PERANCANGAN APLIKASI SPEECH TO TEXT BAHASA INGGRIS KE BAHASA BALI MENGGUNAKAN POCKETSPHINX BERBASIS ANDROID (Design Application Speech to Text English to Balinese Language Using PocketSphinx Base On Android)

I Kadek Suryadharma[1] **Gelar Budiman, ST., MT.**[2] **Budhi Irawan, Ssi.,MT.**[3]
(kd.suryadharma@gmail.com) (glb@ittelkom.ac.id) (bir@ittelkom.ac.id)
Jurusan Teknik Telekomunikasi – FakultasTeknik Elektro – UniversitasTelkom
Jl. Telekomunikasi, Dayeuh Kolot Bandung 40257 Indonesia

ABSTRAK

Speech recognition atau pengenalan ucapan merupakan teknologi yang mampu mengenali pembicaraan atau perkataan tanpa memperdulikan siapa pembicaranya. Masukan berupa suara mampu diubah menjadi text yang mampu dibaca. Speech recognition banyak di implementasikan dengan perangkat pintar, mobil, television, ruangan dan masih banyak yang lainnya. Dengan menggunakan teknologi seperti ini memudahkan kita untuk melakukan perintah menggunakan suara semisal pada mobile application.

Saat ini perkembangan smart phone sudah sangat maju. Penerapan speech recognition pada aplikasi android terus dilakukan, speech to text salah satunya. Untuk itu dibuat sebuah aplikasi android speech to text dari bahasa inggris ke bahasa bali menggunakan pocketsphinx. Aplikasi ini tidak memerlukan akses internet sehingga dapat digunakan di mana saja bagi para wisatawan yang berminat mengetahui sedikit kata-kata dalam bahasa bali. Tidak hanya itu, aplikasi ini juga ditujukan bagi pengguna android yang ingin menambah perbendaharaan kata dalam bahasa bali.

Aplikasi ini mampu memberikan akurasi diatas 80% dari hasil analisis parameter-parameter yang digunakan. Dari hasil pengujian dengan MOS didapat nilai di atas 4. Dengan kata lain aplikasi dapat diterima user dan dinilai baik oleh pengguna.

Kata kunci *: Speech Recognition, Speech to Text, Android, Pocketsphinx.*

ABSTRACT

Speech recognition is a technology that can recognize speech or words regardless of who the speaker. Input as voice can converted into text that is able to read. Speech recognition is implemented with a lot of smart devices, cars, television, room and many others. By using this kind of technology allows us to use voice commands such as the mobile application.

Currently, the development of smart phones are very advanced. The implementation of speech recognition in android application continuing, speech to text are one of them. So it created an android application speech to text from English to Balinese language use pocketsphinx. This application does not require internet access so it can be used anywhere for tourists who are interested in knowing a few words in the Balinese language. Not only that, this application is also intended for android users who want to increase the vocabulary words in Balinese language.

This application is able to provide accuracy over 80% from results of the analysis parameters used. From the test result with MOS parameter, obtained values above 4. In other words, the application can be accepted and judged good by the user.

Keywords *: Speech Recognition, Speech to Text, Android, Pocketsphinx.*

I. PENDAHULUAN

1.1 LatarBelakang

Telepon genggam atau telepon selular (ponsel) merupakan perangkat telekomunikasi yang mempunyai kemampuan dasar sama dengan telepon konvensional yang sering kita kenal telepon rumah namu telepon selular mampu dibawa kemana-mana dan tetap dapat berkomunikasi tanpa harus terhubung dengan jaringan telepon kabel. Perkembangan telepon seluar atau *handphone* saat ini semakin maju. Mulai dari super *handphone* dengan spesifikasi *processor* dan pengolahan gambar yang canggih, teknologi kamera ponsel yang menyamai kamera profesional, hingga ponsel yang bisa digunakan sebagai televisi. Karena itulah telepon selular saat ini lebih dikenal dengan sebutan *smartphone*.

Banyak pengembang smartphone di dunia dengan berbagai macam sistem operasi diantaranya ialah windows phone, iOS, BlackBerry dan Android. Belakangan ini Android menjadi *Operating System* (OS) terlaris tidak hanya di Indonesia tapi diseluruh dunia. Saat ini *smart phone* berbasis Android khususnya sudah menjadi kebutuhan masyarakat karena menjanjikan banyak kemudahan sehari-hari salah satunya ialah *speech recognition* yang dikembangkan oleh perusahaan Google. Dengan *fiture* seperti ini, pengguna dapat dimudahkan mencari lokasi, artikel dan apapun yang kita butuhkan saat kita sibuk dalam berkendara dengan hanya menggunakan suara kita saja.

Saat bepergian keluar kota pun kita sangat memerlukan sebuah *smart phone* yang mampu mendampingi kita dalam perjalanan semisal untuk melakukan komunikasi dengan bahasa daerah yang kita kunjungi. Melihat dari perkembangan ini, maka diperlukan sebuah aplikasi pintar berbasis android dengan memanfaatkan *speech recognition*. Oleh karena itu penulis akan membuat Aplikasi *Speech to Text* Bahasa Inggris ke Bahasa Bali Berbasis Android Menggunakan *Pocketsphinx*. Aplikasi ini nantinya akan dapat menjadi panduan bagi parawisatawan asing atau lokal yang berkunjung ke Bali untuk dapat mengetahui sedikit tentang kata-kata dalam bahasa Bali. Aplikasi ini nantinya menggunakan *speech recognition offline* sehingga kita tidak perlu menggunakan akses internet.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas maka dirancang suatu aplikasi yang mencakup permasalah-permasalahan berikut :

1. Bagaimana merancang aplikasi *Speech to text* bahasa inggris ke bahasa bali berbasis android menggunakan *pocketsphinx*.
2. Bagaimana mengetahui parameter terbaik dalam mengimplementasikan pocketsphinx untuk speech to text bahasa inggris ke bahasa bali.
3. Bagaimana mengukur akurasi dan WER *speech to text* dengan database referensi yang digunakan.

1.3 Tujuan

Yang diharapkan dari penelitian ini adalah :

1. Mampu merancang aplikasi *speech to text* bahasa inggris ke bahasa bali berbasis android menggunakan *pocketsphinx*.
2. Dapat mengetahui parameter terbaik dalam mengimplementasikan pocketsphinx untuk *speech to text* bahasa inggris ke bahasa bali.
3. Dapat mengukur akurasi dan WER *speech to text* dengan database referensi yang digunakan.

1.4 Batasan Masalah

Pada tugas akhir ini, permasalahan yang dibahas dibatasi dengan beberapa batasan diantaranya :

1. Format penyimpanan file kamus suara adalah ".dic".
2. Yang menjadi masukan adalah sebuah kata yang diucapkan oleh pengguna.
3. Metode yang digunakan dalam *speech recognition* adalah *Hidden Markov Model* (HMM).
4. Ekstraksi ciri menggunakan *Mel frequency cepstral coefficient* (MFCC).
5. Menggunakan program Java-Eclipse SDK dan NDK dalam pembuatan aplikasi android.
6. Platform yang digunakan yakni Android versi *Ice Cream Sandwich (ICS)* 4.0.
7. Sistem operasi yang digunakan dalam membangun *pocketsphinx* adalah Ubuntu (Linux).
8. Keluaran berupa text hasil dari speech recognition dan hasil translate text tersebut.

II. LANDASAN TEORI

2.1 Pengenalan Ucapan (*Speech Recognition*)[1][2][4][5]

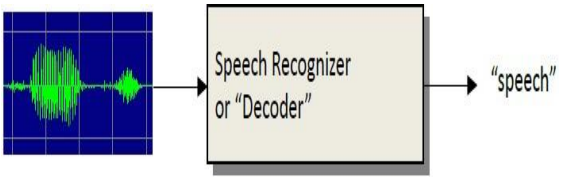
Pengenalan ucapan atau suara (speech recognition) adalah suatu teknik yang memungkinkan sistem komputer untuk menerima input berupa kata yang diucapkan. Kata-kata tersebut diubah bentuknya menjadi sinyal digital dengan cara mengubah gelombang suara menjadi sekumpulan angka lalu disesuaikan dengan kode-kode tertentu dan dicocokkan dengan suatu pola yang tersimpan dalam suatu perangkat. Hasil dari identifikasi

kata yang diucapkan dapat ditampilkan dalam bentuk tulisan sehingga dapat dibaca menggunakan perangkat teknologi.

Teknologi pengenalan ucapan merupakan gabungan dari banyak disiplin ilmu diantaranya :

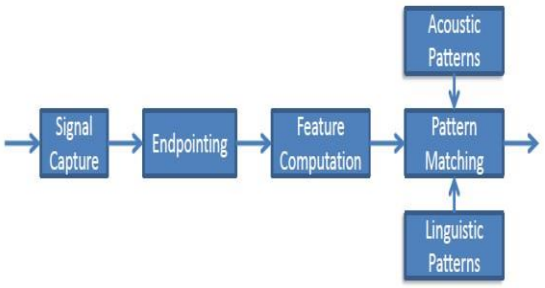
1. Pemrosesan sinyal.
2. Aljabar linear.
3. Probabilitas.
4. Linguistic (Ilmu bahasa).
5. Computer Science (Ilmu Komputer) dan masih banyak ilmu penunjang lainnya.

Secara ummum alur dari pengenalan suara yakni seperti gambar di bawah ini :



Gambar 2.1 *Speech Recognition* Secara Umum

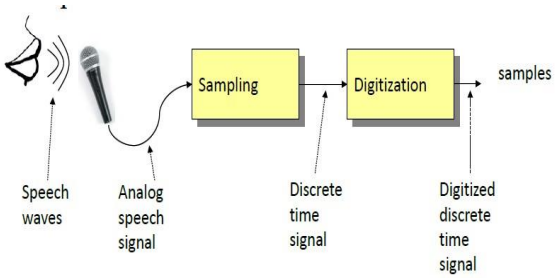
Input merupakan aliran suara yang masuk dan akan didigitalisasi oleh sistem lalu masuk ke dalam decoder yang akan mengenali suara yang masuk dan mengeluarkan hasil pengenalan suara berupa urutan kata yang diucapkan. Di bawah ini akan dijelaskan lebih lanjut tentang alur system dari pengenalan suara dengan langkah-langkah seperti gambar 2.1b:



Gambar 2.1 Alur Sistem Pengenalan Suara

2.1.1 *Speech Signal capture* (Penangkapan sinyal)

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



Gambar 2.3 Langkah Awal Penangkapan Suara

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.

2.1.2 *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 diingikan masuk ke sistem saat pengenalaan suara.

2.1.3 *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 tugas akhir ini ialah *Mel frequency cepstral coefficient* (MFCC). MFCC merupakan salah satu medode ekstraksi ciri yang banyak digunakan dalam bidang *speech technology*, baik *speaker recognition* maupun *speech recognition* dengan mengkonversikan signal suara menjadi beberapa parameter. Beberapa keunggulan dari metode ini adalah :

1. Mampu untuk menangkap karakteristik suara yang sangat penting bagi pengenalan suara atau dengan kata lain dapat menangkap informasi-informasi penting yang terkandung dalam signal suara.
2. Menghasilkan data seminimal mungkin, tanpa menghilangkan informasi-informasi penting yang dikandungnya.
3. Mereplikasi organ pendengaran manusia dalam melakukan persepsi terhadap signal suara.

2.1.4 *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 dicocokan dengan berbagai macam model. Ada 3 jenis model yang umum digunakan pada speech recognition yakni *Accoustic models*, *Pronunciation models*, *Language models*.

2.2 Pengantar Sistem Operasi Android[3][12]

Android adala sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar atau *smartphone* dan computer tablet. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak.

Secara umum arsitektur android dibagi menjadi 5 layer diantaranya :

1. Layer Applications dan Widget.
2. Layer Applications Framework
3. Layer Libraries
4. Android RunTime
5. Linux Kernel
6.

2.3 CMUSphinx[5][7][8][10][11][13]

Sphinx adalah opensourcetoolkit untuk *speech recognition* yang dikembangkan oleh Carneige Mellon University (CMU) yang berlokasi di Amerika Serikat. Untuk lebih mengenal dan menghormati pembuatnya maka Sphinx juga sering disebut dengan CMUSphinx. CMUSphinx menggunakan metode HMM dan n-gramstatisticallanguage model untuk membangun sebuah sistem *Automatic Speech Recognition* (ASR). CMUSphinx dikembangkan pertama kali oleh Kai-Fu Lee.

Sphinx project yang dikembangkan oleh Carneige Mellon University telah melahirkan beberapa produk berupa kumpulan library untuk keperluan penelitian tentang ASR. Produk-produk tersebut antara lain :

1) *Pocketsphinx* : library untuk pengenalan suara (*recognizer*) ditulis dengan menggunakan bahasa C, versi ringan dari sphinx
2) *Sphinxbase* : library yang diperlukan untuk menjalankan pocketsphinx
3) *Sphinx4* : library untuk pengenalan suara (*recognizer*) ditulis dengan menggunakan bahasa Java
4) *CMUclmtk* : tools untuk membangun language model
5) *Sphinxtrain* : tools untuk membangun acoustic model
6) *Sphinx3* : decoder untuk speechrecognition yang ditulis dengan menggunakan bahasa C.

2.3.1 Pocketsphinx

Pocketsphinx merupakan library pengenalan ucapan versi mobile dari sistem Sphinx yang dirancang oleh Carnegie Mellon University. Proses pembelajaran unit-unit suara disebut training, sedangkan proses menggunakan pengetahuan yang diperoleh untuk menyimpulkan urutan yang paling mungkin dari unit dalam sinyal yang diberikan disebut decoding, atau secara sederhana disebut pengenalan (recognition). Karena terdapat dua proses tersebut maka diperlukan sphinx trainer dan sphinx decoder. Pocketsphinx adalah library yang berkorelasi dengan library lain yaitu Sphinxbase yang menyediakan fungsionalitas umum untuk semua tools yang ada di CMUSphinxproject. Komponen utama untuk membangun pocketsphinx ini ada 3 yakni pocketsphinx, sphinxbase, dan SphinxTrain.

1. Komponen Untuk Training

2. Komponen Untuk Decoding
3. Komponen Sphinxbase

2.4 Hidden Markov Model (HMM)[5][9][11]

Markov Model biasa disebut sebagai Markov Chain atau Markov Process. Model ini ditemukan oleh Andrey Markov dan merupakan bagian dari proses stokastik yang memiliki properti Markov. Dengan memiliki property tersebut berarti, apabila diberikan inputan keadaan saat ini, keadaan akan datang dapat diprediksi dan ia lepas dari keadaan di masa lampau. Artinya, deskripsi kondisi saat ini menangkap semua informasi yang mempengaruhi evolusi dari suatu sistem dimasa depan. Dengan kata lain, Kondisi masa depan dituju dengan menggunakan probabilitas bukan dengan deterrninitas.

Model ini merupakan bagian dari finite state atau finite automaton. Finite automation sendiri adalah kumpulan state yang transisi antar state-nya dilakukan berdasarkan masukan observasi. Pada Markov Chain. setiap busur antar state berisi probabilitas yang mengindikasikan kemungkinan jalur tersebut akan diambil. Jumlah probabilitas semua busur yang keluar dari sebuah simpul adalah satu.

III. PERANCANGAN SISTEM

Pada tugas akhir ini dirancang suatu aplikasi yang mampu mengenali sebuah kata berbahasa inggris yang diucapkan dan menterjemahkannya ke dalam kata berbahasa bali dalam format text pada *platform* android. Dengan menggunakan pocketsphinx sebagai library, aplikasi ini dapat digunakan dalam platform android tanpa harus menggunakan akses internet.

3.1 Gambaran Umum Sistem

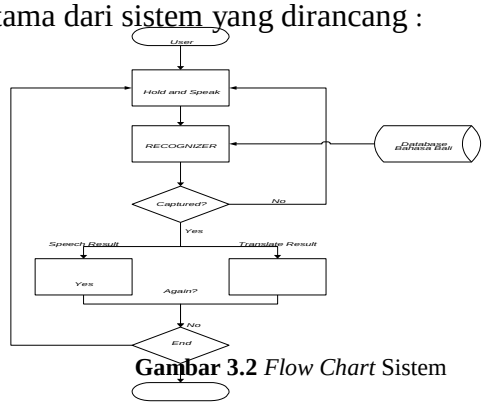
Secara umum, gambaran sistem pada tugas akhir ini dapat dilihat pada gambar di bawah ini :

Gambar 3.1 Gambaran Umum Sistem

Pada gambar 3.1 *user* akan mengucapkan sebuah kata yang kemudian diteruskan ke *smartphone* android. Aplikasi ini akan memproses suara yang masuk untuk dapat dikenali dan diterjemahkan kedalam *text*. Hasil akhirnya berupa text hasil terjemahan kata yang diucapkan.

3.2 *Flow Chart* Sistem

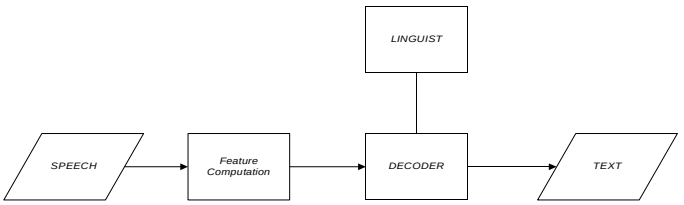
Berikut ini adalah blok utama dari sistem yang dirancang :



Gambar 3.2 *Flow Chart* Sistem

Dari diagram alur aplikasi ini, dapat dilihat bahwa inputan berupa suara pengguna. Langkah-langkah untuk menggunakan aplikasi ini yakni pertama-tama *user* harus menekan tombol yang ada pada main menu. Kemudian si pengguna mengucapkan sebuah kata sambil tetap menekan tombil tersebut. Jika kata telah selesai di ucapkan, lepas tombol. Kemudian block *recognizer* akan melakukan proses *recognition* dan mengubah suara yang masuk menjadi text. Hasil keluaran berupa text bahasa inggris yang di ucapkan dan hasil translate ke bahasa Bali.

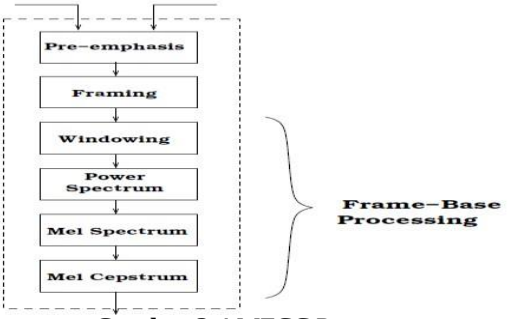
Pada blok *recognizer* terdiri dari 3 komponen utama yakni *Front-End*, *Decoder* dan bagian *Linguist*. Di bawah ini block diagram dari *recognizer* itu sendiri :



Gambar 3.3 Block diagram *Recognizer*

3.2.1 *Feature Computation*

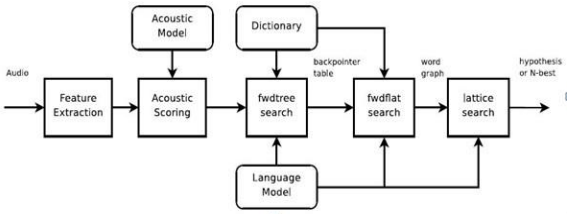
Bagian *feature computation* merupakan masukan dari *decoder* itu sendiri. Bagian ini sendiri merupakan bagian yang berperan dalam merubahan atau tranformasi bentuk gelombang suara menjadi ciri-ciri yang unik dimana nantinya akan digunakan dalam proses pengenalan ucapan. Pocketsphinx sendiri menggunakan mel-frequency cepstral coefficient (MFCC).



Gambar 3.4 MFCC Proses

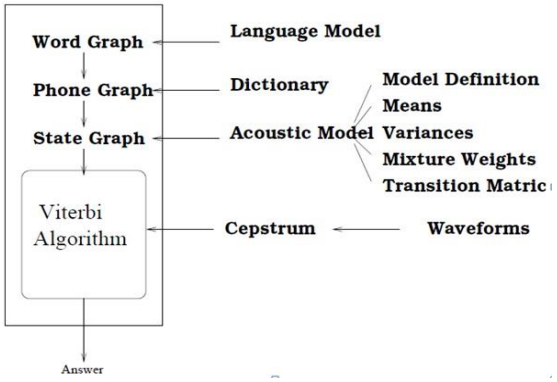
3.2.2 *Decoder*

Decoder merupakan bagian inti dari pengenal suara. Decoder sendiri terbagi menjadi tiga modul utama yakni : *acoustic modeling*, *forward search*, and *graph search*.



Gambar 3.5 Arsitektur *Decoder* Pocketsphinx

Gambar di atas merupakan keseluruhan proses *decoding*. Secara sederhana proses tersebut dapat dilihat pada gambar berikut ini :



Gambar 3.6 Penggunaan *Resource o*leh Decoder

Alur dari decoding itu sendiri secara umum yakni :

1. *Word Graph.*

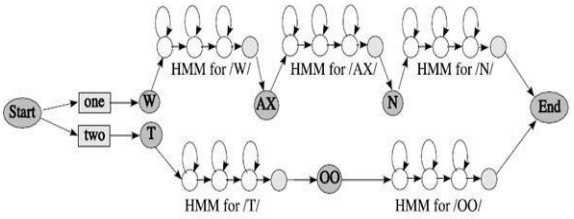
   Word graph merupakan graf dari kata dan dapat dinyatakan sebagai graf yang diberi label. Dengan kata lain, graph ini graph yang merepresentasikan *grammar* dimana tiap node terdiri dari 1 kata saja. Pada akhirnya dengan menggabungkan *word graph* dapat diperoleh makna dari kesatuan teks.

2. *Phone Graph*.
   Graph yang tiap node nya merepresentasikan fonem dari kata. *Graph* ini nantinya akan menentukan kecenderungan suatu fonem berpindah ke fonem lain sesuai *node* (titik).
3. *State Graph*
   State graph disini merupakan graf yang menandakan kondisi. *State graph* juga menentukan representasi statistikal dari tiap fonem yang membentuk kata. Nantinya akan di cocokkan dengan data yang ada dan parameter dari akustik model sendiri.
4. Viterbi Algorithm[6]
   Viterbi algoritma adalah metode decoding untuk mengkodekan kembali bit yang telah dikodekan oleh convolutional code dengan prinsip mencari kemungkinan bit yang paling mirip atau dapat disebut maximum likelihood. Proses decoding dapat disamakan dengan membandingkan deretan bit yang diterima dengan semua kemungkinan bit terkode, dari proses perbandingan tersebut akan dipilih bit yang paling mirip antara deretan bit yang diterima dengan kemungkinan deretan bit bit yang ada.

3.2.2 Linguist

Linguist terdiri dari 3 buah komponen utama yakni :

1. *Language Model*
   Modul ini menyediakan struktur bahasa pada tingkat kata. Dengan kata lain *language model* merepresentasikan urutan kata yang valid dan paling masuk akal. Implementasi dari model ini mendukung berbagai format yakni :
   - SimpleWordListGrammar : mendefinisikan tata bahasa berdasarkan daftar kata.
   - JSGFGrammar : mendukung format Java™ Speech API Grammar Format (JSGF)
   - LMGrammar : mendefinisikan tata bahasa berdasarkan pada model bahasa statistik.
   - FSTGrammar : mendukung transduser finite-state (FST) dalam format tata bahasa Advanced Research Projects Agency (ARPA) FST.
   - SimpleNGramModel : menyediakan dukungan untuk model ASCII N-Gram dalam format ARPA sehingga cocok untuk model bahasa yang kecil.
   - LargeTrigramModel : menyediakan dukungan untuk model N-Gram yang dihasilkan oleh Cambridge Statistical Language Modeling Toolkit. Model ini cocok untuk ukuran yang besar.
2. *Dictionary*
   Modul ini menyediakan *pronunciations* atau pelafalan untuk kata-kata yang ada dalam *Language Model*. Pengucapan memecah kata menjadi urutan unit sub-kata yang ditemukan pada model akustik.
3. *Acoustic Model*
   Model akustik menyediakan pemetaan antara unit ucapan atau *speech* dan HMM dimana nilainya dapat dicocokkan dengan hasil ekstraksi ciri yang disediakan oleh bagian *Front-End*.
   Secara keseluruhan, bagian *linguist* menghasilkan SearchGraph yang digunakan oleh decoder selama pencarian.



Gambar 3.7 SearchGraph

SearchGraph berupa *direct graph*. Masing-masing *"state"* atau kondisi merepresentasikan komponen dari bagian *linguist*. Missal kata "one" dan "two" merupakan bagian dari *Language Model*. Pecahan kata-kata pada lingkaran yang berwarna hitam merupakan bagian dari *Dictionary*. Bagian HMM mencermikan *Acoustic Model*.

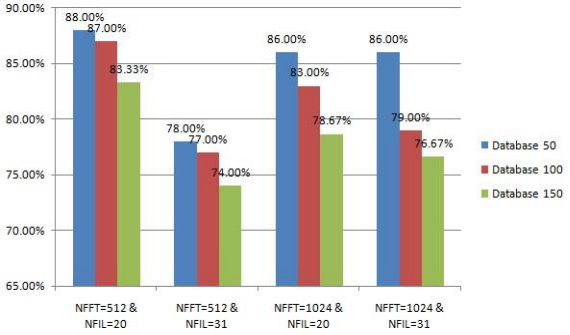
IV. PENGUJIAN DAN ANALISIS

4.1 Pengujian *Word Error Rate* (WER)

Pengujian ini dilakukan dengan cara merubah nilai parameter nfft, nfil, *lower* frekuensi dan *upper* frekuensi agar mendapatkan hasil akurasi terbaik. Dengan kata lain *word error rate* (WER) yang dihasilkan aplikasi akan sekecil mungkin.

4.1.1 Parameter NFFT dan NFIL

Dengan merubah nilai parameter NFFT dan NFIL ini, akurasi yang dihasilkan dapat berbeda-beda. Untuk hasil pengujian lengkap dapat dilihat pada grafik di bawah ini :

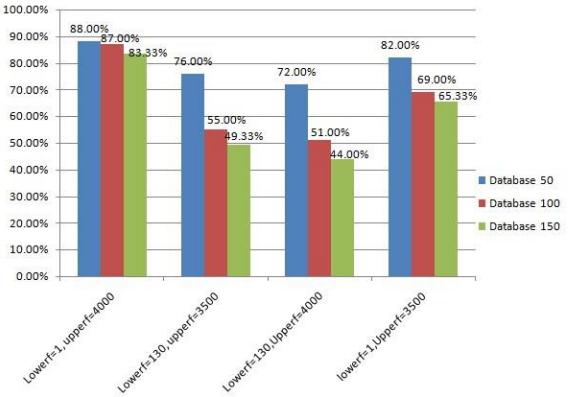


Gambar 4.1 Grafik Perbandingan Akurasi untuk Seluruh NFFT & NFIL

Dari hasil pengujian yang didapat, nilai terbaik untuk parameter NFFT dan NFIL yakni NFFT = 512 dan NFIL= 20. Akurasi yang dihasilkan untuk setiap database yang digunakan oleh nilai parameter ini merupakan akurasi tertinggi. Sehingga dapat disimpulkan merupakan nilai parameter terbaik untuk aplikasi *speech to text* menggunakan Pocketsphinx.

4.1.2 Parameter Lower Frekuensi dan Upper Frekuensi

Akurasi yang dihasilkan juga dipengaruhi dengan merubah nilai parameter lowerf dan upperf ini,. Untuk hasil pengujian lengkap dapat dilihat pada grafik di bawah ini :



Gambar 4.2 Grafik Perbandingan Akurasi untuk Seluruh Lowerf & Upperf

Dari hasil pengujian yang didapat, nilai terbaik untuk parameter *lower* frekuensi dan *upper* frekuensi yakni Lowerf = 1Hz dan Upperf= 4000Hz. Akurasi yang dihasilkan untuk setiap database yang digunakan oleh nilai parameter ini merupakan akurasi tertinggi. Sehingga *word error rate* untuk sistem dapat ditekan sekecil mungkin agar pencarian kata menggunakan ucapan dapat lebih baik.

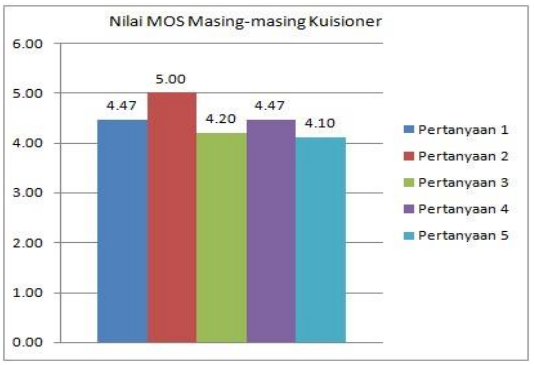
Kombinasi dari nilai NFFT, NFIL, Lowerf dan Upperf tertentu menghasilkan *word error rate* yang berbeda-beda. Dapat disimpulkan bahwa nilai parameter dengan NFFT=512, NFIL=20, Lowerf=1Hz dan Upperf=4000Hz merupakan nilai terbaik untuk mengimplementasikan *speech to text* menggunakan Pocketsphinx.

4.2 Pengujian *Mean Opinian Score* (MOS)

Pengujian secara subjektif ini menggunakan MOS (*Mean Opinian Score*). MOS didapat dengan cara mengajukan kuisioner kepada 30 responden dengan 5 buah pertanyaan seputar aplikasi *speech to text* ini. Masing-masing jawaban memiliki bobot nilai dari 1-5. Untuk menghitung MOS dapat digunakan persamaan seperti dibawah ini :

$$MOS = \frac{\text{[illegible]}}{\text{[illegible]}}$$

Berikut adalah analisis dari hasil pengujian yang dilakukan dengan cara mengajukan pertanyaan ke 30 responden dimana responden mencoba aplikasi ini :



Gambar 4.3 Grafik Nilai MOS

Terlihat pada grafik di atas bahwa nilai rata-rata opini 30 responden diantara 4 hingga 5. Ini menunjukkan hasil yang baik. Hasil yang baik menandakan bahwa pengguna atau *user* menilai aplikasi ini baik.

Dapat diambil kesimpulan bahwa aplikasi ini memiliki tampilan yang sangat baik dan *user friendly*. Dari sisi fungsionalitas, masing-masing menu pada aplikasi ini sudah berjalan sesuai dengan kegunaannya. Hasil terjemahan sudah cukup baik. Aplikasi ini juga dinilai membantu pengguna dalam menambah perbendaharaan kata bahasa bali.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil pengujian dan analisis yang telah dilakukan pada sistem *speech to text* bahasa inggris ke bahasa bali menggunakan pocketsphinx ini, dapat diambil kesimpulan sebagai berikut :

1. Implementasi dari pocketsphinx ke dalam aplikasi *speech to text* dapat direalisasikan dengan hasil pengenalan yang baik. Ini terlihat dari tingkat akurasi yang dihasilkan di atas 80%.
2. Parameter terbaik yang mampu menekan *word error rate* (WER) sekecil mungkin yakni sebagai berikut :
   - NFFT = 512
   - NFIL = 20
   - Lowerf = 1Hz
   - Upperf = 4000Hz

   Penggabungan parameter di atas menjadikan aplikasi *speech to text* ini dapat mengenali kata dengan baik.
3. Jumlah kata pada database dapat mempengaruhi akurasi dari sistem. Jumlah kata dalam database dengan akurasi berbanding terbalik. Semakin banyak jumlah kata yang terdapat pada database, semakin kecil akurasi yang dihasilkan dan sebaliknya.
4. Dalam sistem pengenalan suara, banyak faktor yang mempengaruhi akurasi. Faktor-faktor yang mempengaruhi nilai akurasi diantaranya yaitu aksen saat mengucapkan sebuah kata, dialek tiap orang berbeda, cara pengucapan bisa pelan ataupun kasar, dan banyaknya kata yang pengucapannya mirip.
5. Dari hasil pengujian berdasarkan survei terhadapat 30 responden, aplikasi *speech to text* ini mendapat nilai MOS diatas 4 dari batas maksimal 5. Ini berarti aplikasi dinilai baik oleh *user*.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan pada tugas akhir ini, masih banyak kekurangan yang terdapat pada sistem. Beberapa saran yang dapat dikembangkan pada penelitian selanjutnya, diantaranya adalah:

1. Metode yang digunakan untuk aplikasi *speech to text* dapat diganti dengan metode lain yang lebih baik lagi dalam mengenali suatu kata.
2. Kedepannya pada sistem ini bukan hanya dapat merubah suatu *speech* ke *text* dan menterjemahkannya kedalam *text*, namun hasil terjemahan dapat kita dengar tidak hanya sebatas *text* yang terlihat.
3. Kedepannya pada sistem ini dapat diimplementasikan pada bidang lain misalnya perintah suara untuk menjalankan aplikasi pada *smartphone*, perintah suara untuk melakukan kontrol rumah atau mobil melalui *smartphone* android.
4. Kedepannya sistem ini dapat dikembangkan ke *platform handphone* lain seperti Blackberry, Iphone dan Windows Phone.

DAFTAR PUSTAKA

[1] Maharani, Warih. 2009. *Analisis Performansi Recognition Experimental System (RES) Untuk Bahasa Indonesia*. Institut Teknologi Telkom. Bandung.

[2] Bhiksha, Raj dan *Rita Singh*. 2013. Design and Implementation of Speech Recognition Systems. Carnegie Melon University. United States.

[3] Safaat, Nazruddin. 2011. *Pemrograman Aplikasi Mobile Smartphone dan Table PC Berbasis Android.* Bandung: Informatika.

[4] Putra, Darma., dan Adi Resmawan. 2011. Verifikasi *Biometrika Suara Menggunakan Metode MFCC Dan DTW*. Lontar Komputer Vol.2 No.1. Universitas Udayana.

[5] Monika, Vera. 2012. *Perancangan Program Aplikasi Android Speech To Text Bahasa Indonesia dan Inggris Menggunakan Metode Hidden Markov Model*. Binus University. Jakarta.

[6] Mahyadi, Aslam. 2011. *Visualisasi Kinerja Pengkodean Menggunakan Algoritma Viterbi*. Politeknik Elektronika Negeri Surabaya

[7] Chan, Arthur., Evandro Gouvea., dkk. 2007. *(Third Draft) The Hieroglyphs: Building Speech Applications Using CMU Sphinx and Related Resources*

[8] Daines, David Huggins. 2011. *An Architecture for Scalable, Universal Speech Recognition.* Carnegie Mellon University. Pittsburgh.

[9] Prasetyo, Muhammad Eko Budi. 2010. *Teori Dasar Hidden Markov Model*. Institut Teknologi Bandung.

[10] Walker, Willie., Paul Lamere., dkk. 2004. *Sphinx-4: A Flexible Open Source Framework for Speech Recognition*. SUN Microsystems INC.

[11] Young, Steve., Gunnar Everman., dkk. 2006. *The HTK Book version 3.4*. Cambridge University Engineering Department.

[12] Supatra, N. Kanduk. 2010. *Kamus Bahasa Bali.* CV. Kayumas Agung. Denpasar.