

Rancang Bangun Alat Pendeteksi Suara Panggilan Manusia Berbahasa Indonesia Untuk Tunarungu Menggunakan Metode *Hidden Markov Model* (HMM) Berbasis *Embedded system*

M. Wildan Alauddin¹, Wijaya Kurniawan, S.T, M.T.², Budi Darma Setiawan, S.Kom, M.Cs.³

Program Studi Teknik Komputer
Fakultas Ilmu Komputer
Universitas Brawijaya, Malang
Jl. Veteran No 8, Malang 65145, Indonesia
E-mail: mesutwildan@gmail.com

ABSTRAK

Komunikasi adalah salah satu hal terpenting dalam kehidupan manusia. Namun, tidak semua manusia dapat menggunakan indera pendengarannya secara sempurna. Manusia yang mengalami disabilitas indera pendengaran disebut sebagai tunarungu. Disabilitas tunarungu dapat disebabkan oleh banyak hal antara lain faktor keturunan dan kecelakaan.

Penyembuhan indera pendengaran bukanlah satu-satunya hal yang efektif sebagai solusi atas permasalahan ini. Hal ini disebabkan oleh tidak semua disabilitas pendengaran dapat disembuhkan, terutama akibat faktor keturunan. Selain itu, biaya untuk penyembuhan sangat mahal sehingga tidak semua penyandang tunarungu dapat melakukannya.

Atas beberapa pertimbangan tersebut penulis melakukan sebuah penelitian memanfaatkan teknologi untuk mengatasi permasalahan komunikasi antara manusia normal dengan para penyandang disabilitas tunarungu. Penulis mengaplikasikan sebuah mini PC bertipe *Raspberry Pi 2 B+* sebagai inti dari sistem yang akan dirancang memanfaatkan *library* PocketSphinx sebagai perangkat lunak utama dalam proses konversi ucapan menjadi tulisan (*speech to text*) agar para tunarungu dapat membaca tulisan sebagai output sehingga mereka tetap dapat berkomunikasi. Sistem tersebut akan dirancang dengan menggunakan mini *microphone* sebagai sensor utama dan metode *Hidden Markov Model* (HMM) untuk diterapkan oleh sistem dalam mengenali karakteristik pengucapan manusia.

Pada penelitian kali ini, peneliti dapat menilai bahwa penggunaan mini PC dapat memungkinkan sistem dapat digunakan secara *mobile*. Namun diperlukan mini PC dengan spesifikasi lebih tinggi untuk meningkatkan performa sistem dan mempercepat eksekusi program dan *library* yang digunakan. Hal ini juga berlaku untuk penggunaan *mini microphone* agar sistem dapat mendeteksi sumber suara yang lebih jauh dan daya tangkap yang lebih tinggi.

Kata Kunci: Tunarungu, *Raspberry*, PocksetSphinx, *Hidden Markov Model*.

1. Pendahuluan

Saat ini jumlah permasalahan yang timbul akibat kekurangpedulian masyarakat semakin banyak. Salah satu contohnya ketika kita mengetahui jumlah sukarelawan pendamping tunarungu yang sedikit dan minimnya masyarakat Indonesia yang mampu memahami bahasa isyarat. Hal ini semakin meminimalisir peluang komunikasi antara manusia yang normal dengan manusia berkebutuhan khusus atau biasa kita sebut dengan penyandang disabilitas, khususnya penyandang tunarungu.

Melalui situs resminya (<http://www.who.int/pbd/deafness/estimates/en/>), Organisasi Kesehatan Dunia atau *World Health Organization* (WHO) merilis data hasil survei pada tahun 2012 tentang jumlah kasus disabilitas gangguan pendengaran. Survei tersebut didasarkan pada *review* dari 42 studi kasus berbasis sampel populasi yang dilakukan hingga 2010. Data hasil survei tersebut menyatakan bahwa ada 360 juta

orang di dunia dengan disabilitas gangguan pendengaran (5,3% dari populasi dunia), 328 juta (91%) di antaranya adalah orang dewasa (183 juta laki-laki, 145 juta perempuan) 32 (9%) juta tersebut adalah anak-anak.

Menurut hasil Survei Sosial Ekonomi Nasional (Susenas) yang dilaksanakan Biro Pusat Statistik (BPS) tahun 2012, jumlah penyandang disabilitas di Indonesia sebanyak 6.008.661 orang. Dari jumlah tersebut sekitar 1.780.200 orang adalah penyandang disabilitas netra, 472.855 orang penyandang disabilitas rungu wicara, 402.817 orang penyandang disabilitas grahita/intelektual, 616.387 orang penyandang disabilitas tubuh, 170.120 orang penyandang disabilitas yang sulit mengurus diri sendiri, dan sekitar 2.401.592 orang mengalami disabilitas ganda.

Banyak teknologi sudah dikembangkan untuk membantu para penyandang disabilitas tunarungu agar dapat mengetahui kondisi lingkungan di sekitarnya tanpa harus melakukan pengobatan atau

operasi terhadap indera pendengarannya. Beragam teknologi yang sudah dikembangkan tersebut mengimplementasikan banyak metode pengolahan data suara sebagai *input* menjadi *output* bentuk lain, salah satu metode tersebut adalah *Hidden Markov Model*. *Hidden Markov Model* (HMM) merupakan model statistik di mana suatu sistem yang dimodelkan diasumsikan sebagai markov process dengan kondisi yang tidak terobservasi. Suatu HMM dapat dianggap sebagai jaringan Bayesian dinamis yang sederhana. (Prasetio, 2010).

Pengembangan teknologi yang mengimplementasikan *Hidden Markov Model* sudah sering kita jumpai di kehidupan sehari-hari, khususnya yang memanfaatkan salah satu dari *library*-nya, yaitu *library PocketSphinx*. *Library* ini mendukung implementasi fitur *speech recognition* guna mendeteksi suara panggilan manusia kepada pengguna sistem ini. Untuk mengoptimalkan kinerja *library* ini, penulis berencana menerapkannya pada *Raspberry Pi* dengan bantuan Sistem Operasi Raspbian yang digunakan dalam Mini PC *Raspberry*.

Berdasarkan uraian permasalahan yang telah dipaparkan di atas, dan dengan perkembangan teknologi yang ada serta ketersediaan perangkat keras untuk mengimplementasikan teknologi tersebut, diharapkan bisa menjadi solusi dari masalah yang telah dipaparkan pada paragraf sebelumnya. Oleh karena itu disini penulis mengangkat sebuah skripsi dengan judul “Rancang Bangun Alat Pendeteksi Suara Panggilan Manusia Berbahasa Indonesia Untuk Tunarungu Menggunakan Metode *Hidden Markov Model* (HMM) Berbasis *Embedded system*”. Sistem ini dirancang berdasarkan pengukuran dan analisa untuk mengetahui kelayakannya serta dapat memberikan acuan untuk penerapan dalam studi kasus sehingga fitur *speech recognition* sebagai pendeteksi suara panggilan manusia berbahasa Indonesia ini dapat diimplementasikan dengan baik.

2. Dasar Teori

Dasar teori yang dibutuhkan untuk penyusunan skripsi ini adalah pengenalan ucapan, sistematika pengenalan ucapan, metode *Hidden Markov Model* (HMM), *PocketSphinx* dan *Raspberry Pi*.

2.1 Pengenalan Ucapan

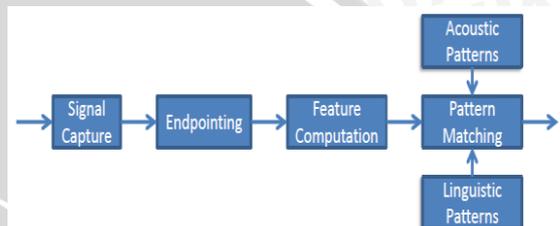
Pengenalan suara merupakan kemampuan sebuah alat/komputer dalam menerjemahkan suara menjadi bentuk teks ataupun komando. Teknologi ini mempunyai fasilitas untuk mengubah *input* suara kedalam rangkaian kosakata dalam bentuk sinyal-sinyal suara lalu menghasilkan probabilitas model statistik untuk menemukan persamaan terhadap *input* suaranya (Prasetyo, 2010).

Pada umumnya pengenalan ucapan digunakan untuk mengartikan ucapan seseorang ke dalam bentuk kata secara tertulis misalnya untuk

penggunaan *voice control* pada *smarthome system* maupun sebagai sumber informasi untuk komunikasi kaum lanjut usia dan disabilitas.

2.2 Sistematika Pengenalan Ucapan

Sistem pengenalan ucapan (*voice recognition*) adalah sebuah rangkaian teknik yang memungkinkan sebuah sistem komputer untuk menerima *input* berupa suara lisan untuk kemudian diubah menjadi bahasa tulisan (*speech to text*). Selanjutnya sistem akan mengidentifikasi kata atau kalimat yang diucapkan dan menghasilkan teks yang sesuai dengan apa yang diucapkan. Pada gambar 2.1 merupakan alur system pengenalan ucapan.

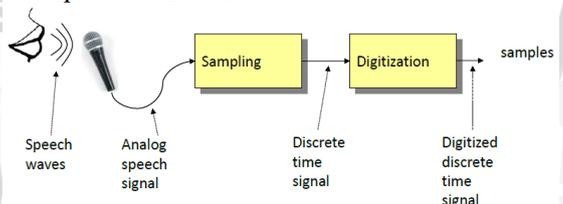


Gambar 2.1 Alur Sistem Pengenalan Suara

Sumber: I Kadek Suryadharma (2014)

2.2.1 *Speech Signal Capture* (Penangkapan Sinyal Ucapan)

Langkah ini merupakan langkah awal dalam *speech recognition*. Suara dihasilkan oleh saluran vokal manusia berupa serangkaian gelombang yang mampu didengar oleh telinga pendengar. Secara umum dapat dilihat pada gambar 2.2 berikut:



Gambar 2.2 Langkah Awal Penangkapan Suara

Sumber: I Kadek Suryadharma (2014)

Sinyal suara yang masuk berupa sinyal analog. Sampling dilakukan untuk mencuplik sinyal analog menjadi bit-bit sinyal analog diskrit yang nantinya memudahkan dalam pemrosesan dan hasilnya berupa sampel-sampel bilangan biner (sinyal digital) yang merupakan informasi dari sinyal asli.

2.2.2 *Endpointing*

Pada langkah ini digunakan untuk mengidentifikasi bagaimana hasil sinyal suara yang sudah di capture tadi dapat diproses. Misalnya *press and speak* (tekan dan bicara). Ini dilakukan agar dapat menghindari suara-suara yang tidak diinginkan masuk ke sistem saat pengenalan suara.

2.2.3 *Feature Extraction*

Feature extraction (ekstraksi ciri) merupakan suatu pengambilan ciri / feature dari suatu sinyal informasi yang nantinya nilai yang didapatkan akan dianalisis untuk proses selanjutnya. Setiap informasi memiliki ciri yang berbeda (unik). Prinsip kerja ekstraksi ciri adalah dengan mengkonversi sinyal suara ke dalam beberapa parameter, dimana ada sebagian informasi tidak berguna yang dibuang tanpa menghilangkan arti sesungguhnya dari sinyal suara tersebut. Hasil keluaran dari ekstraksi ciri ini menjadi masukan pada proses pengenalan pola.

Ekstraksi ciri yang akan digunakan pada penelitian ini ialah *Mel-Frequency Cepstral Coefficient* (MFCC). MFCC merupakan salah satu metode ekstraksi ciri yang banyak digunakan dalam bidang *speech technology*, baik *speaker recognition* maupun *speech recognition* dengan mengkonversikan signal suara menjadi beberapa parameter.

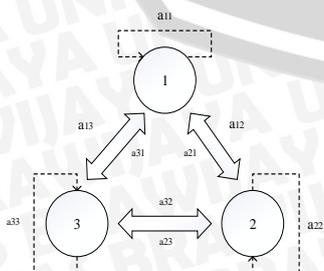
2.2.4 Matching

Matching atau pencocokan ini merupakan proses akhir pada *speech recognition*. Hasil dari ekstraksi ciri menjadi masukan pada proses pengenalan pola ini. Metode yang digunakan dalam pengenalan pola ialah metode *Hidden Markov Model* (HMM). Pola yang didapat akan dicocokkan dengan berbagai macam model. Ada 3 jenis model yang umum digunakan pada *speech recognition* yakni *acoustic models*, *pronunciation models* dan *language models*.

2.3 Hidden Markov Model (HMM)

Model Markov ditemukan oleh Andrey Markov dan merupakan bagian dari proses stokastik. Model ini memiliki kehandalan untuk memprediksi keadaan yang akan datang artinya kondisi saat ini menangkap semua informasi yang mempengaruhi evolusi dari suatu sistem dimasa depan.

Model ini merupakan dari *finite automation* yaitu beberapa kumpulan state yang transisi antar state-nya dilakukan berdasarkan masukan observasi. Pada proses Markov, setiap busur antar state berisi probabilitas yang mengindikasikan kemungkinan dari jalur tersebut. Jumlah probabilitas semua jalur yang keluar dari sebuah simpul memiliki nilai total satu.



Gambar 2.3 Rantai Markov
Sumber: Monika, 2012

Pada Gambar 2.3, misalkan dari simpul nomer 1 dan 2 mempunyai kemungkinan a21 dan a12. Maka jumlah probabilistic a12 dan a21 memiliki nilai satu. Hal ini juga berlaku pada simpul lainnya. Proses Markov mempunyai manfaat untuk menghitung probabilitas suatu peristiwa yang diamati dan secara umum dapat dirumuskan sebagai berikut:

$$P(q_1)[q_t = j | q_{t-1} = i, q_{t-2} = k \dots] = P[q_t = j | q_{t-1} = i] \dots \dots \dots (2-1)$$

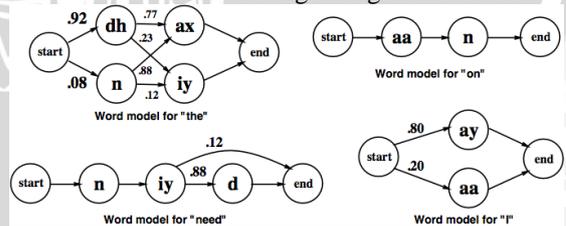
q1 adalah kondisi pada saat ini, dan qt adalah kondisi pada satu waktu tertentu yang berhubungan erat dengan q1. Sedangkan qt-1 adalah kondisi sebelum qt. Selanjutnya kita dapat menyimpulkan bahwa keseluruhan dari sistem, transisi di antara keadaan tertentu tetap sama dengan hubungan probabilitiknya.

Berdasarkan persamaan pada paragraf sebelumnya, kita dapat membuat persamaan lain di mana persamaan tersebut memiliki transisi keadaan bebas terhadap waktu.

$$A_{ij} = P[q_t = j | q_{t-1} = i] \dots \dots \dots (2-2)$$

Lahirnya metode *Hidden Markov Model* menjadi salah satu pemicu pembaruan teknik pengolahan *speech* (ucapan / suara) yang dapat menghasilkan tingkat kesalahan seminimal mungkin dan meningkatkan ketangguhan pengklasifikasian pola pada berbagai kondisi yang kurang baik.

Penerapan HMM pada pengenalan suara digunakan untuk memprediksi suara yang diucapkan berdasarkan dictionary file, language model, dan acoustic modelnya. HMM akan memberikan probabilitas berdasarkan kosakata yang akan diucapkan. Probabilitas kemudian dihitung berdasarkan pembobotannya sehingga mempunyai nilai terbaik yang memungkinkan kata tersebut dikeluarkan berdasarkan persamaan suara. Pencarian probabilitas terbaik dari masing-masing suku kata diselesaikan dengan Algoritma Viterbi.



Gambar 2.4 Pencarian Kata Terbaik dengan HMM

<http://cmusphinx.sourceforge.net/wiki/tutorialPocketsphinx/>

Perbedaan model kata "the" dengan "need" tergantung kesamaan suku kata yang dimiliki kata tersebut di dalam dictionary file. Apabila kata pada dictionary file memiliki suku kata yang hampir sama dengan kata yang lain maka percabangan HMM akan semakin banyak sehingga memiliki banyak kemungkinan dari pembentukan kosakata berdasarkan suku katanya. Jika masalah ini terjadi



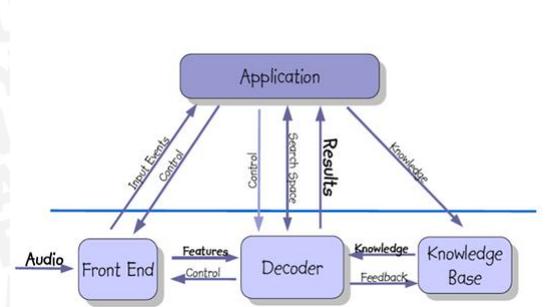
HMM hanya dapat memprediksi kemungkinan berdasarkan pola suara yang paling sering diucapkan oleh seseorang.

2.4 PocketSphinx

PocketSphinx ialah sistem pengenalan suara tercepat yang dikembangkan oleh Carnegie Mellon University (CMU). *PocketSphinx* menggunakan bahasa pemrograman C murni dan sangat optimal karena bersifat *real-time* sehingga mesin dapat mengenali suara dengan akurat. Dengan mengandalkan respon yang sangat cepat dan konsumsi sumber daya yang minim sehingga memungkinkan pengembangan pengenalan suara cocok digunakan untuk aplikasi *desktop*, komando dan kontrol serta pendidikan. Selain itu, *engine* ini dapat berguna bagi perangkat tertanam dengan *fixed-point arithmetics* yang berhasil digunakan pada iPhone, perangkat Nokia Maemon dan Windows Mobile (Anonymous, 2014).

Library ini menyediakan fasilitas untuk para penggunaannya untuk memodifikasi bahasa yang akan digunakan untuk mengenali suara seseorang. Aplikasi *PocketSphinx* merupakan sistem pengenalan suara yang biasa digunakan untuk aplikasi *desktop*. Pada saat ini, *PocketSphinx* biasa digunakan untuk kebutuhan *embedded system* dan robot karena *library* didukung memiliki kelebihan komputasi yang cepat dan ringan dalam mengenali suara seseorang.

2.4.2 Arsitektur Sphinx



Gambar 2.5 Arsitektur Sphinx

Sumber: <http://cmusphinx.sourceforge.net>

Gambar 2.5 menunjukkan arsitektur umum yang digunakan oleh Sphinx. Setiap elemen pada gambar ini dapat diganti sesuai dengan kebutuhan peneliti. Dengan demikian, kita dapat mengubah beberapa fitur tanpa mengubah sisi fungsionalitasnya secara umum.

Arsitektur umum aplikasi ini terdiri komponen umum untuk menunjang hasil yang presisi pada saat pengujian aplikasi. Komponen tersebut terdiri dari *front end*, *decoder*, *knowledge base*, dan aplikasi itu sendiri. *Front end* bertanggung jawab untuk mengumpulkan, memberikan keterangan dan pengolahan *input* data. Selain itu, *front end* juga menyediakan kemudahan untuk pengaksesan API audio yang merupakan fitur untuk merekam masukan pengguna ucapan dan memutar hasil

rekaman tersebut. Pada *knowledge base* memberikan sebuah informasi decoder untuk melakukan tugasnya. Informasi tersebut mencakup model akustik dan model bahasa. Basis pengetahuan pula yang memberikan umpan balik dari decoder sehingga memungkinkan basis pengetahuan dapat dimodifikasi secara dinamis. Sedangkan pada decoder sendiri berfungsi sebagai komponen utama pada aplikasi pengenalan suara. Decoder berfungsi untuk menentukan urutan yang paling mungkin dari kata-kata yang dapat diwakili oleh serangkaian fitur dan dilakukan secara dinamis selama proses *testing/decoding*. (Anonymous, 2014)

2.4.3 Language Model

Language model merupakan pembangkit grammar kosakata pada aplikasi pengenalan suara. Kompleksitas grammar tergantung pada sistem yang akan dikembangkan. Dalam penelitian kali ini, *language model* dibangun berdasarkan statistik yang digunakan oleh *library* *PocketSphinx*. *Toolkit language model* dibuat berdasarkan pemodelan uni-gram, bi-gram, dan tri-gram dari bahasa yang akan dikenali. Penciptaan *language model* terdiri dari komputasi kata jumlah uni-gram yang kemudian diubah menjadi kosakata dengan frekuensinya sehingga nantinya dapat menghasilkan bi-gram dan tri-gram dari teks pelatihan dan akhirnya mengubah n-gram ke dalam format biner *language model* dan format ARPAbet. Pada penelitian kali ini, *language model* dibuat dengan cara menuliskan data artikel yang ingin dikenali oleh aplikasi *voice command* menggunakan notepad lalu mengunggahnya ke server CMU Sphinx.

2.4.4 Acoustic Model

Acoustic model merupakan fasilitas yang diberikan oleh *PocketSphinx* untuk mengenali ciri suara pengguna. *Acoustic model* menggunakan *Hidden Markov Model* dan *Gaussian Mixture* sebagai metode dalam menentukan sinyal suara yang disesuaikan dengan fonem seseorang. *Hidden Markov Model* pada *acoustic model* merepresentasikan beberapa parameter berupa *hidden state* yang berupa *input* suara, *observed state* berupa fitur vektor *input* suara, dan probabilitas transisi yang berupa nilai probabilitas terbaik untuk pengucapan suara sehingga dapat membangkitkan ciri suara yang digunakan pada aplikasi.

Pelatihan *acoustic model* dilakukan dengan cara merekam suara yang disesuaikan dengan transcription file dari setiap kalimat yang diucapkan dalam rekaman suara. *Input* *acoustic model* berupa rekaman suara dengan format .wav. Setiap file berisikan satu kalimat dari satu pembicara. Selanjutnya rekaman suara tersebut disesuaikan dengan transcription file. Pedoman pada transcription file merujuk dictionary file. Filler file yang berisikan suara non-speech yang

bertujuan memberikan jeda pada sebelum kalimat mulai dan berakhir diucapkan (Anonymous, 2014).

2.4.5 Dictionary File

Dictionary file berisikan kosakata yang digunakan dalam teks artikel dan cara-cara pengucapan tiap kosakata yang terdiri dari fonem-fonem yang dapat disesuaikan dengan model bahasa yang akan digunakan. Pada keadaan awal, teks artikel diupload ke *server* CMU Sphinx menggunakan web browser dengan alamat <http://www.speech.cs.cmu.edu/tools/lmtool-new.html>. Setelah file tersebut diunggah, maka akan dihasilkan *dictionary file* yang berisikan kosakata yang dapat diucapkan oleh pengucapan bahasa Inggris. Oleh karena itu, fonem-fonem harus disesuaikan dengan gaya pengucapan orang Indonesia (Noora, 2010).

Dictionary file menjadi pedoman utama untuk pengenalan suara yang akan dirancang karena kosakata yang diucapkan terbatas sesuai dengan kosakata yang ada di *dictionary file*. Penulisan kosakata pada *dictionary file* menggunakan peraturan Arpabet. Arpabet adalah transkripsi fonetik yang dikembangkan oleh *Advanced Research Projects Agency* (ARPA) sebagai bagian dari *Speech Understanding Project* (1971-1976). Arpabet mewakili setiap fonem dari dalam bahasa Inggris Amerika umum (*General American English*) dengan urutan yang berbeda dari karakter ASCII (ICCE, 2010).

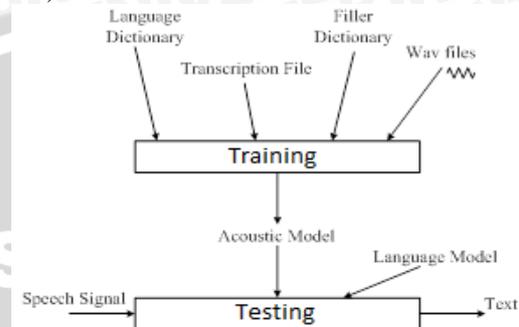
2.4.6 Training

Training merupakan upaya pelatihan suara untuk membangkitkan kosakata yang akan diucapkan oleh seseorang. *Training* berfungsi sebagai pengenalan ciri suara manusia sehingga nantinya suara dapat dikenali dengan baik. *Training* akan mempelajari model unit suara menggunakan kumpulan sampel sinyal suara pada database *training* berdasarkan *transcription file*. Selain *transcription file*, dibutuhkan juga kamus pengucapan (*dictionary file*) yang memetakan setiap kata ke urutan unit suara. Hasil *training* menghasilkan akurasi rekaman suara yang telah dipetakan pada *database training*, hasil akurasi yang terbaik dapat didapatkan apabila nilai parameter-parameter pada saat *training* digunakan secara maksimal. Setelah proses *training* berakhir dengan menghasilkan hasil akurasi yang baik lalu peneliti dapat mengambil file *acoustic model* yang akan dapat digunakan untuk membantu testing pengenalan suara (Monika, 2012). Proses *training* suara menggunakan *library* dari CMU Sphinx yaitu SphinxTrain. *Training* membantu untuk membangkitkan probabilitas suatu kosakata, apabila kosakata sering dilatih maka akurasi kosakata yang diucapkan pada saat testing akan semakin baik.

2.4.7 Testing

Testing merupakan percobaan pengenalan suara dengan memberikan executable tunggal yang

dapat melakukan tugas pengenalan suara saat diberi *input* suara. File-file yang dibutuhkan pada saat melakukan testing ialah *acoustic model*, *dictionary file*, dan *language model*. Testing dapat dilakukan dengan baik apabila proses pada saat *training* dilakukan dengan baik untuk mengenali pola suara orang Indonesia. Pada percobaan testing menggunakan *library* PocketSphinx (Monika, 2012).



Gambar 2.6 Hubungan *Training* dan *Testing*
Sumber: Eko Budi Prasetyo, 2010

2.5 Raspberry Pi



Gambar 2.7 Raspberry Pi 2B+

Salah satu mini PC yang paling populer di pasaran adalah *Raspberry Pi*. Pada penelitian kali ini penulis akan memanfaatkan *Raspberry Pi 2* untuk merealisasikan rancangan sistem yang telah dibuat sebelumnya. Memiliki bentuk dan ukuran yang sama dengan *Raspberry Pi 1 B+*, *Raspberry Pi 2* dilengkapi dengan *Random Access Memory* (RAM) sebesar 1 GB disertai peningkatan prosesor dari BCM2835 (single core ARMv6) ke BCM2836 (quad core ARMv7) yang meningkatkan performa mini PC ini menjadi hampir 2 kali lipat.

3. Perancangan dan Implementasi

3.1 Gambaran Umum Sistem

Secara umum, sistem memiliki sebuah *microphone* sebagai sensor pendeteksi suara yang kemudian dihubungkan ke *Raspberry* melalui port USB dengan perantara *soundcard*.

3.1.1 Building PocketSphinx

Building PocketSphinx

```
cd ~/  
wget http://sourceforge.net/projects/cmusphinx/files/pocketsphinx/Sprealpha/pocketsphinx-  
Sprealpha.tar.gz  
tar -zxvf pocketsphinx-Sprealpha.tar.gz  
cd ./pocketsphinx-Sprealpha  
./configure  
make clean all  
make check  
sudo make install
```

Gambar 3.1 Building PocketSphinx

PocketSphinx merupakan versi ringan dari Sphinx. Library mengimplementasikan HMM dalam pengaplikasiannya.

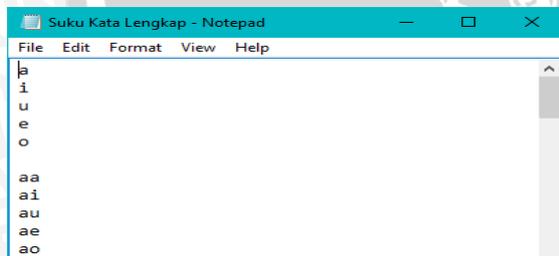
3.1.2 Building SphinxBase

```
Building Sphinxbase  
cd ~/  
wget http://sourceforge.net/projects/cmusphinx/files/sphinxbase/Sprealpha/sphinxbase-  
Sprealpha.tar.gz  
tar -zxvf ./sphinxbase-Sprealpha.tar.gz  
cd ./sphinxbase-Sprealpha  
./configure --enable-fixed  
make clean all  
make check  
sudo make install
```

Gambar 3.2 Building SphinxBase

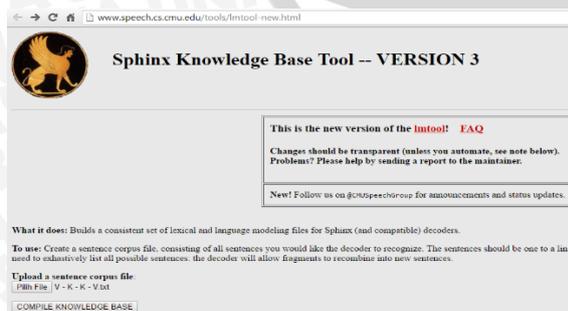
SphinxBase diperlukan untuk menyimpan log seluruh *state* dan *parameter* yang pernah digunakan oleh sistem pada proses sebelumnya untuk mempercepat proses apabila *state* dan *parameter* diakses kembali.

3.1.3 Unggah Dictionary



Gambar 3.3 Daftar Suku Kata

Daftar kata atau suku kata yang ingin dikenali sebagai *input* dari sistem kita nantinya terlebih dahulu kita sertakan dalam sebuah file .txt untuk diunggah ke <http://www.speech.cs.cmu.edu/tools/lmtool-new.html> untuk dikompilasi.

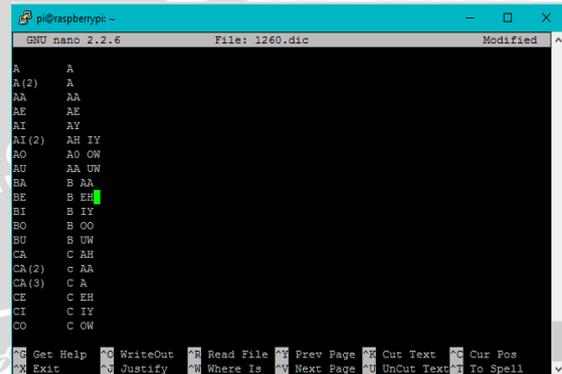


Gambar 3.4 Tampilan Awal Sphinx Knowledge Base Tool

Setelah kompilasi selesai dilakukan, maka *dictionary file* dan *language model* dapat diunduh di

http://www.speech.cs.cmu.edu/tools/product/1467152359_28133/4863.dic dan di http://www.speech.cs.cmu.edu/tools/product/1467152359_28133/4863.lm

Kemudian lakukan adaptasi fonem yang ada di *dictionary file* agar fonem yang semula secara *default* dalam bahasa Inggris dapat disesuaikan menjadi fonem bahasa Indonesia.



Gambar 3.5 Penyesuaian Fonem

Hal ini dilakukan sebagai batasan permasalahan user dan koresponden yang dapat menggunakan sistem ini hanya dalam bahasa Indonesia.

4. Pengujian

4.1 Skenario Pengujian

Skenario pengujian sistem ini ada 3 macam, yaitu pengujian *microphone*, pengujian jarak dan pengujian tingkat kerumitan ejaan nama pengguna sistem.

4.2 Hasil Pengujian

Pengujian pertama dilakukan dengan menguji sistem agar dapat mendeteksi nama "Wildan" pada jarak 1 meter dari sensor menggunakan Sony Mini Microphone dan TOA Microphone Jepit.

Pengujian 1

	1	2	3	4	5	6	7	8	9	10	Persentase Keberhasilan
L1	WI IL DA AN	WI DA	WI IL DA AN	90%							
L2	WI IL DA AN	100%									
L3	WI IL DA AN	100%									
L4	WI IL DA AN	90%									
L5	WI IL DA AN	100%									
P1	WI IL DA AN	*****	WI IL DA AN	90%							
P2	WI IL DA AN	*****	WI IL DA AN	90%							
P3	WI IL DA AN	*****	90%								
P4	WI IL DA AN	100%									
P5	WI IL DA AN	100%									
Rata-rata Keberhasilan											
94%											

Gambar 4.1 Hasil Pengujian 1

Pengujian 1											
	1	2	3	4	5	6	7	8	9	10	Persentase Keberhasilan
L1	WI IL DA AN	WI DA	WI IL DA AN	90%							
L2	WI IL DA AN	100%									
L3	WI IL DA AN	100%									
L4	WI IL DA AN	90%									
L5	WI IL DA AN	100%									
P1	WI IL DA AN	90%									
P2	WI IL DA AN	*****	WI IL DA AN	90%							
P3	WI IL DA AN	90%									
P4	WI IL DA AN	100%									
P5	WI IL DA AN	100%									
Rata-rata Keberhasilan											94%

Gambar 4.2 Hasil Pengujian 2

Pengujian 3											
	1	2	3	4	5	6	7	8	9	10	Persentase Keberhasilan
L1	WI IL DA AN	100%									
L2	WI IL DA AN	100%									
L3	WI IL DA AN	100%									
L4	WI IL DA AN	90%									
L5	WI IL DA AN	100%									
P1	WI IL DA AN	100%									
P2	WI IL DA AN	*****	WI IL DA AN	90%							
P3	WI IL DA AN	90%									
P4	WI IL DA AN	100%									
P5	WI IL DA AN	100%									
Rata-rata Keberhasilan											97%

Gambar 4.3 Hasil Pengujian 3

Pengujian 4											
	1	2	3	4	5	6	7	8	9	10	Persentase Keberhasilan
L1	*****	EM BR IS	90%								
L2	EM BR IS	100%									
L3	EM BR IS	100%									
L4	EM BR IS	100%									
L5	*****	EM BR IS	90%								
P1	EM BR IS	100%									
P2	EM BR IS	100%									
P3	EM BR IS	100%									
P4	EM BR IS	100%									
P5	EM BR IS	100%									
Rata-rata Keberhasilan											98%

Gambar 4.4 Hasil Pengujian 4

Pengujian 5											
	1	2	3	4	5	6	7	8	9	10	Persentase Keberhasilan
L1	*****	*****	WI IL DA AN	*****	WI IL DA AN	70%					
L2	*****	WI IL DA AN	90%								
L3	*****	WI IL DA AN	90%								
L4	*****	WI IL DA AN	WI IL DA AN	*****	WI IL DA AN	80%					
L5	*****	WI IL DA AN	90%								
P1	*****	*****	*****	WI IL DA AN	*****	WI IL DA AN	60%				
P2	*****	*****	WI IL DA AN	*****	WI IL DA AN	70%					
P3	*****	WI IL DA AN	*****	80%							
P4	*****	*****	WI IL DA AN	80%							
P5	*****	*****	WI IL DA AN	*****	70%						
Rata-rata Keberhasilan											76%

Gambar 4.5 Hasil Pengujian 5

5. Kesimpulan dan Saran

5.1 Kesimpulan

1. Sistem dapat bekerja semakin baik apabila semakin sedikit *noise* yang ikut terekam pada proses training. Semakin jernih suara yang dilatih pada saat proses *training*, semakin baik sistem mengolah suara yang akan diproses.
2. Perubahan parameter sangat diperlukan untuk mengetahui frekuensi optimal dimana informasi paling banyak berada dan menghindari *noise* pada frekuensi yang lain.
3. Pelafalan nama pengguna sistem harus sesuai dengan data *training* agar sistem dapat mendeteksinya sebagai *input*.

5.2 Saran

1. Untuk pengembangan lebih lanjut, penerapan aplikasi masih menggunakan *library* dengan *database* bahasa Inggris sehingga kata-kata yang terdapat di kamus pengucapan harus disesuaikan dengan pelafalan orang Indonesia. Harapan untuk penelitian berikutnya dapat membuat *library* yang dapat mengenali pola pengucapan bahasa Indonesia murni.
2. Pada saat penambahan perbendaharaan kata, proses perekaman suara sebaiknya dilakukan di ruangan yang kedap suara agar tidak banyak *noise* yang ikut terekam dan menggunakan *microphone* yang kualitasnya bagus supaya suara terdengar lebih jelas.
3. Penggunaan Raspberry Pi memang sangat baik untuk perancangan sistem yang dibuat *mobile*, namun langkah lebih baik apabila pada penelitian berikutnya menggunakan mini PC dengan spesifikasi yang lebih tinggi seperti Odroid atau Roseapple Pi.

Daftar Pustaka

- Eko Budi Prasetyo, M. 2010. "Teori Dasar Hidden Markov Model", Program Studi Sistem dan Teknologi Informasi, Institut Teknologi Bandung.
- Tunarungu. "Anak Tunarungu". 16 Mei 2014. Tersedia di <<http://tunarungu.com>> [Diakses 25 Januari 2016]
- Anonymous. Tersedia di : <<http://www.who.int/pbd/deafness/estimates/en/>> [Diakses 25 Januari 2016]
- Marietha, Sonya, Ayu Purwarianti, dan Dessi Puji Lestari. 2012, "SMS suara Application with Automatic Speech Recognition and Text to Speech on Mobile Phone", Jurnal Sarjana ITB bidang Teknik Elektro dan Informatika Vol.1, No.1, hal 39-43.
- Ahsan, K. M. T. 2011, "Implementation of Bangla Speech Recognition System on Cell Phones", School of Engineering and Computer Science, BRAC University.
- Munawar, B. 2010-08-20, "Word Identification Using Hidden Markov Model (HMM) Through Feature Extraction Linear Predictive Coding (LPC)", Undergraduate Theses from Perpustakaan UNIKOM.
- Monika, V. 2012, "Perancangan Program Aplikasi Android Speech To Text Bahasa Indonesia dan Inggris Menggunakan Metode Hidden Markov Model", Universitas Binus.
- Anonymous. Carnegie Mellon University. Tersedia di : <<http://cmusphinx.sourceforge.net/wiki/tutorialPocketSphinx/>> [Diakses 30 Januari 2016]
- Independent Arabic Speech Recognition System Based on Hidden Markov Models Using Sphinx Tools". International Conference on

Computer and Communication Engineering
(ICCCE 2010).

Anonymous. Carnegie Mellon University.
Tersedia di :
<<http://cmusphinx.sourceforge.net/wiki/tutorial>
alam> [Diakses pada 30 Januari 2016]

Noora M. Rodliyah, I Ketut Eddy Purnama, dan
Ahmad Zaini. 2010. "Speech to Video (S2v),
Sistem Komunikasi Portabel Untuk Penderita
Tunarungu", Jurusan Teknik Elektro, Institut
Teknologi Sepuluh Nopember.

I Gede Kusuma Negara. 2011. Panduan Archlinux
untuk Pemula. Jurusan Informatika Wearness
Education Center Bali.

Adafruit Industries. 2015. Introducing the
Raspberry Pi 2 – B Model.

