amplitudo pada satuan waktu tertentu untuk menghasilkan sejumlah angka. Tiap satuan pengukuran ini dinamakan “*sample*”. *Analog to Digital Conversion* (ADC) adalah proses mengubah amplitudo gelombang bunyi ke dalam waktu interval tertentu (*sampling*), sehingga menghasilkan representasi digital dari suara. Dalam teknik *sampling* dikenal istilah *sampling rate* yaitu beberapa gelombang yang diambil dalam satu detik. Sebagai contoh jika kualitas CD Audio dikatakan memiliki frekuensi sebesar 44.100 Hz, berarti jumlah *sampel* sebesar 44.100 per detik. Proses *Sampling Audio Analog ke Digital* dapat dilihat seperti pada Gambar 2.3.



Gambar 2.3 Proses *Sampling Audio Analog ke Digital* (Binanto, 2010).

Langkah-langkah dalam proses digitasi adalah:

1. Membuang frekuensi tinggi dari *source signal*.
2. Mengambil *sample* pada interval waktu tertentu (*sampling*).
3. Menyimpan amplitudo sampel dan mengubahnya ke dalam bentuk diskrit (kuantisasi).
4. Merubah bentuk menjadi nilai biner.

Teknik *sampling* yang umum pada file *audio* seperti *Nyquist Sampling Rate* dimana untuk memperoleh representasi akurat dari suatu sinyal analog secara *lossless*, amplitudonya harus diambil *sample*-nya setidaknya pada kecepatan (*rate*) sama atau lebih besar dari 2 kali lipat komponen frekuensi maksimum yang akan didengar. Misalnya untuk sinyal analog dengan *bandwith* 15Hz – 10kHz \rightarrow *sampling rate* = 2 x 10KHz = 20 kHz (Gunawan, 2005).

2.3 Speech Recognition

Speech Recognition adalah sebuah alternatif metode tradisional untuk berinteraksi dengan komputer, seperti input teks melalui *keyboard*. Teknik input data menggunakan *speech recognition* dapat mengganti masukan dari *keyboard* maupun melalui *mouse*. *Speech Recognition* sangat membantu jika seseorang dalam situasi sebagai berikut:

1. Orang yang memiliki sedikit keterampilan dalam menggunakan *keyboard*, dalam arti lambat dalam melakukan pengetikan.
2. Orang yang memiliki masalah dengan karakter atau penggunaan kata dan manipulasi dalam bentuk tekstual.
3. Orang yang memiliki cacat fisik yang tidak mampu untuk melakukan entri data melalui *keyboard* maupun *mouse*, atau memiliki ketidakmampuan melihat apa yang mereka inputkan. (Kirriemuir, 2003).

2.4 Microsoft *Speech Recognition*

Microsoft *Speech Recognition* merupakan perangkat yang dikeluarkan oleh Microsoft Corporation yang memungkinkan kita melakukan perintah pengetikan ke dalam *Personal Computer*(PC) yang beroperasi di bawah sistem operasi Windows XP maupun Windows 7. Suara yang kita masukkan akan secara otomatis dikonversikan ke dalam bentuk teks. (Syarif et al, 2011).

Dengan menggunakan *speech recognition* memungkinkan kita untuk mengurangi kepenatan pada jari-jari dan pergelangan tangan dan meningkatkan kecepatan dan efisiensi dalam melakukan suatu pekerjaan pengetikan. (University *Information Technology Training&Documentation Department*, 2011).

2.4.1 Kemampuan Sistem *Speech Recognition Library*

Kemampuan sistem *speech recognition library* antara lain:

1. Informatif, sistem mampu menghasilkan informasi yang akurat. Pemakai hanya membutuhkan informasi yang dihasilkan oleh sistem dengan memberikan beberapa interaksi. Informasi yang dihasilkan oleh sesuai dengan kebutuhan pemakai.
2. *User Friendly*, sistem mudah digunakan karena menggunakan bahasa yang umum dipakai oleh user dan mudah dimengerti sehingga user mudah berinteraksi dengan sistem.
3. *Up to date*, informasi yang dihasilkan oleh sistem dapat dengan mudah diperbarui sehingga informasi yang dihasilkan sesuai dengan keadaan yang sebenarnya.

2.4.2 Ruang Lingkup Sistem *Speech Recognition Library*

Ruang lingkup Sistem *Speech Recognition Library* terdiri dari:

1. Pemakai, sistem ini nantinya dipakai oleh Basis data Administrator, serta pengguna lain yang berhubungan dengan permintaan data atau informasi dan akan mengikuti petunjuk yang ada pada perangkat lunak tersebut.
2. Elemen Sistem, menurut Pressman (2002:276) suatu sistem terdiri dari sejumlah elemen yang saling bekerja sama membentuk satu kesatuan.

Elemen-elemen tersebut antara lain:

a. Perangkat lunak.

Program komputer, struktur data, dan dokumen yang berhubungan yang berfungsi untuk mempengaruhi metode logis, prosedur, dan kontrol yang dibutuhkan. Dalam sistem ini menggunakan perangkat lunak antara lain:

1. Operating Sistem: Microsoft Windows 7.
2. Aplikasi, terdiri dari Microsoft Visual Basic versi 2008 yang digunakan untuk mendesain program, Microsoft Visio 2000 yang digunakan untuk membuat diagram alur proses, *Microsoft Recognize* yang digunakan untuk menerima input data berupa suara.

b. Perangkat keras.

Perangkat elektronik yang memberikan kemampuan penghitungan, dan perangkat elektromekanik (misalnya, sensor, rotor, pompa) yang memberikan fungsi dunia eksternal. Dalam sistem ini menggunakan perangkat keras dengan spesifikasi, prosessor Intel Pentium Dual Core 2.6 Ghz, Hardisk 80GB, RAM 1 GB, Monitor SVGA.

c. Manusia.

Pemakai dan operator perangkat keras dan perangkat lunak.

d. Basis data.

Kumpulan informasi yang besar dan terorganisasi yang diakses melalui perangkat lunak. Dalam sistem ini menggunakan basis data dengan format MySQL.

e. Dokumentasi.

Manual, formulir, dan informasi deskriptif lainnya yang menggambarkan penggunaan dan atau pengoperasian sistem.

f. Prosedur.

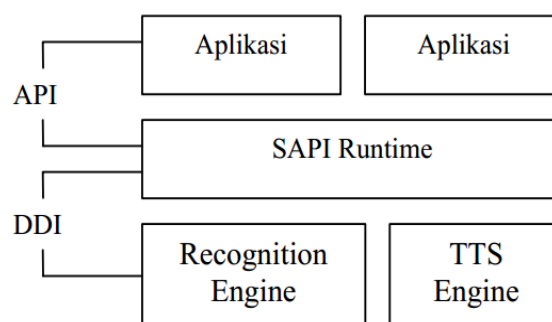
Langkah-langkah yang menentukan penggunaan khusus dari masing-masing elemen sistem atau konteks prosedural di mana sistem berada.

2.4.3 Arsitektur *Speech Application Programming Interface* (S-API 5.1)

S-API 5.1 terdiri dari 2 antar muka yaitu *application programming interface* (API) dan *device driver interface*(DDI).

2.4.4 *Application Programming Interface* (API)

Pada sistem pengenalan pembicaraan, aplikasi akan menerima even pada saat suara yang diterima telah dikenali oleh *engine* (Syarif et al, 2011).



Gambar 2.4 Blok Diagram Arsitektur Speech-API

Komponen SAPI yang akan menghasilkan even ini diimplementasikan oleh antarmuka `ISpNotifySource`. Lebih spesifik, Speech-API menggunakan

SetNotifySink, yaitu aplikasi akan meneruskan pointer IspNotify -Sink ke IspNotifySourceSetNotifySink. IspNotifySourceSetNotifySink ini akan menerima pemanggilan melalui IspNotifySink Notify ketika terdapat satu atau lebih even yang menyatakan bahwa aplikasi dapat mengambil data. Biasanya aplikasi tidak mengimplementasikan IspNotifySink secara langsung tetapi menggunakan Co Create 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 even yang ditimbulkan oleh notifikasi tersebut. Dengan menggunakan obyek IspEventSource, aplikasi dapat menerima informasi tentang even yang ditimbulkan oleh notifikasi. IspEventSource juga menyediakan mekanisme untuk menyaring dan membuat antrian even. Biasanya aplikasi tidak menerima notifikasi dari IspEventSource sampai terjadi pemanggilan terhadap IspEventSourceSetInterest untuk menentukan even mana yang akan menghasilkan notifikasi dan even mana yang berulang sehingga harus dimasukkan ke daftar antrian. Even diidentifikasi dengan menggunakan tanda SPEVENTENUM.

2.4.5 Device Driver Interface (DDI)

DDI menyediakan fungsi untuk menerima data suara dari Speech-API dan mengembalikan pengenalan frasa pada level Speech-API paling dasar (Noertjahyana, A. & Adipranata R. 2010). Terdapat dua antar muka yang digunakan oleh DDI yaitu IspSREngine, yang diimplementasikan oleh engine dan IspSREngineSite yang diimplementasikan oleh S-API. Antar muka Speech-API yaitu IspSREngineSite juga menyediakan metode untuk memberikan informasi lebih detail mengenai apa yang dikenali oleh engine. Grammars dan speakers menyediakan informasi ke engine yang dapat membantu engine untuk melakukan pengenalan pembicaraan lebih baik, disamping juga merupakan bagian penting komunikasi yang menghubungkan Speech-API dan *speech engine*. Terdapat 2 aspek terakhir yang berhubungan dengan komunikasi antara engine dan S-API yaitu urutan pemanggilan yang terjadi serta masalah *threading*. Salah satu keuntungan menggunakan Speech-API 5.1 ini adalah penyederhanaan masalah *threading*. Engine menyediakan layanan ke Speech-API melalui antar muka IspSREngine. Semua fungsi pengenalan terjadi melalui

ISpSREngineRecognizeStream. Ketika Speech-API memanggil ISpSREngineSetSite, maka Speech-API memberikan pointer ke antar muka ISpSREngineSite dimana kemudian engine dapat berkomunikasi dengan Speech-API selama ISpSREngineRecognizeStream dieksekusi. Speech-API membuat sebuah thread ke obyek ISpSREngine dan engine tidak boleh meninggalkan ISpSREngineRecognizeStream sampai terjadi kesalahan atau Speech-API sudah terindikasi dengan menggunakan ISpSREngineSiteRead, dimana tidak ada lagi data yang dapat diproses dan engine telah selesai melakukan tugasnya.

Speech-API memisahkan pembuat engine dari kerumitan untuk mengatur peralatan suarasecara detail. Speech-API menjaga logical stream dari raw audio data dengan membuat indeks posisi stream. Dengan menggunakan indeks posisi stream, engine dapat melakukan pemanggilan terhadap ISpSREngineSiteRead untuk menerima buffer dari raw audio data selama ISpSREngineRecognize Stream dieksekusi. Pemanggilan ini akan terjadi sampai semua data yang dibutuhkan tersedia. Jika ISpSREngineSiteRead menghasilkan data yang lebih sedikit dari yang dibutuhkan, yang berarti tidak ada data lagi, maka engine akan menghentikan eksekusi ISpSREngineRecognizeStream.

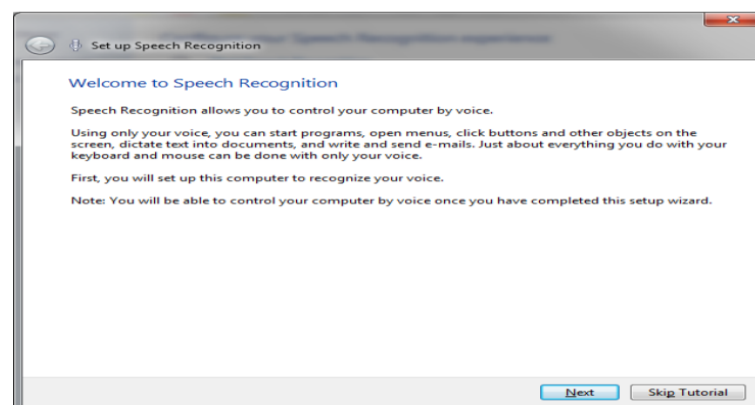
DDI memungkinkan engine untuk hanya mempunyai satu buah thread yang dieksekusi antara Speech-API dan engine. Satu-satunya metode yang tidak mengijinkan IspSREngine untuk masuk dan keluar secara cepat ialah ISpSREngineRecognizeStream. Engine akan selalu tereksekusi dan terpisah di dalam RecognizeStream sampai terjadi kegagalan atau tidak ada lagi data yang diterima dari ISpSREngineRead. Ketika engine mempunyai kesempatan untuk memberikan Speech-API giliran melakukan pemanggilan kembali ke ISpSREngine, maka engine harus memanggil IspSREngineSynchronize dan memberikan indeks posisi stream dimana engine telah selesai melakukan pengenalan data. Darimanapun Synchronize dipanggil, Speech-API dapat melakukan pemanggilan kembali kemetode ISpSREngine yang lain kecuali Recognize. Sebagai contoh, speaker dapat berubah, grammars dapat di-unload, diaktifkan ataupun dinonaktifkan secara dinamis. Pada bagian manapun, IspSREngine akan dipanggil hanya pada thread semula yang membuat ISpSREngine atau thread dimana engine memanggil ISpSREngineSiteSynchronize. Jika engine hanya memanggil

ISpSREngineSiteSynchronize pada *thread* yang sama dengan ISpSREngineRecognizeStream, maka hal ini berarti Speech-API hanya memanggil satu thread yang merupakan turunan dari ISpSREngine.

2.4.6 Menjalankan Microsoft *Speech Recognition*

Sebelum menjalankan aplikasi Microsoft *Speech Recognition*, terlebih dahulu mempersiapkan peralatan seperti headset dan *microphone*. Selanjutnya jalankan Microsoft *Speech Recognition*. Pertama-tama akan harus dilakukan tahap-tahap setup untuk menentukan seting awal aplikasi sebagai berikut:

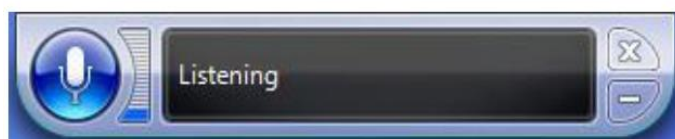
1. Siapkan *headset* dan *microphone*.
2. Selanjutnya hubungkan *microphone* ke dalam konektor komputer (biasanya berwarna merah jambu).
3. Kemudian klik Start > Control Panel > Speech Recognition.
4. Klik Start *Speech Recognition*.
5. Ikuti langkah-langkah di dalam wizard.



Gambar 2.5 Tampilan setup Microsoft *Speech Recognition*

Setelah selesai melakukan setup, tahap selanjutnya adalah sebagai berikut :

1. Pilih Start > Control Panel > Speech Recognition, akan tampil konfigurasi Speech Recognition.
2. Klik Train your computer to better understand you. Selanjutnya akan tampil *window voice training Speech Recognition*.
3. Klik *Next* dan akan muncul suara dari *voice training* untuk menuntun penggunaan *speech recognition*.
4. Selanjutnya ikuti petunjuk yang tampil di layar.
5. Jika selesai untuk menjalankan aplikasi *Speech Recognition* klik Start > Control Panel > Speech Recognition > *Start Speech Recognition*.
6. Selanjutnya akan muncul tampilan seperti gambar 2.6 berikut.



Gambar 2.6. Tampilan *Speech Recognition*

Jika muncul tampilan seperti di atas maka *speech recognition* sudah siap untuk menerima perintah dari masukan suara.

2.5 Penelitian Terdahulu

Pada penelitian Herdianto (2012) yang berjudul Penerapan *Speech Recognition Library* Untuk *Query* Basis Data. Pada kasus lain seringkali manusia dihadapi dengan ketidak mampuan secara fisik untuk berinteraksi dengan komputer. Sebagai contoh misalnya penderita cacat mata atau buta dan ketidak mampuan anggota badan dalam menggunakan media *input* seperti *mouse* dan *keyboard*. Hal ini yang menjadi dasar pemikiran betapa pentingnya *Natural Language* bagi kalangan penderita cacat yang akan berinteraksi dengan komputer. Hasil akhir yang diharapkan nantinya pengguna akhir melakukan *query* basis data melalui aplikasi tertentu yang dibuat khusus. Hasil penelitian adalah sebuah aplikasi *speech recognition* yang digunakan untuk

menterjemahkan perintah suara menjadi perintah *query* yang dimengerti oleh komputer. Dengan adanya aplikasi ini maka akan sangat membantu bagi *database administrator* terutama yang memiliki kekurangan dalam berinteraksi dengan komputer melalui *keyboard* tetapi tidak menutup kemungkinan aplikasi ini digunakan untuk umum.

Pada penelitian Nurcahyono (2010) dengan judul Kontrol Peralatan Rumah Tangga Jarak Jauh Didalam Aplikasi Menggunakan *Speech Recognition* sebagai input yang akan diproses menjadi *speech-to-text* dan diubah ke dalam bit-bit biner untuk dikirim ke *microcontroller* melalui *serial port*. Pada *paper* ini telah dilakukan pembuatan aplikasi *speech recognition* menggunakan bahasa pemrograman Delphi. *Speech-API* menggunakan *ISpRecoContext* untuk antar muka utama bagi aplikasi sehingga memerlukan komponen *CLSID_SpSharedRecoContext*. Kemudian pengaturan notifikasi untuk *event* yang dibutuhkan menggunakan *ISpNotifySource*. Danyang terakhir adalah me-load grammar dari dalam file yang telah dibuat dan diaktifkan menggunakan *IspReco Grammar* untuk mengenali kata yang berbeda.

Hasil pada proyek akhir ini menitik beratkan pada rata-rata kualitas aplikasi yang bernilai lebih besar sama dengan 80% dengan responden 1 yang memiliki rata-rata kualitas sebesar 98%. Kemudian diintegrasikan dengan *microcontroller* melalui *serial port* untuk kontrol peralatan rumah tangga jarak jauh.

Pada penelitian Noertjahyana (2003) yang berjudul Implementasi Sistem Pengenalan Suara Menggunakan Sapi 5.1 Dan Delphi 5. Pada sistem ini juga dikembangkan standard interface *Speech-API (Speech Application Programming Interface)* yang saat ini telah mencapai versi 5.1. Dengan adanya *SAPI* memungkinkan pembuat aplikasi untuk mengimplementasikan sistem pengenalan pembicaraan dengan menggunakan *engine* yang berbeda tanpa merubah aplikasi yang telah dibuat. Penelitian ini mengimplementasikan sistem pengenalan pembicaraan dengan menggunakan *SAPI 5.1*, *engine* Microsoft Speech Engine dan bahasa pemrograman Delphi 5 yang digunakan untuk melakukan diktasi berbahasa Inggris pada aplikasi berbasis teks.

Dengan menggunakan Microsoft Speech Engine pembuat aplikasi dapat mengimplementasikan kemampuan pengenalan pembicaraan yang dibuat untuk sistem

operasi Windows secara cepat dan mudah dan tidak tergantung bahasa pemrograman yang dipakai. Dengan menggunakan Speech-API 5.1 untuk pengenalan pembicaraan, aplikasi tidak terbatas menggunakan salah satu *engine* tertentu saja untuk pengenalan pembicaraan, tetapi dapat menggunakan *engine* lain yang diinginkan selama *engine* tersebut didesain sesuai dengan standard Speech-API 5.1. Speech-API 5.1 memberikan hampir semua antar muka, tipe dan konstanta yang penting melalui registered type library, sehingga memungkinkan pembuat aplikasi untuk mengakses Speech-API 5.1 melalui *late bound* ataupun *early bound automation* secara mudah. Untuk pengembangan lebih lanjut bisa dilakukan perbandingan kemampuan *engine* yang mendukung Speech-API 5.1 yang ada di pasaran sehingga didapatkan *engine* yang mempunyai keakuratan tinggi.

Pada penelitian Permadi, T (2008) yang berjudul Pemanfaatan Microsoft Speech Application Programming Interface Pada Pembuatan Aplikasi Perintah Suara. Dengan menggunakan masukan suara, akan lebih mempermudah pengguna dalam melakukan pengekseskusan suatu file. Walaupun setiap orang memiliki ciri khas suara masing-masing namun pada aplikasi perintah suara ini terdapat menu pelatihan yang akan membuat pengguna dapat melakukan sinkronisasi dengan komputer tersebut. Dan itu merupakan kunci utama dalam pembuatan aplikasi perintah suara ini Aplikasi ini dibuat dengan maksud agar seseorang dapat melakukan pengekseskusan file melalui suara (vokal) sebagai alternatif lain dari pengekseskusan file secara konvensional, dengan memanfaatkan components yang terdapat pada MICROSOFT Speech Application Programming Interface. Sehingga pengguna hanya akan menyuarakan kata kunci yang telah ia masukan dan aplikasi perintah suara ini telah mengetahui direktori tempat file yang akan dipanggil itu berada, maka file bertipe *.exe tersebut akan muncul (aktif).

Dari pengujian yang telah dilakukan diatas maka dapat disimpulkan bahwa aplikasi perintah suara ini dapat mengenali dengan baik masukan yang diinginkan oleh pengguna. dan Speech Application Programming Interface dapat pula dimanfaatkan untuk menciptakan aplikasi perintah suara. Dengan menggunakan pemanggilan method yang terdapat pada Speech Application Programming Interface dan dihubungkan dengan program yang telah dibuat pada Microsoft Visual Basic 6.0 aplikasi perintah suara ini dapat terwujud. Dimana keuntungan dari penggunaan

aplikasi ini adalah adanya metode baru dalam pengeksekusian suatu file. Pengguna dapat dengan mudah melakukan pengeksekusian file (.exe). Pengguna tidak perlu repot membuat sistem yang dapat mengenali suara, karena pada hal tersebut telah terdapat pada Speech Application Programming Interface.

Pada penelitian Supriyono, S. (2008) yang berjudul Perancangan Aplikasi Voice User Interface dengan Menggunakan Microsoft Speech-API. Metode baru yang baru diperkenalkan untuk menggantikan GUI adalah VUI (Voice User Interface). Perkembangan VUI ini ditunjang oleh teknologi ASR (Automatic Speech Recognizer). Dengan adanya teknologi ASR tersebut memungkinkan komputer untuk mengenali suara manusia sesuai dengan bahasa yang digunakan. Dengan VUI ini nantinya diharapkan komunikasi antara manusia dan komputer dapat dilakukan senatural mungkin seperti komunikasi antara manusia dan manusia. Microsoft menyertakan teknologi ASR tersebut pada Sistem Operasi yang dibuatnya. Oleh Microsoft, teknologi ASR digabungkan dengan teknologi Speech Synthesis, yaitu teknologi yang dapat mensintesis suara manusia, menjadi suatu API (Application Programming Interface) yang disebut Microsoft Speech-API.

Berdasarkan hasil pengujian pengaruh kualitas *microphone* terhadap aplikasi *Voice User Interface*, disimpulkan hal-hal berikut adalah penggunaan SONY Dynamic Microphone VJ-F22/C menghasilkan presentase ketepatan pengenalan suara sebesar 96 % pada *mode command* dan 90,0 % pada *mode dictation*. Penggunaan SPEED SP-501 Headset menghasilkan presentase ketepatan pengenalan suara sebesar 100 % pada *mode command* dan 92,5 % pada *mode dictation*. Penggunaan Logitech Premium Headset menghasilkan presentase ketepatan pengenalan suarasebesar 100 % pada *mode command* dan 92,5 % pada *mode dictation*.

Semakin sering dilakukan pelatihan pengenalan suara maka presentase ketepatan pengenalan suara oleh aplikasi *Voice User Interface* akan semakin mendekati 100%.. Aplikasi *Voice User Interface* yang berjalan pada *mode command* memiliki presentase ketepatan pengenalan suara yang hampir mendekati 100 %, hal ini disebabkan karena kata-kata yang akan didengarkan oleh *speech recognizer engine* telah ditetapkan terlebih dahulu. Aplikasi *Voice User Interface* yang berjalan pada *mode dictation* memiliki presentase ketepatan pengenalan suara yang kurang baik, hal ini disebabkan karena *speech recognizer engine* menggunakan seluruh kata dalam

database di dalam *grammarnya* sehingga pengucapan yang salah sedikit saja akan menyebabkan kesalahan pengenalan.

Pada penelitian Gupta, A (2014) yang berjudul *A Secure Face Tracking System*. Dalam penelitian ini, aplikasi baru dari sistem pengenalan wajah menggunakan Eigenvektor, Fisher berdasarkan PCA dan LBPH (Pola biner Lokal Histogram) berdasarkan LBP diusulkan. Kombinasi Fisher dan LBPH meningkatkan kemampuan pendekatan Eigen Vector. Fisher memberikan lebih akurat dan LBPH memberikan citra yang lebih seragam yang lebih tahan terhadap perubahan dalam pencahayaan. Aplikasi ini berfokus pada tiga modul yaitu:

1. Face Recognition: Fase ini meliputi deteksi wajah dan bagian. Dengan menggunakan pendekatan dan algoritma yang berbeda, wajah adalah terdeteksi pada kamera dan cocok dengan gambar yang sudah disimpan dalam Database. Kompatibilitas melangkah lebih jauh dengan beberapa gambar kamera pengolahan.
2. Authentication: Ini bagian dari penelitian ini mencakup otentikasi dan memeriksa pengguna sebelum masuk dia ke sistem dengan menggunakan pidato ke teks dan teks ke fitur speech.
3. Lokasi Tracking: ini adalah modul yang paling penting dari studi dimana lokasi seseorang dilacak dengan mengenali wajah orang di berbagai kamera CCTV dipasang di berbagai lokasi ditempat. Fitur terakhir terlihat juga dapat ditambahkan untuk memberikan yang lebih baik fungsi dari sistem.

Setelah ketiga modul ini digabung bersama-sama dalam sebuah aplikasi akan membantu dalam mengembangkan sistem yang akan sangat membantu bagi masyarakat.

Pada penelitian Kamarudding, R (2013) yang berjudul *Low Cost Smart Home Automation via Microsoft Speech Recognition*. Evolusi dan perkembangan di rumah secara otomatisasi bergerak maju menuju masa depan dalam menciptakan lingkungan yang ideal rumah pintar. Opsional, desain sistem otomatisasi rumah juga telah mengembangkan untuk situasi tertentu yang bagi mereka yang membutuhkan perhatian khusus seperti orang tua, pasien yang sakit, dan orang cacat. Dengan demikian, memberikan skema kontrol yang sesuai dengan menggunakan modus protokol nirkabel dapat membantu mereka melakukan rutinitas sehari-hari mereka. Selain itu, akses kontrol suara akan digunakan sebagai perintah untuk tujuan yang

lebih baik. Dalam penelitian ini, kami menyajikan sistem otomatisasi rumah pintar menggunakan pengenalan suara. Ruang lingkup pekerjaan penelitian ini akan mencakup kontrol dan sistem pemantauan untuk peralatan rumah dari Graphical User Interface (GUI) menggunakan perangkat lunak Microsoft Visual Basic yang menggunakan mesin Microsoft Speech Recognition sebagai sumber masukan dan menjadi kontrol nirkabel. Metodologi penelitian yang terlibat adalah penerapan pengetahuan di bidang komunikasi frekuensi radio, mikrokontroler dan pemrograman komputer. Akhirnya, hasilnya akan diamati dan menganalisis untuk mendapatkan solusi yang lebih baik di masa depan.

Proyek ini menyajikan desain dan implementasi sistem otomatisasi rumah dengan komputer pemantauan berdasarkan pengenalan suara dan fitur kontrol jaringan. Proyek yang diusulkan sepenuhnya dilaksanakan dan dikembangkan. Hasil penelitian ini dapat memudahkan orang terutama mereka yang memiliki kekurangan seperti gangguan penglihatan dan juga orang tua. Meskipun saat ini banyak pengembang menciptakan sistem rumah pintar, hanya menggunakan suara beberapa untuk mengakses pusat kendali. Dengan Speech-API 5.3.

Penggunaan komunikasi nirkabel membuat sistem lebih nyaman dan mudah untuk menginstal sebagai gantinya menggunakan komunikasi kabel. Selain itu, disediakan konsumsi daya biaya rendah. Berdasarkan proyek ini, itu sepenuhnya memenuhi persyaratan dari tujuan yang diberikan.

Pada penelitian Miah, S. (2010) yang berjudul To Design Voice Control Keyboard System using Speech Application Programming Interface. Seiring berjalannya waktu, orang-orang lebih bergantung pada perangkat elektronik. Tujuan utama dari proyek ini adalah untuk merancang dan mengembangkan Sistem Pengendalian Keyboard suara, yang dikendalikan sepenuhnya oleh komputer, dan menampilkan output pada layar perangkat dengan waktu yang telah ditetapkan. Jadi proyek ini akan bekerja sebagai sistem untuk membantu orang-orang yang memiliki pengetahuan tentang sistem komputer. Bahkan orang-orang yang buta huruf mereka dapat mengoperasikan komputer. Kita dapat menerapkan sistem yang dikembangkan di lain sistem, sistem kontrol mobil sebagai contoh suara. Suara proyek ini adalah input, suara antar muka pemrograman aplikasi (Speech-API 4.1). Pengenalan perintah

transfer suara ini untuk mikroprosesor yang sesuai dengan kode pemrograman dan perangkat display.

BAB 3

ANALISIS DAN PERANCANGAN

3.1 Analisis

3.1.1 Analisis Masalah

Aplikasi ini merupakan suatu alat yang berfungsi untuk membantu proses proteksi / unproteksi *folder* dengan menggunakan perintah suara (*Speech Command*). Suara yang diinputkan dalam bentuk bahasa alami dalam hal ini menggunakan bahasa Inggris yang kemudian diolah sehingga terbentuk suatu perintah yang berfungsi untuk memanipulasi *folder*. Perintah yang dimasukkan berbentuk perintah sederhana yaitu Close untuk menutup aplikasi, Key untuk memproteksi / unproteksi *folder*. Selanjutnya perintah dalam bentuk suara diterjemahkan ke dalam bentuk teks dan kemudian dirubah dalam bentuk perintah. Aplikasi manipulasi *folder* dapat digunakan untuk penerapan bagi user yang memiliki kurangan fisik sehingga mereka mudah menggunakannya hanya dengan perintah melalui suara. Aplikasi ini juga dapat digunakan dalam suatu perangkat keras yang tidak memiliki media input seperti *mouse* atau *keyboard*.

3.1.2 Usulan Pemecahan Masalah

Setelah melihat masalah keterbatasan fisik pengguna komputer dan mempelajarinya, maka diputuskan untuk merancang suatu program aplikasi yang dapat melakukan proteksi *folder* pada komputer yang berbasis Windows. Hasilnya adalah *folder* yang belum diproteksi akan diproteksi dan *folder* yang terproteksi akan di unproteksi. Sebagai hasil akhir adalah daftar listing aktifitas yang dilakukan user dalam mengoperasikan aplikasi yang dapat ditampilkan.

3.1.3 *Speech Application Programming Interface (S-API)*

Speech Application Programming Interface (Speech-API) merupakan *library* yang dapat digunakan pada pengenalan suara dalam bahasa Inggris dengan format streaming. Output library *Speech-API* ini berupa teks yang dapat digunakan untuk memanipulasi perintah-perintah pada komputer berbasis Windows.

3.1.4 Analisis Kebutuhan Perangkat Lunak

Kebutuhan *software* merupakan kebutuhan akan perangkat lunak yang digunakan untuk membangun program aplikasi. Diantara kebutuhan tersebut adalah sebagai berikut:

1. Sistem operasi berbasis Windows, yang penulis gunakan adalah sistem operasi Windows 7.
2. .NET Framework merupakan suatu komponen Windows yang terintegrasi dan dibuat agar dapat menjalankan berbagai macam aplikasi berbasis .NET yang penulis gunakan adalah .NET framework 3.5.
3. Tools / bahasa pemrograman, yang penulis gunakan adalah Microsoft Visual Basic.NET 2010.

3.1.5 Analisis Kebutuhan Perangkat Keras

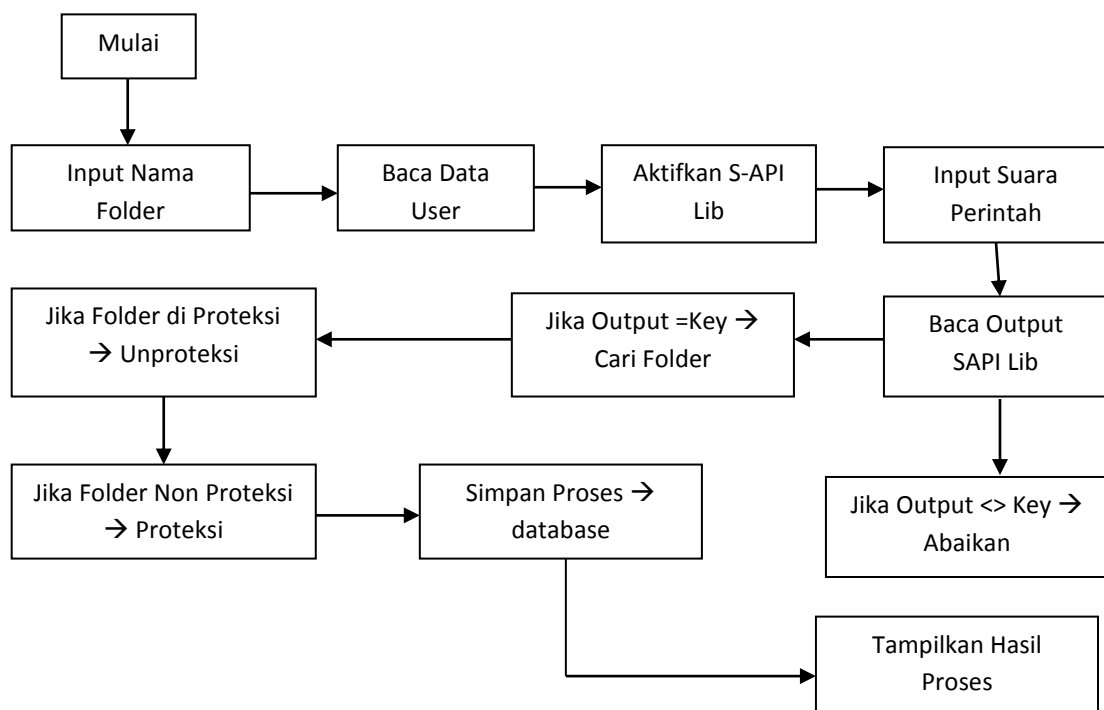
Kebutuhan perangkat keras (*hardware*) diantaranya yaitu sebagai berikut:

1. *Processor* minimum Intel Dual Core atau yang lebih baru.
2. *Harddisk* space minimum 250 GB.
3. RAM minimal 2 GB
5. *Motherboard*
6. Seperangkat I/O (*mouse, keyboard, monitor*)

3.2 Perancangan Sistem

3.2.1 General Architecture

Desain Arsitektur adalah perancangan penting yang menggambarkan proses, alur dan interaksi antar komponen dalam suatu sistem. Desain arsitektur dari suatu sistem yang merepresentasikan struktur data komponen pada aplikasi yang diperuntukkan dalam membangun suatu sistem (Pressman, 2010). Perancangan keseluruhan aplikasi ini akan dijabarkan pada arsitektur umum yang dapat dilihat pada Gambar 3.1.



Gambar 3.1 *General Architecture*

Penjelasan dari bagian - bagian yang ada pada *general architecture* adalah sebagai berikut:

a. **Input Nama Folder**

User memilih tombol Browse untuk menjalankan sistem melakukan *input*-an nama folder yang akan di-lock (proteksi).

b. Baca Data User

Sebelum melakukan proteksi berupa *lock folder*, sistem membuka database user untuk memastikan user yang sedang aktif. Jika user tidak aktif, maka proses proteksi folder dibatalkan dengan menampilkan notifikasi.

c. Aktifkan Speech-API Library

Jika user aktif, maka sistem menjalankan file *Speech-API library*, untuk pengenalan suara. Tampilan Speech-API berupa kotak kecil dengan tombol *Listening* dan sebuah tampilan teks untuk menampilkan kata yang masuk melalui micropone sistem.

d. Input Suara Perintah

User melakukan pemasukan suara melalui microphone dan diproses oleh Speech-API, jika suara dikenali, maka teks suara ditampilkan pada kotak teks sebagai output Speech-API.

e. Baca Output *Speech-API Lib*

Sistem membaca output dari proses *Speech-API* berupa teks yang berfungsi menjalankan proteksi folder berupa lock maupun unlock.

f. Jika Output =Key, maka cari folder

Sistem mencari folder pada komputer yang telah di-input pada saat pemilihan tombol Browse.

g. Jika Folder Non Proteksi

Sistem akan me-lock folder jika folder yang dicari belum diproteksi (unlocked).

h. Jika Folder Proteksi

Sistem akan me-unlock folder jika folder yang dicari sudah diproteksi (locked).

i. Simpan Proses

Sistem akan menyimpan data proteksi maupun unproteksi folder ke dalam database. Informasi yang disimpan berupa nama user yang melakukan proses

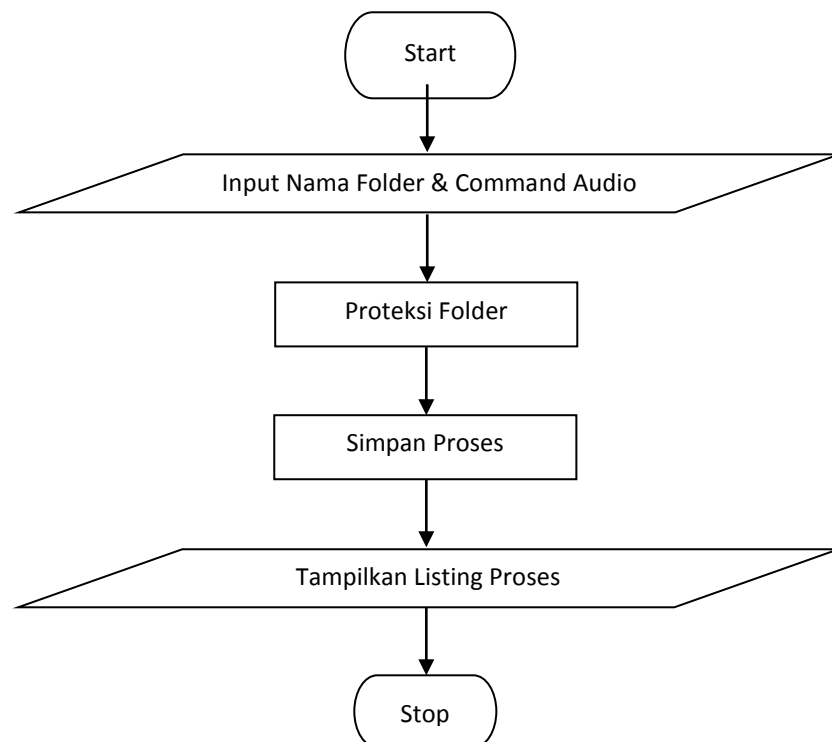
proteksi maupun unproteksi folder, tanggal proteksi, status folder, status proses (lock/unlock) serta nama folder.

j. Tampilkan Hasil Proses

Sistem akan menampilkan listing semua proses yang terjadi dalam hal proteksi maupun unproteksi folder yang ada pada database.

3.2.2 Flow Chart Sistem

Flow chart sistem proteksi folder menggunakan *Speech Application Programming Interface (Speech-API)* dapat dilihat seperti pada Gambar 3.2.

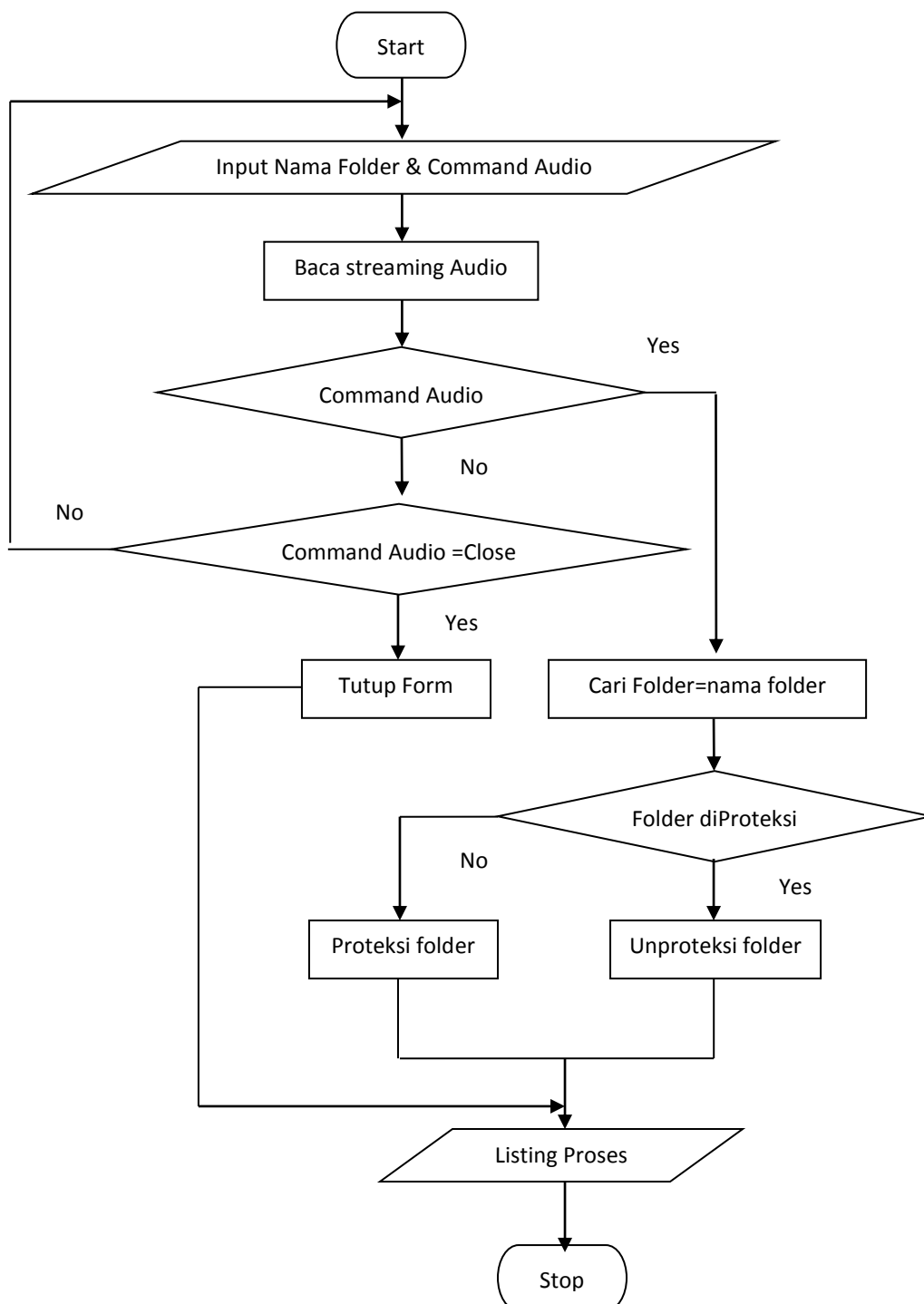


Gambar 3.2 Flow Chart Sistem

Pada Gambar 3.2 di atas, input berupa nama folder dan perintah suara yang akan dikenali oleh *Speech-API Library*. Selanjutnya *Speech-API Lib* memberikan *output* berupa teks ke sistem dan sistem melakukan eksekusi sesuai dengan perintah.

3.2.3 Flow Chart Proteksi Folder

Flow Chart Proteksi Folder adalah gambaran secara garis besar sistem dalam mendapatkan perintah dalam bentuk suara untuk proteksi folder seperti pada Gambar 3.3.



Gambar 3.3 Flow Chart Proteksi Folder

3.2.4 Perancangan Antarmuka

Interface (antarmuka sistem) diperlukan untuk mempermudah seorang *user* dalam menggunakan atau mengakses sebuah aplikasi. Antarmuka sistem merupakan sebuah alur komunikasi antara *user* dengan sistem.

Dengan kata lain antarmuka sistem digunakan sebagai media antara *user* dan komputer agar dapat berinteraksi satu sama lain. Sehingga *user* dapat lebih mudah mengerti dan menggunakan sistem tersebut.

3.2.4.1 Rancangan antarmuka (*User Interface*)

Beberapa prinsip dari perancangan antarmuka yang baik telah dikembangkan dan banyak di antaranya menekankan pada antar muka yang *user friendly* sehingga mudah digunakan. Penelitian pada bidang ini sering disebut sebagai *human factors engineering*. Berikut adalah elemen-elemen yang harus dipertimbangkan dalam perancangan antar muka.

a. Desain layar

Beberapa hal yang perlu dipertimbangkan dalam desain layar yaitu mengenai warna serta desain grafis yang digunakan pada antarmuka sistem dengan mengedepankan kenyamanan pengguna.

b. Umpan balik

Aspek yang paling penting dalam umpan balik adalah waktu respon, yaitu waktu antara saat *user* memasukkan data dengan respon yang diberikan oleh sistem.

c. Bantuan

Pada saat *user* sedang mengoperasikan sistem, seringkali mengalami kesulitan dan tidak mengetahui apa yang harus dikerjakan berikutnya. Perancangan antarmuka bantuan menyediakan langkah-langkah dalam menggunakan aplikasi.

d. Pengendalian kesalahan

Perancangan antarmuka harus memperhatikan pengendalian kesalahan yang dapat berupa sebagai berikut.

- Pencegahan kesalahan.

Sistem harus menyediakan instruksi yang jelas kepada pengguna tentang apa yang harus dilakukan sehingga pengguna tidak melakukan kesalahan yang seharusnya tidak perlu terjadi.

- Pendeteksian kesalahan.

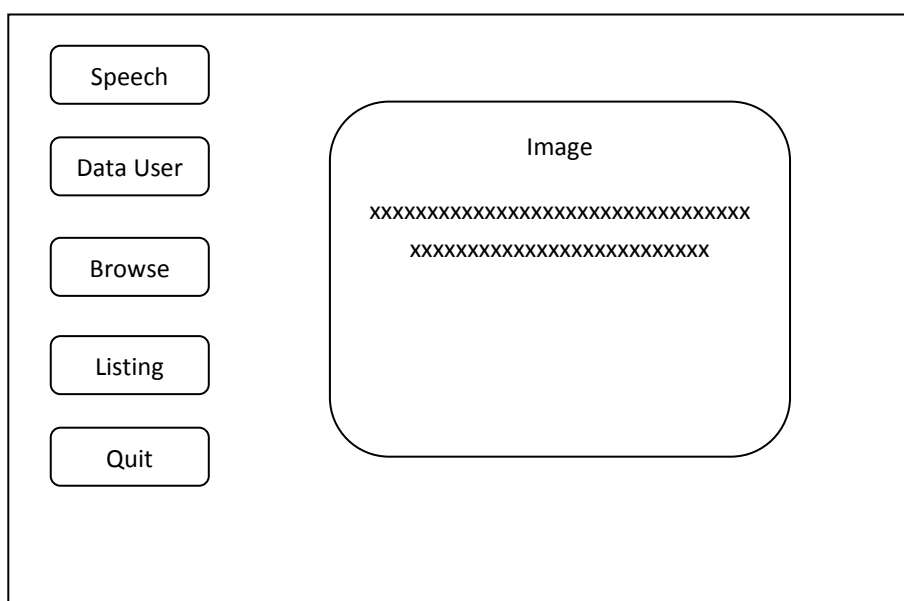
Jika suatu kesalahan terjadi, sistem harus dapat mengidentifikasi kesalahannya dengan jelas dan dapat menampilkan berita kesalahan.

- Pembetulan kesalahan.

Jika suatu data yang dimasukkan salah, maka sistem harus dapat memberikan kesempatan kepada *user* untuk dapat mengoreksinya.

3.2.4.2 Rancangan Menu Utama

Rancangan Menu Utama berfungsi untuk menampilkan program-program aplikasi pengenalan suara dalam memproteksi folder. Rancangan Menu Utama dapat dilihat seperti pada Gambar 3.4.



Gambar 3.4 Rancangan Menu Utama

Pada Gambar 3.4 di atas, terdapat 5 tombol menu yaitu *Speech*, *Data User*, *Browse*, *Listing*, *Help*, *About* serta *Quit*.

3.2.4.3 Rancangan *Speech*

Rancangan *Speech* berfungsi untuk melakukan proteksi/unproteksi *folder*. Rancangan *Speech* dapat dilihat pada Gambar 3.5.

The diagram illustrates the layout of the *Speech* application interface. It consists of several labeled components and numbered callouts:

- Perintah**: A text box containing placeholder text 'xxxxxxxxxxxxxxxxxxxxxx', labeled with a circled '1'.
- Nama Folder**: A text box containing placeholder text 'xxxxxxxxxxxxxxxxxxxxxx', labeled with a circled '2'.
- Data Suara**: A text box containing placeholder text 'Xxxxxxxxxxxxxxxxxxxxxx' and 'xxxxxxxxxxxxxxxxxxxxxx', labeled with a circled '3'.
- Volume Suara**: A text box containing placeholder text 'xxxxxxxxxx', labeled with a circled '4'.
- List Folder**: A list box containing placeholder text 'Xxxxxxxxxxxxxxxxxx', 'xxxxxxxxxxxxxxxxxxxx', 'xxxxxxxxxxxxxxxxxxxx', and 'xxxxxxxxxxxxxxxxxxxx', labeled with a circled '5'.

Gambar 3.5 Rancangan *Speech*

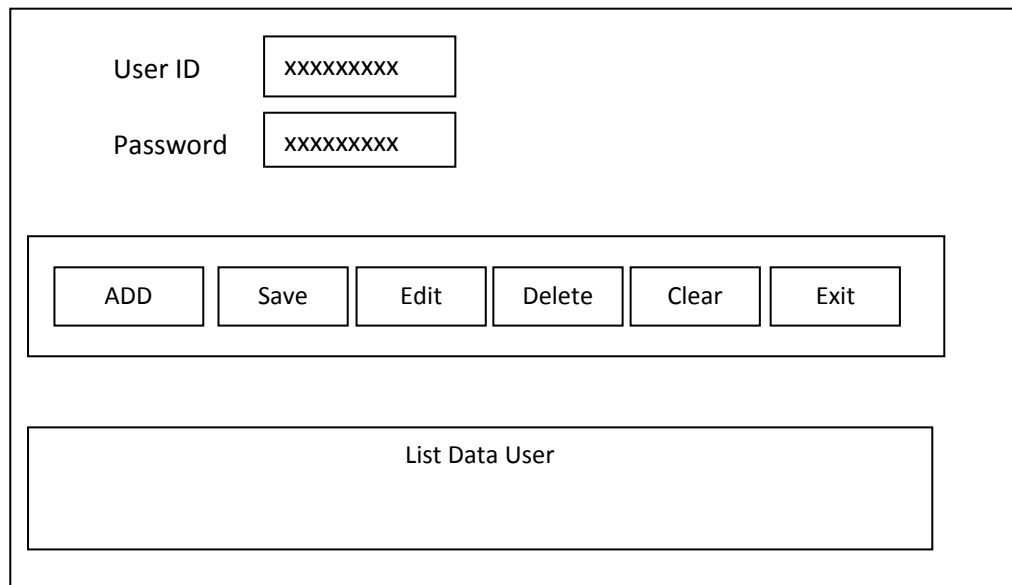
Keterangan :

1. *Text box* untuk menampilkan perintah suara.
2. *Text box* untuk menampilkan nama folder yang diproteksi.
3. *Text box* untuk menampilkan data file suara.
4. *Progressbar* untuk menampilkan volume suara.
5. *Text box* untuk menampilkan folder-folder dalam komputer.

3.2.4.4 Rancangan *Data User*

Rancangan *User* adalah tampilan berfungsi untuk melakukan pemasukan data nama pengguna (*UserId*) dan *Password*. Pada rancangan ini terdapat empat tombol. Tombol *Add* berfungsi untuk menambah data *User*, tombol *Save* berfungsi untuk menyimpan, tombol *Edit* berfungsi untuk melakukan perubahan data, tombol *Delete* berfungsi untuk menghapus data, tombol *Clear* berfungsi untuk melakukan

pembatalan dan tombol Exit berfungsi untuk keluar dari halaman User dan kembali ke Menu Utama. Bentuk perancangan User dapat dilihat pada Gambar 3.6.

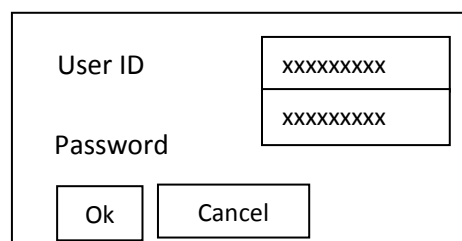


The diagram illustrates a user management interface. It features two input fields at the top: 'User ID' and 'Password', both containing the placeholder text 'xxxxxxxx'. Below these fields is a horizontal row of six buttons: 'ADD', 'Save', 'Edit', 'Delete', 'Clear', and 'Exit'. At the bottom of the interface is a large rectangular box labeled 'List Data User'.

Gambar 3.6 Rancangan User

3.2.4.5 Rancangan Login

Rancangan *Login* adalah rancangan untuk melakukan otoritas petugas sebagai pengguna sistem. Tampilan ini terdiri dari pemasukan *UserID* serta *password*. Setelah mengisi data di atas, maka pilih tombol *Ok* dan jika hendak menutup login pilih *Cancel*. Rancangan *Login* dapat dilihat pada Gambar 3.9.



The diagram shows a login interface. It includes two input fields: 'User ID' and 'Password', each with a placeholder 'xxxxxxxx'. Below the 'Password' field are two buttons: 'Ok' and 'Cancel'.

Gambar 3.7 Rancangan Login

3.2.4.6 Rancangan Browse

Rancangan *Browse* adalah rancangan untuk memilih folder yang akan diproteksi. Tampilan ini terdiri dari tombol Browse untuk melakukan pencarian folder pada komputer, tombol *Ok* untuk menetapkan folder, tombol Clear untuk membatalkan folder dan jika hendak menutup pilih Exit. Rancangan *Browse* dapat dilihat pada Gambar 3.8.

Nama Folder Yang Di-Lock

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Browse Ok Clear Exit

Gambar 3.8 Rancangan Browse

3.2.4.7 Rancangan Listing Proses

Rancangan Listing Proses adalah rancangan untuk proses-proses yang telah dilakukan untuk memproteksi maupun melakukan unproteksi folder. Tampilan ini terdiri dari tombol Browse untuk melakukan pencarian folder pada komputer, tombol *Ok* untuk menetapkan folder, tombol Clear untuk membatalkan folder dan jika hendak menutup pilih Exit. Rancangan *Browse* dapat dilihat pada Gambar 3.9.

ID	Tanggal	Jam	Nama Proses	Nama Folder	UserID
Xx	dd-mm-yyyy	hh:mm	xxxxxxxxxx	xxxxxxxxxx	xxxxxx
Xx	dd-mm-yyyy	hh:mm	xxxxxxxxxx	xxxxxxxxxx	xxxxxx
Xx	dd-mm-yyyy	hh:mm	xxxxxxxxxx	xxxxxxxxxx	xxxxxx
Xx	dd-mm-yyyy	hh:mm	xxxxxxxxxx	xxxxxxxxxx	xxxxxx

Gambar 3.9 Rancangan Listing Proses

BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Implementasi Keamanan Folder Menggunakan *Speech Application Programming Interface (Speech-API)* adalah tampilan hasil penulisan program (*coding*) yang terdiri dari program pengenalan suara untuk memproteksi folder serta program pendukung lainnya.

4.1.1 *Tampilan Menu Utama*

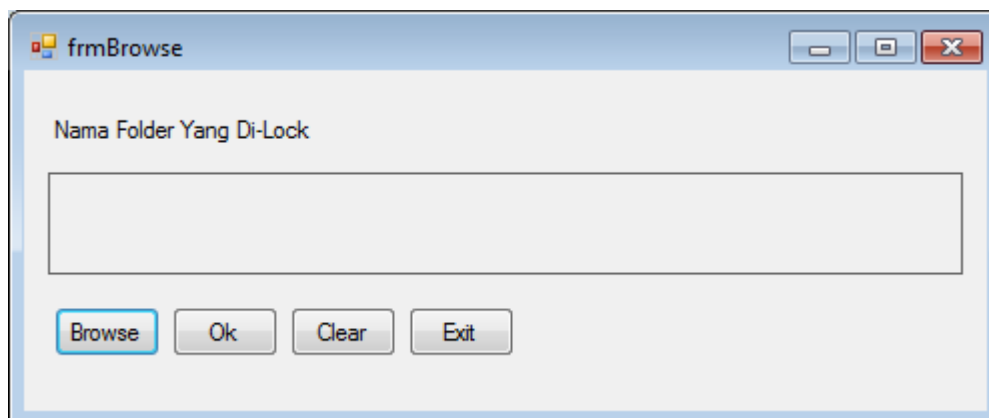
Tampilan Menu Utama berisi judul skripsi, gambar latar serta tampilan menu. Tampilan Menu terdiri dari tombol menu Speech, Data User, Browse, Listing, *Help*, *About* serta *Quit* dapat dilihat pada Gambar 4.1.



Gambar 4.1 Tampilan Menu Utama

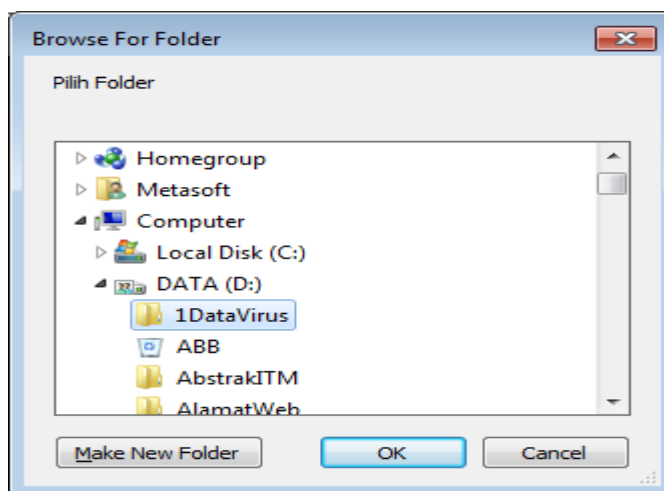
4.1.2 Tampilan Browse

Tampilan Browse merupakan tampilan berguna untuk melakukan pemilihan nama folder yang akan diproteksi maupun di-unproteksi. Tampilan Browse dapat dilihat pada Gambar 4.2.



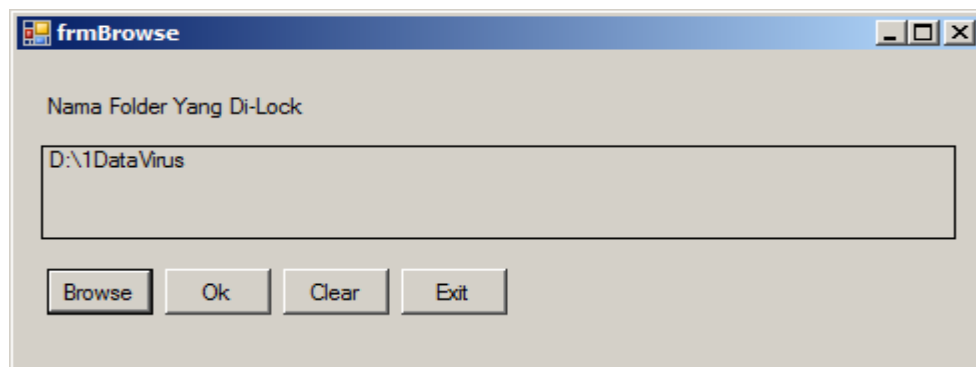
Gambar 4.2 Tampilan Browse

Pada Gambar 4.2 di atas pilih tombol Browse untuk menetapkan nama folder yang akan diproteksi seperti pada Gambar 4.3.



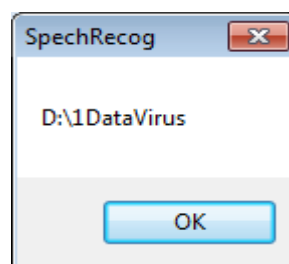
Gambar 4.3 Pemilihan Folder yang akan diproteksi

Pada Gambar 4.3 di atas pilih salah satu nama folder yang akan diproteksi, hasilnya seperti seperti pada Gambar 4.4.



Gambar 4.4 Tampilan Browse

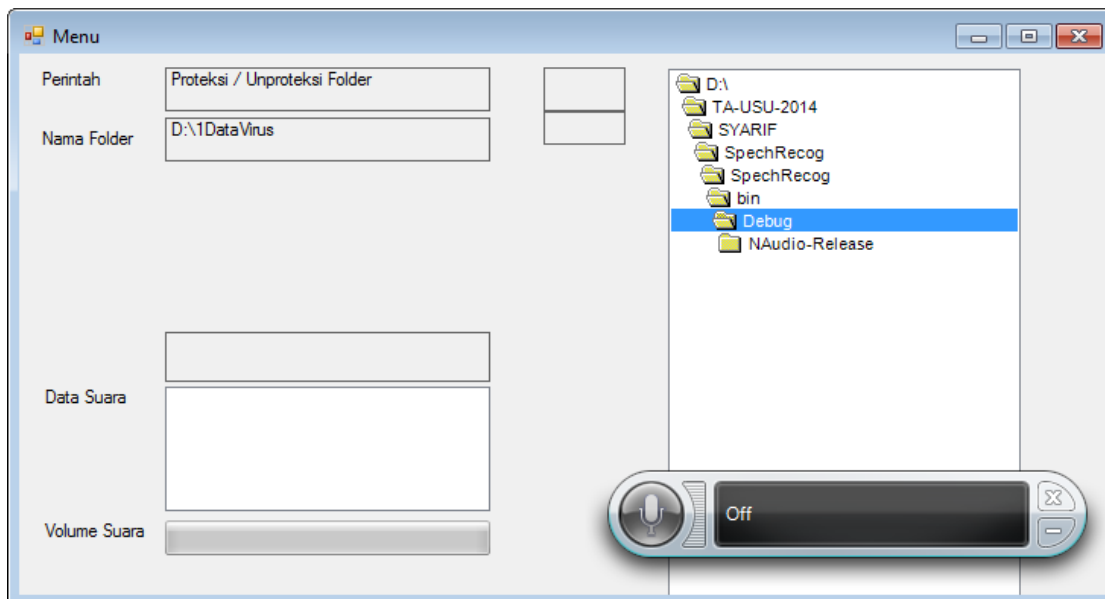
Pada Gambar 4.4, terlihat pemilihan nama folder yang akan diproteksi dengan nama D:\1DataVirus dan untuk melanjutkannya pilih tombol Ok untuk menampilkan konfirmasi nama folder. Selanjutnya pilih Ok pada tampilan konfirmasi pemilihan folder untuk kembali ke Menu Utama.



Gambar 4.5 Tampilan Konfirmasi Pemilihan Folder

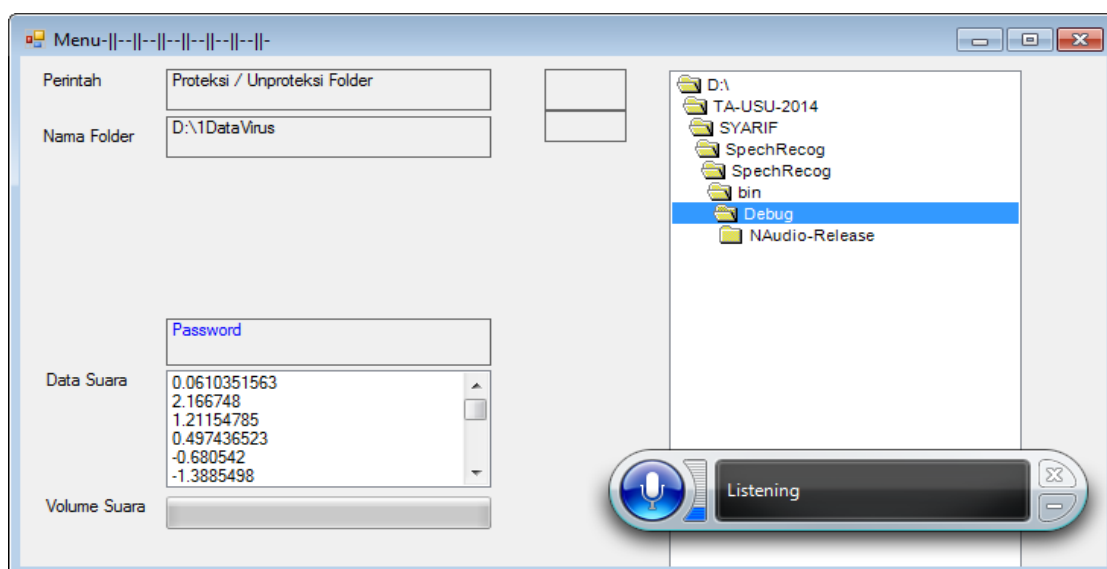
4.1.3 Tampilan SpeechLock Folder

Dari menu utama pilih tombol Speech yang berguna untuk menampilkan form Speech untuk me-lock maupun un-lock folder. Tampilan Speech untuk me-lock folder dapat dilihat pada Gambar 4.6.



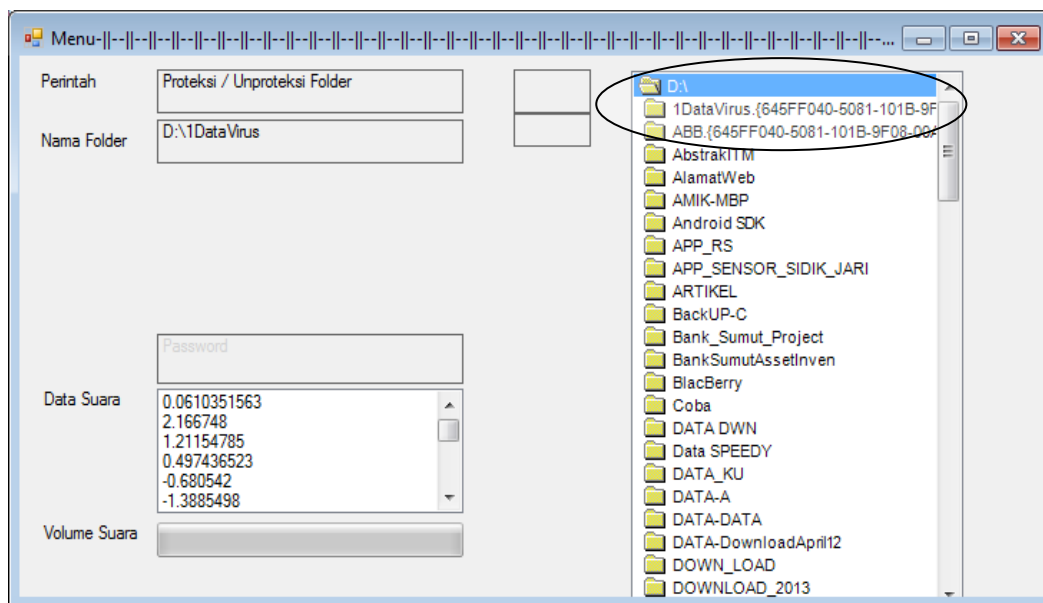
Gambar 4.6 Tampilan Speech

Pada Gambar 4.6 di atas, terlihat nama perintah, nama folder yang akan di-lock data suara, volume suara, tampilan tree folder serta tools Speech-API untuk mengenali suara perintah dari user. Selanjutnya aktifkan tools Speech-API untuk merekam suara perintah, dilanjutkan dengan penginputan suara melalui micropone berupa perintah untuk login ke sistem dengan mengucapkan “KEY”. Selanjutnya jika suara perintah dikenal, maka sistem akan me-lock folder yang sudah dipilih sebelumnya seperti pada Gambar 4.7.



Gambar 4.7 Tampilan Speech dalam memproteksi folder.

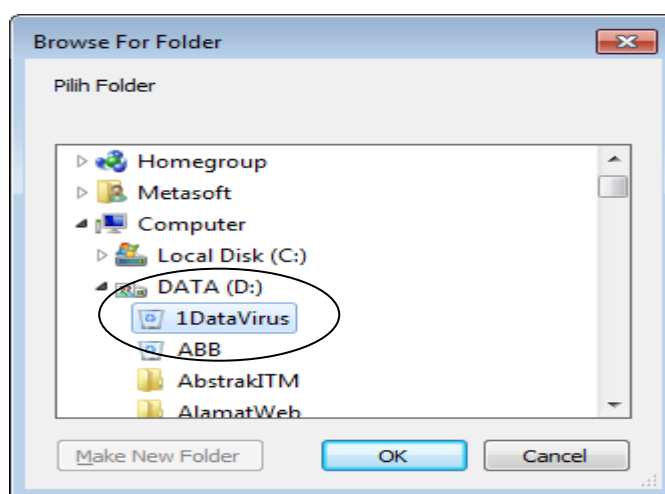
Untuk melihat hasil proteksi folder, maka buka folder pada tampilan tree folder, dimana folder yang telah diproteksi telah berubah seperti pada Gambar 4.8.



Gambar 4.8 Tampilan Hasil Proteksi folder.

4.1.4 Tampilan SpeechUnLock Folder

Dari menu utama pilih tombol Speech yang berguna untuk menampilkan *form Speech* untuk mem-unlock folder. Pertama-tama pilih tombol Browse dari menu utama dan pilih folder yang akan di-unlock seperti yang terlihat pada Gambar 4.9.



Gambar 4.9 Pemilihan Folder yang akan di-unlock

Tampilan Speech untuk me-unlock folder dapat dilihat pada Gambar 4.8.

4.1.5 Tampilan Data User

Tampilan Data User merupakan tampilan berguna untuk melakukan pemasukan nama-nama user yang akan mengoperasikan perangkat lunak. Tampilan Data user dapat dilihat pada Gambar 4.10.

The 'Data User' window contains input fields for 'User ID' and 'Password', both with the value 'syarif'. Below these is a 'Tombol' (Button) section with 'Add', 'Save', 'Query', 'Edit', 'Delete', 'Clear', and 'Quit'. At the bottom is a table with columns 'UserID' and 'Password'.

UserID	Password
syarif	syarif

Gambar 4.10 TampilanData User

4.1.6 Tampilan Login

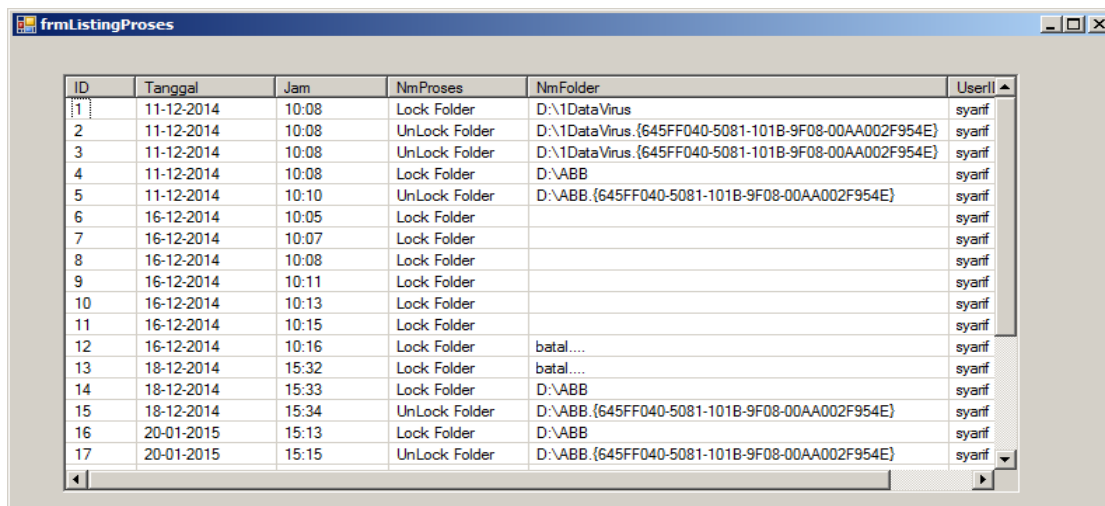
Tampilan Login merupakan tampilan berguna untuk melakukan pemasukan otoritas pengguna sistem pengenalan suara. Tampilan Login dapat dilihat pada Gambar 4.11.

The 'Login' window has input fields for 'User name' (containing 'syarif') and 'Password' (containing seven asterisks). Below these is a 'Tombol' (Button) section with 'Ok' and 'Cancel' buttons.

Gambar 4.11 TampilanData User

4.1.7 Tampilan Listing

Tampilan Listing merupakan tampilan berguna untuk menampilkan semua proses yang telah dilakukan. Tampilan Listing dapat dilihat pada Gambar 4.12.



ID	Tanggal	Jam	NmProses	NmFolder	UserID
1	11-12-2014	10:08	Lock Folder	D:\1DataVirus	syarif
2	11-12-2014	10:08	UnLock Folder	D:\1DataVirus.{645FF040-5081-101B-9F08-00AA002F954E}	syarif
3	11-12-2014	10:08	UnLock Folder	D:\1DataVirus.{645FF040-5081-101B-9F08-00AA002F954E}	syarif
4	11-12-2014	10:08	Lock Folder	D:\ABB	syarif
5	11-12-2014	10:10	UnLock Folder	D:\ABB.{645FF040-5081-101B-9F08-00AA002F954E}	syarif
6	16-12-2014	10:05	Lock Folder		syarif
7	16-12-2014	10:07	Lock Folder		syarif
8	16-12-2014	10:08	Lock Folder		syarif
9	16-12-2014	10:11	Lock Folder		syarif
10	16-12-2014	10:13	Lock Folder		syarif
11	16-12-2014	10:15	Lock Folder		syarif
12	16-12-2014	10:16	Lock Folder	batal....	syarif
13	18-12-2014	15:32	Lock Folder	batal....	syarif
14	18-12-2014	15:33	Lock Folder	D:\ABB	syarif
15	18-12-2014	15:34	UnLock Folder	D:\ABB.{645FF040-5081-101B-9F08-00AA002F954E}	syarif
16	20-01-2015	15:13	Lock Folder	D:\ABB	syarif
17	20-01-2015	15:15	UnLock Folder	D:\ABB.{645FF040-5081-101B-9F08-00AA002F954E}	syarif

Gambar 4.12 Tampilan Listing

4.2 Hasil Pengujian

Pengujian aplikasi Implementasi Keamanan Folder Menggunakan *Speech Application Programming Interface (Speech-API)* adalah pengumpulan hasil proteksi dan un-proteksi folder. Hasil pengujian dapat dilihat seperti pada Tabel 4.1.

Tabel 4.1 Hasil Pengujian Proteksi Folder

No	Nama Folder Asli	Hasil Proteksi Folder	Jumlah Pengulangan Perintah	Keterangan
1	D:\Syarif_coba	D:\Syarif_coba	1x	Proses dilakukan didalam tertutup
2	D:\Syarif_coba	D:\Syarif_coba	4x	Proses dilakukan dalam keadaan ramai
3	D:\Syarif_coba	D:\Syarif_coba	1x	Proses dilakukan dengan suara berbeda
4	D:\Syarif_coba	D:\Syarif_coba	2x	Proses dilakukan didalam ruang terbuka
	Jumlah	8 x		

Keakuratan: $\text{Jumlah data} / \text{Jumlah Pengulangan} = 4 / 8 \times 100 = 50 \%$

Tabel 4.2 Hasil Pengujian Un-Proteksi Folder

No	Nama Folder Asli	Hasil Proteksi Folder	Jumlah Pengulangan Perintah	Keterangan
1	D:\Syarif_coba	D:\Syarif_coba	1x	Proses dilakukan didalam ruang tertutup
2	D:\Syarif_coba	D:\Syarif_coba	3x	Proses dilakukan dalam keadaan ramai
3	D:\Syarif_coba	D:\Syarif_coba	2x	Proses dilakukan dengan suara berbeda
4	D:\Syarif_coba	D:\Syarif_coba	2x	Proses dilakukan didalam ruang terbuka
	Jumlah		8 x	

Keakuratan: $\text{Jumlah data} / \text{Jumlah Pengulangan} = 4 / 8 \times 100 = 50 \%$

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah merancang dan mengaplikasikan perangkat lunak Keamanan Folder Menggunakan *Speech Application Programming Interface* (S-API), maka diperoleh hasil pengujian adalah sebagai berikut:

- a. Aplikasi dapat melakukan proteksi folder yang ada pada komputer.
- b. Tingkat keakuratan untuk memproteksi folder: 50%
- c. Tingkat keakuratan untuk me Un-proteksi folder: 50 %

5.2 Saran

Adapun saran-saran yang untuk penelitian maupun pengembangan berikutnya adalah:

1. Melakukan proses selain proteksi antara lain meng-*copy* file, mematikan komputer dan lain sebagainya.
2. Melakukan *scanning* folder serta menampilkan filenya.
3. Agar meningkatkan akurasi keberhasilan aplikasi, agar suara yang masuk dapat dibatasi hanya suara perintah manusia tanpa adanya noise (suara lain).
4. Agar hasil dari aplikasi ini dapat dibandingkan dengan aplikasi keamanan folder dengan algoritma yang lainnya antara lain adalah algoritma Jaringan syaraf tiruan (JST).

DAFTAR PUSTAKA

- Binanto, I. 2010. *Multimedia Digital, Dasar Teori + Pengembangan*, Yogyakarta: ANDI.
- Syarif, A., Daryanto, T.&Arifin, M. Z. 2011. *Aplikasi Speech Application Programming Interface (S-API) 5.1 Sebagai Perintah Untuk Pengoperasian Aplikasi Berbasis Windows*. Seminar Nasional Aplikasi Teknologi Informasi 2011 (SNATI 2011) Yogyakarta, 17-18 Juni 2011.
- Sadeli, M. 2012. *Aplikasi Mini Market dengan Visual Basic 2010 untuk Orang Awam*. Penerbit: Maxikom. Palembang.
- Jogiyanto. 2005. *Analisis dan desain Sistem Informasi*. Penerbit: ANDI. Yogyakarta.
- Supriyono, B. 2010. *Perancangan Aplikasi Voice User Interface dengan Menggunakan Microsoft Speech API*. Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang.
- Noertjahyana, A. & Adipranata R. 2010. Implementasi Sistem Pengenalan Suara Menggunakan Sapi 5.1 Dan Delphi 5. Jurnal Ilmiah Fakultas Teknologi Industri, Jurusan Teknik Informatika – Universitas Kristen Petra. Surabaya.
- Chandrakant, P. V., Sampat, T. S. & Manikra, G. K. 2014. Web application for visually impaired Deaf User. International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 10, October 2014.
- Dorozhkin, D. V. & Vance, J. M. 2002. Implementing Speech Recognition In Virtual Reality. Proceedings of DETC.02ASME 2002 Design Engineering Technical Conferences and Computer and Information in Engineering Conference. Montreal, Canada, September 29-October 2, 2002. Department of Mechanical Engineering Virtual Reality Applications Center Iowa State University.
- Fulkerson, M. S. & Biermann, A. W. 2010. A Toolkit for Building Speech-Enabled Applications Department of Computer Science Duke University Durham, North Carolina 27708, USA.
- Gupta, A., Gandhi, P. & Jagga, U. 2014. A secure face tracking system. International Journal of Information & Computation Technology. Volume 4, Number 10 (2014), pp. 959-964. Paschim Vihar East, Delhi, INDIA.

Ian, B.& Paul, M. 2012. *Web Based Maltese Language Text to Speech Synthesiser*. Faculty of ICT, Department of Computer & Communications Engineering.

Kadam,A. J., Deshmukh, P., Kamat, A., Joshi, N., Doshi, R.2012. *Speech Oriented Computer System Handling*. International Conference on Intelligent Computational Systems Jan. 7-8, 2012 Dubai.

Kamarudin, R., Aiman F.,&Yusof. 2013. *Low Cost Smart Home Automationvia Microsoft Speech Recognition*. International Journal of Engineering & Computer Science Vol:13 No:03 6. June 2013. Faculty of Electronic and Computer EngineeringUniversiti Teknikal Malaysia Melaka Hang Tuah Jaya76100 Durian Tunggal, Melaka, Malaysia.

Manjushree Mahajan et al, 2012. *A Multimedia Interactive System*. International Journal of Computer Science and Information Technologies, Vol. 3 (2),Computer Engineering DepartmentG.H.Raisoni college of Engg.(Pune University),Pune, India.

Maind, S. B. &Dehankar A. V. 2014. Review on Hand And Speech Based Interaction With Mobile Phone. Journal of Computer Science (IOSR-JCE). International Conference on Advances in Engineering & Technology – 2014. Department of Computer Technology Priyadarshini College of Engineering Nagpur Nagpur, India.

Miah, S.&Godder, T. K. To Design Voice Control Keyboard System using SpeechApplication Programming Interface. International Journal of Computer Science Issues, Vol. 7, Issue 6, November 2010. Department of Information and Communication Engineering, Islamic University, Kushtia, 7003, Bangladesh.

Potirakis,S. M., , Ganchev, T., Tuna, G.,Tatlas,N. A.&Zogo, R. 2012. Design Considerations of an Interactive Robotic Agent for Public Libraries. Department of Electronics Engineering, Technological Education Institute of Piraeus, Aigaleo, GreecebDepartment of Electronics, Technical University Varna, Bulgaria.

Rudzionis,A, Ratkevicius,K., Rudzionis,V., Maskeliunas, 2006. Examples of Lithuanian voice dialogue applications. R.Speech Research Group, Kaunas University of TechnologyKaunas Faculty of Humanities, Vilnius University. SPECOM'2006, St. Petersburg, 25-29 June 2006.

Sharma, K., Prasad, T.V. & Prasad, S. V. A. V. 2013. *Hindi speech enabled windows application using Microsoft SAPI*. International journal of computer engineering & technology (IJCET). Research Scholar, Dept. of CSE, Lingaya's University, Faridabad, Haryana, India.

Shi, H. & Maier, A. 2006. *Speech-enabled windows application using Microsoft SAPI*. International Journal of Computer Science and Network Security, VOL.6 No.9A, September 2006. Victoria University, Australia.

Sultana, S, Akhand, M. A. H., Das, P. K, Rahman, M. M. H. 2012. *Bangla Speech-to-Text Conversion using SAPI*. International Conference on Computer and Communication Engineering, 3-5 July 2012, Kuala Lumpur, Malaysia.

Herdianto. 2012. Penerapan *speech recognition library* untuk *query* Basis data. Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Bina Darma 2012.