Pembuatan Speech Recognition Dan Database Wicara Untuk Kontrol Peralatan Rumah Tangga Jarak Jauh

Didik Nurcahyono[1], Prima Kristalina[2], Miftahul Huda[2]
[1]Mahasiswa Politeknik Elektronika Negeri Surabaya, Jurusan Teknik Telekomunikasi
[2]Politeknik Elektronika Negeri Surabaya Institut Teknologi Sepuluh Nopember
Kampus ITS, Surabaya 60111
e-mail : dee2dee2@gmail.com e-mail : prima@eepis-its.edu, huda@eepis-its.edu

Abstrak

Kontrol peralatan rumah tangga jarak jauh di dalam 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 pemrogaman 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. Dan yang terakhir adalah me-*load* grammar dari dalam file yang telah dibuat dan diaktifkan menggunakan ISpRecoGrammar untuk mengenali kata yang berbeda.

Hasil pada proyek akhir ini menitik-beratkan pada rata-rata kualitas aplikasi yang bernila 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.

Kata kunci: *SAPI 5.1, Speech Recognition, grammar XML*

1. Pendahuluan

Seiring berkembangnya kompleksitas kehidupan manusia, menyebabkan karakteristik kehidupan manusia semakin memiliki mobilitas yang tinggi. Yang memungkinkan manusia untuk berkeinginan praktis yang memudahkan manusia untuk memenuhi kebutuhan dan kenyamanan hidupnya. Hal tersebut sangat menarik minat pakar teknologi untuk melakukan pengembangan riset pada sinyal wicara. Salah satunya adalah penelitian dalam bidang pengolahan sinyal wicara[1]. Teknologi pengolahan wicara telah banyak mengalami kemajuan yang sangat pesat. Hal itu dapat dibuktikan dengan adanya penelitian-penelitian tentang "intelligent machine". Sebuah mesin pintar yang menggunakan metode pengenalan ucapan pada manusia sehingga dapat berinteraksi dengan manusia.

Adanya komponen DCLSAPI51 memudahkan interaksi dari Borland Delphi 7.0 dengan SAPI SDK 5.1 sehingga proses speech recognition bisa dikerjakan menggunakan bahasa pemrogaman Delphi. Input yang akan diproses merupakan sinyal suara kemudian proses yang akan dihasilkan akan tampak dalam 3 parameter, yaitu speaker untuk dibunyikan, *display* untuk ditampilkan dan port paralel untuk dihubungkan dengan *interface* lain.

Teori penunjang paper ini berada pada sub bab 2, sementara itu sub bab 3 berisi tentang perancangan sistem secara umum dan bagian. Sub bab 4 merupakan hasil progress proyek akhir. Dan yang terakhir pada sub bab 5 membahas tentang rencana proyek akhir untuk tahapan kedepan.

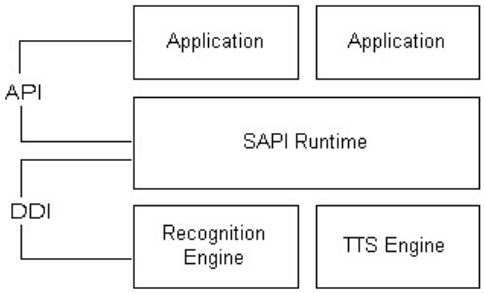
2. Teori Penunjang

Arsitektur Speech Application Programming Interface (SAPI 5.1) [10]

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

Aplication Programming Interface (API)

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



Gambar 1. Blok Diagram Arsitektur SAPI

Komponen SAPI yang akan menghasilkan even ini diimplementasikan oleh antar muka ISpNotifySource. Lebih spesifik, SAPI menggunakan SetNotifySink, yaitu aplikasi akan meneruskan pointer IspNotify- Sink ke ISpNotifySource:: SetNotifySink. ISpNotifySource ::SetNotify Sink ini akan menerima pemanggilan melalui IspNotify- Sink:: Notify ketika terdapat satu atau lebih even yang menyatakan bahwa aplikasi dapat mengambil data. Biasanya aplikasi tidak mengimplementasikan ISpNotifySink secara langsung tetapi menggunakan CoCreate Instance untuk membuat obyek IspNotify Sink, 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.

Ketika aplikasi menerima notifikasi, ada kemungkinan terdapat informasi yang sama pada beberapa even. Dengan memanggil ISpEventSource::GetInfo, maka variable anggota ulCount akan mengembalikan nilai yang berupa struktur SPEVENT SOURCEINFO yang didalamnya terdapat jumlah even yang mempunyai informasi yang sama. Dengan menggunakan IspEvent Source::GetEvents, aplikasi akan mengeluarkan sejumlah struktur SPEVENT, di mana masing-masing mempunyai informasi tentang even tertentu.

Ketika terjadi notifikasi pada saat pengenalan pembicaraan bekerja, maka lParam 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 even yang telah didefinisikan. Sebuah ISpRecoContext dapat mempunyai lebih dari satu ISpRecoGrammars di mana masing-masing ISpRecoGrammar tersebut digunakan untuk mengenali tipe yang berbeda.

Device Driver Interface (DDI)

DDI menyediakan fungsi untuk menerima data suara dari SAPI dan mengembalikan pengenalan frasa pada level SAPI paling dasar. Terdapat dua antar muka yang digunakan oleh DDI yaitu ISpSREngine, yang diimplementasikan oleh *engine* dan ISpSREngineSite yang diimplementasikan oleh SAPI.

Engine menyediakan layanan ke SAPI melalui antar muka ISpSREngine. Semua fungsi pengenalan terjadi melalui IspSREngine::RecognizeStream. Ketika SAPI memanggil ISpSREngine::SetSite, maka SAPI memberikan pointer ke antar muka ISpSREngineSite dimana kemudian *engine* dapat berkomunikasi dengan SAPI selama ISpSREngine::RecognizeStream dieksekusi. SAPI membuat sebuah *thread* ke obyek ISpSREngine dan *engine* tidak boleh meninggalkan ISpSREngine::Recognize Stream sampai terjadi kesalahan atau SAPI sudah terindikasi dengan menggunakan ISpSREngineSite::Read, dimana tidak ada lagi data yang dapat diproses dan *engine* telah selesai melakukan tugasnya.

SAPI memisahkan pembuat *engine* dari kerumitan untuk mengatur peralatan suara secara detail. SAPI menjaga *logical*

stream dari *raw audio data* dengan membuat indeks posisi *stream*. Dengan menggunakan indeks posisi *stream*, *engine* dapat melakukan pemanggilan terhadap ISpSREngineSite::Read untuk menerima *buffer* dari *raw audiodata* selama ISpSREngine::Recognize Stream dieksekusi. Pemanggilan ini akan terjadi sampai semua data yang dibutuhkan tersedia. Jika ISpSREngineSite::Read menghasilkan data yang lebih sedikit dari yang dibutuhkan, yang berarti tidak ada data lagi, maka *engine* akan menghentikan eksekusi ISpSREngine::RecognizeStream DDI memungkinkan *engine* untuk hanya mempunyai satu buah thread yang dieksekusi antara SAPI dan *engine*. Satu-satunya metode yang tidak mengijinkan Isp SREngine untuk masuk dan keluar secara cepat ialah ISpSREngine::Recognize Stream.

XML

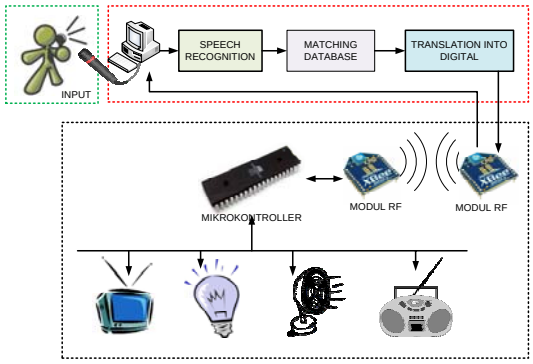
XML (*eXtensible Markup Language*) adalah sebuah bahasa markah untuk mendeskripsikan data. XML digunakan unt

- **Memulai XML**
  XML dalam format grammar, juga sama seperti format XML yang biasa, yaitu ada tag pembuka dan tag penutup.
- **Attribute**
  Atribut dari element XML dimunculkan di dalam tag pembuka. Setiap atribut mewakili sebuah susunan grammar dan untuk penulisannya diawali dengan kata LANGID yang harus diisi dengan *numeric value*, diikuti oleh tanda "sama dengan" (=) dan 2 tanda petik ganda sebagai pembuka dan penutup.
  Angka 809 merupakan nilai untuk memunculkan *English – American grammar* di dalam format XML.
- **Contents dan Coments**
  Merupakan isi yang terdiri atas *subelement* dan *text*. Untuk *coments* diawali dengan <!-- dan diakhiri dengan -->. Sebagai contoh bisa dilihat program di bawah ini. Untuk *contents* merupakan isi dari *grammar* XML tersebut.

3. Perancangan

Perencanaan Sistem Umum

Proyek akhir ini menggunakan sistem pengenalan suara untuk mengontrol peralatan rumah tangga jarak jauh. Proyek ini dibagi menjadi 2 bagian yaitu bagian perangkat lunak dan bagian perangkat keras, dan yang dikerjakan oleh peneliti adalah proses di dalam perangkat lunak sampai mengirimkan kode untuk masing-masing suara ke bagian perangkat keras (*microcontroller*) yang menggunakan *port paralel* seperti yang dilingkari pada gambar 2.

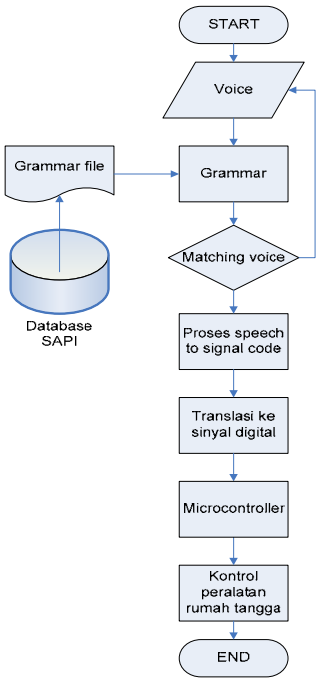


Gambar 2. Blok Diagram Sistem Umum

Sistem pengenalan suara yang dibuat disajikan dalam PC dan *microcontroller*. Pada saat aplikasi diaktifkan, *input* yang akan didapatkan merupakan sinyal suara pengguna. Sinyal suara tersebut akan diubah menjadi kata-kata (*speech-to-text*) yang akan diproses berdasarkan struktur *grammar* yang terhubung dengan database *Speech API*. Kemudian terjadilah proses *matching voice* untuk membedakan proses yang akan dilakukan untuk setiap kata tersebut.

Agar kata-kata ini bisa dibaca oleh microcontroller maka digunakan proses *speech to signal code* yang kemudian akan diubah menjadi bit-bit biner (translasi ke sinyal digital) dan dikirim datanya ke *parallel port* kemudian menuju ke

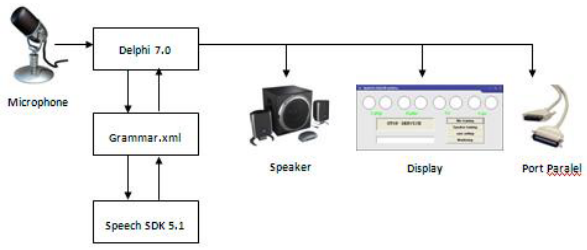
microcontroller. Data yang diterima kemudian diolah oleh *microcontroller* tersebut untuk kontrol peralatan rumah tangga sesuai dengan perintah yang diucapkan.



Gambar 3. Flowchart Sistem Umum

Perencanaan Sistem Bagian

Input yang akan diproses merupakan sinyal suara kemudian proses yang akan dihasilkan akan tampak dalam 3 parameter, yaitu speaker untuk dibunyikan, *display* untuk ditampilkan dan port paralel untuk dihubungkan dengan *interface* lain.

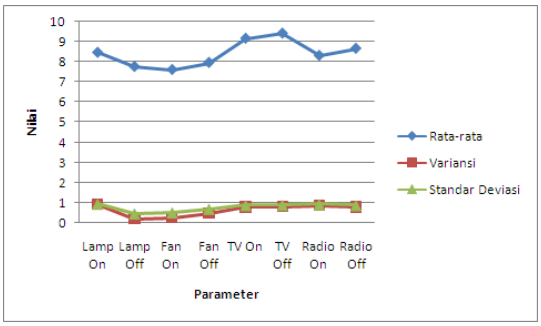


Gambar 4. Blok Diagram Sistem

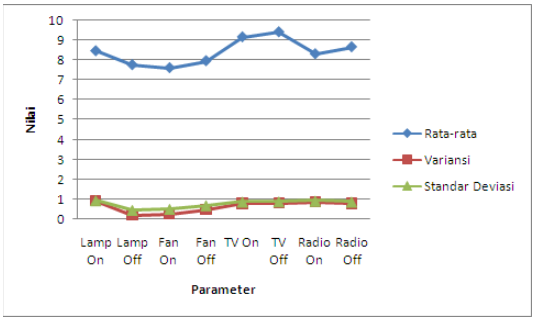
4. Pengujian dan Analisa

Kualitas Aplikasi Berdasarkan Jenis Kelamin

Pengujian yang dilakukan dititik-beratkan pada banyaknya penerimaan kata yang berhasil diproses sesuai konteks yang telah ditentukan, dengan beberapa kali pengulangan. Pengujian yang dilakukan *independent speaker* melibatkan 40 orang yang terdiri dari 20 orang laki-laki dan 20 orang perempuan. Konteks yang dimaksud adalah *"Lamp on"*, *"Lamp off"*, *"Radio on"*, *"Radio off"*, *"TV on"*, *"TV off"*, *"Fan on"* dan *"Fan off"*. Kemudian pengulangan yang dilakukan pada pengujian ini adalah sebanyak 10 kali.



Gambar 5 Grafik Rata-rata, Variansi dan Standar Deviasi pada Responden Laki-laki Menggunakan Sampel Laki-laki

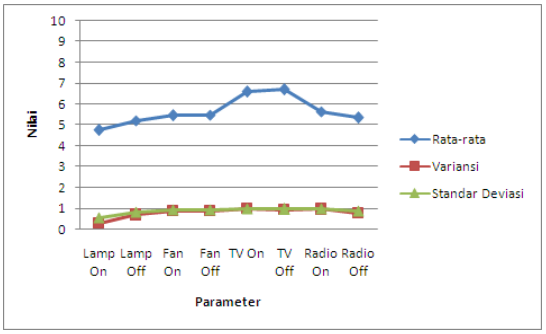


Gambar 6 Grafik Rata-rata, Variansi dan Standar Deviasi pada Responden Perempuan Menggunakan Sampel Perempuan

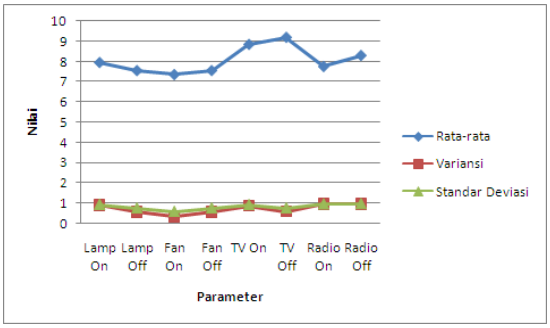
Kualitas Aplikasi Berdasarkan Usia

Pengujian yang dilakukan dititik-beratkan pada banyaknya penerimaan kata yang berhasil diproses sesuai konteks yang telah ditentukan, dengan beberapa kali pengulangan. Pengujian yang dilakukan *independent speaker* melibatkan 60 orang dengan usia dikelompokkan menjadi 3 macam yaitu usia anak-anak (6-14 tahun), usia kerja (15-64 tahun) dan usia lanjut (>65

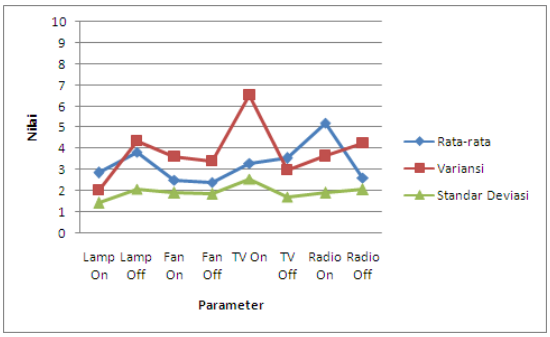
tahun)[13] yang masing-masing berjumlah 20 orang. Konteks yang dimaksud adalah *"Lamp on"*, *"Lamp off"*, *"Radio on"*, *"Radio off"*, *"TV on"*, *"TV off"*, *"Fan on"* dan *"Fan off"*. Kemudian pengulangan yang dilakukan pada pengujian ini adalah sebanyak 10 kali.



Gambar 7 Grafik Rata-rata, Variansi dan Standar Deviasi pada Responden Usia Anak-anak



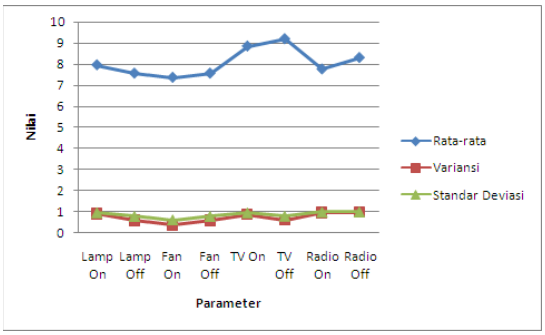
Gambar 8 Grafik Rata-rata, Variansi dan Standar Deviasi pada Responden Usia Kerja



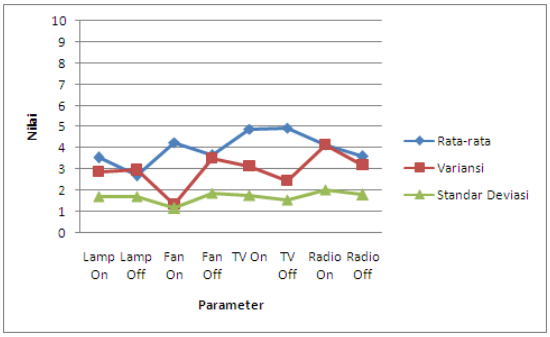
Gambar 9 Grafik Rata-rata, Variansi dan Standar Deviasi pada Responden Usia Lanjut

Kualitas Aplikasi Pada Kondisi *Indoor* dan *Outdoor*

Pengujian yang dilakukan dititik-beratkan pada banyaknya penerimaan kata yang berhasil diproses sesuai konteks yang telah ditentukan, dengan beberapa kali pengulangan. Pengujian yang dilakukan *independent speaker* melibatkan 20 orang yang sama yang ditempatkan pada lokasi *indoor* dan *outdoor*.



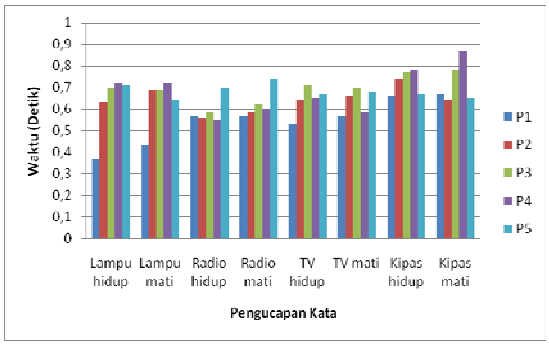
Gambar 10 Grafik Rata-rata, Variansi dan Standar Deviasi Responden pada Lokasi Indoor



Gambar 11 Grafik Rata-rata, Variansi dan Standar Deviasi Responden pada Lokasi Outdoor

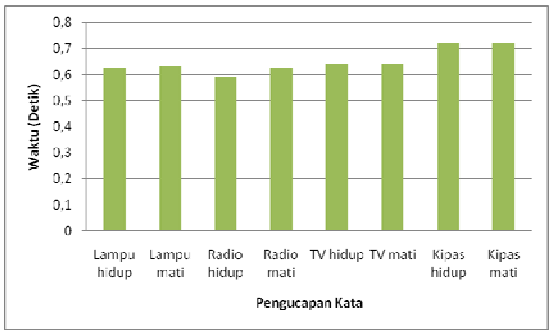
Respon Program

Respon program diambil dengan menggunakan stopwatch yang diukur mulai dari pengucapan kata sampai dengan respon nyala LED di tampilan aplikasi. Pengujian dilakukan untuk masing-masing kata pengucapan yang sudah ditentukan yang melibatkan responden yang memiliki kualitas paling bagus dan menggunakan satuan detik. Pengujian dilakukan sebanyak 5 kali pengulangan.



Gambar 8 Grafik Respon Program Pada Masing-masing Kata

Lalu dilakukan pengambilan rata-rata dengan menjumlahkan semua nilai waktu pada masing-masing kata dibagi dengan jumlah pengulangan yang dilakukan pada pengujian.



Gambar 9 Grafik Rata-rata Respon Program Pada Masing-masing Kata

Rata-rata waktu yang telah dihasilkan pada pengujian ini membuktikan bahwa untuk "Kipas mati" dan "Kipas Hidup" berada pada waktu yang paling lama dalam proses respon suara dengan nilai 0,722 dan 0,724 detik. Sebaliknya untuk respon paling cepat dimiliki oleh kata "Radio hidup" dengan nilai 0,594 detik.

5. Kesimpulan

Berdasarkan pada hasil pengujian dan analisa terhadap hasil yang didapatkan, maka dapat diambil suatu kesimpulan yaitu :

1. Kualitas aplikasi pada saat pengujian dipengaruhi oleh beberapa faktor yaitu jenis kelamin, usia, dan *noise*.
2. Aplikasi tidak bisa memproses grammar yang mempunyai struktur kata yang terdiri dari 1 kata saja dikarenakan tidak mempunyai value untuk diproses.
3. Respon program dipengaruhi oleh letak kata baris program di dalam kondisi pada masing-masing parameter kata yang telah ditentukan.
4. Semakin banyak kata sama pada saat pengambilan sampel memungkinkan aplikasi merespon suara yang masuk lebih baik untuk tiap pengucapan yang berbeda.
5. Kualitas aplikasi yang baik ditentukan oleh koefisien variansi dengan nilai antara 0 sampai dengan 1.

Daftar Pustaka

[1] http://en.wikipedia.org/wiki/Speech_recognition

[2] Andika, I., *Aplikasi Speech Recognition Untuk Penyajian Informasi Kereta Api di Stasiun Gubeng*, PENS-ITS, Surabaya, 2007

[3] http://www.informatika.org/~rinaldi/Stmik/2007-2008/Makalah2008/MakalahIF2251-2008-077.pdf

[4] http://indotts.melsa.net.id/Karakteristik%20Sinyal%20Ucapan

[5] http://www.**code**project.com/KB/audio-video/tambiSR.aspx

[6] John. G. Proakis, Dimitris. G. manolakis, "Digital Signal processing: principles, algorithms, and Application", Prentice Hall, Inc, New jersey, 1995.

[7] Lawrence Rabiner, Biing Hwang Juan, "Fundamentals of Speech Recognition", Prentice Hall International Inc, 1993.

[8] Sadaoki Furui, "Digital Speech Processing, Synthesis, and, recognition", marcel dekker, Inc, New York, 1989

[9] http://indomicron.co.cc/komputer/delphi/belajar-microsoft-speech-api-delphi-5-6-7/comment-page-1/#comment-111

[10] http://puslit2.petra.ac.id/ejournal/index.php/inf/article/viewFile/15838/15830
[11] Microsoft SAPI SDK 5.1 chm file
[12] Microsoft Website. http://microsoft.com/speech/techinfo/apioverview/
[13] http://www.datastatistik-indonesia.com/content/view/920/936/