METODE *QUERY MATCHING* UNTUK SISTEM PENJAWAB PERTANYAAN DENGAN *SPEECH RECOGNITION* MEMANFAATKAN *LIBRARY* POCKETSPHINX

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Febriawan Ariful Fikri
NIM: 115060900111014

PROGRAM STUDI TEKNIK INFORMATIKA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

METODE *QUERY MATCHING* UNTUK SISTEM PENJAWAB PERTANYAAN DENGAN *SPEECH RECOGNITION* MEMANFAATKAN *LIBRARY* POCKETSPHINX

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Febriawan Ariful Fikri
NIM: 115060900111014

Skripsi ini telah diuji dan dinyatakan lulus pada
7 Januari 2016
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Eko Setiawan, S.T., M.T	Adharul Muttaqin, S.T., M.T
NIK: 201102 870610 1 001	NIP: 19760121 200501 1 001

Mengetahui
Ketua Program Studi Teknik Informatika

Drs. Marji, M.T
NIP: 19670801 199203 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 25 Januari 2016

Febriawan Ariful Fikri

NIM: 115060900111014

KATA PENGANTAR

Puji syukur kehadirat Allah SWT, berkat rahmat dan karunia-Nya penulis dapat menyelesaikan penyusunan skripsi dengan baik. Shalawat serta salam semoga senantiasa terlimpahkan kepada Nabi Muhammad SAW.

Penulisan skripsi ini diajukan untuk memenuhi salah satu syarat untuk mendapatkan gelar Sarjana pada Program Studi Sistem Komputer, Program Teknologi Informatika dan Ilmu Komputer, Universitas Brawijaya Malang. Judul skripsi yang penulis ajukan adalah "Metode *Query Matching* untuk Sistem Penjawab Pertanyaan dengan *Speech Recognition* Memanfaatkan *Library* PocketSphinx".

Dalam penyusunan dan penulisan skripsi ini tidak terlepas dari bimbingan serta dukungan dari berbagai pihak. Oleh karena itu penulis menyampaikan banyak terimakasih kepada yang terhormat:

1. Kedua orang tua yang selalu memberikan doa, motivasi, dukungan moril dan materil sebagai penyemangat selama masa studi.
2. Bapak Adharul Muttaqin, S.T., M.T. selaku Kepala Program Studi Sistem Komputer, pembimbing akademik, serta pembimbing skripsi yang telah memberikan masukan-masukan dalam penyusunan skripsi ini.
3. Bapak Eko Setiawan, S.T., M.T., M.Eng. selaku pembimbing skripsi yang telah membimbing penulis dalam penyusunan skripsi.
4. Bapak Barlian Henryranu Prasetio, S.T., M.T. selaku Kepala Laboratorium Sistem Komputer yang memberikan saran dan bimbingan dalam penyusunan skripsi.
5. Rekan-rekan mahasiswa Program Studi Sistem Komputer yang selalu berbagi dukungan dan motivasi selama masa perkuliahan.
6. Semua pihak yang telah membantu penulis dalam penyusunan skripsi ini.

Pada akhirnya, penulis menyadari bahwa skripsi ini masih belum sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun. Penulis berharap semoga skripsi ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi.

Malang, 20 Januari 2016

Penulis

115060900111014@mail.ub.ac.id

ABSTRAK

Pengenalan ucapan adalah suatu sistem yang berfungsi untuk mengubah kata-kata ucapan menjadi bentuk teks terbaca yang dapat diolah oleh komputer. PocketSphinx, yang merupakan *library* pengenal ucapan yang bersifat *open source*, lebih merupakan sebuah *speech to text* yang belum dapat mengenali maksud dari sebuah ucapan. Atas dasar itu, perlu dilakukan penambahan sistem yang mampu menangani keterbatasan tersebut yang diterapkan pada sebuah sistem penjawab pertanyaan.

Pada penelitian ini, pengenalan maksud suatu ucapan dilakukan dengan menambahkan software dengan memanfaatkan metode *query matching*. Teks Bahasa Indonesia yang dihasilkan oleh PocketSphinx selanjutnya diolah dengan teknik *preprocessing* untuk mendapatkan kata kunci. Kata yang sudah terproses kemudian dicocokkan dengan dokumen yang sudah dikelompokkan menggunakan metode *query matching*. Jika kelompok dokumen teks telah ditemukan, maka respon dari teks dapat ditampilkan.

Sistem ini menyediakan 20 pertanyaan seputar ruangan dalam bahasa Indonesia di Fakultas Ilmu Komputer Universitas Brawijaya yang sudah dilatih pada *library* PocketSphinx. Dari 20 pertanyaan yang sudah dilatihkan tersebut, sistem ini dapat memberikan respon yang benar dengan persentase sebesar 82,5% dari pengujian menggunakan 3 responden yang memiliki suara yang berbeda-beda.

Kata kunci: pengenalan ucapan, query matching, PocketSphinx

ABSTRACT

Speech recognition is a system that serves to convert the spoken word into readable text that can be processed by computers. PocketSphinx is an open source library for speech recognition. However PocketSphinx can not understand the meaning of user speech. It needs additional system that can handle that limitation.

This final project offers the additional system that can extract the meaning of user speech by using query matching method. Text in Bahasa Indonesia which converted from speech by PocketSphinx then processed by preprocessing techniques to extract the keyword. Tt then matched with the documents that have been classified using query matching method. If text document group have been found, then the response of the text can be displayed.

The system provides 20 questions about local room locations in Bahasa Indonesia that have been trained in PocketSphinx library. From 20 questions that have been trained, the system can provide the correct response with a percentage of 82.5% with three respondents who have different voices.

Keywords: speech recognition, query matching, PocketSphinx

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB 1 PENDAHULUAN

Bab ini berisi pembahasan tentang hal yang menjadi latar belakang topik, permasalahan yang akan dibahas dari topik, batasan dari analisis, tujuan dan manfaat dari topik yang akan dibahas, serta sistematika dalam penulisan skripsi.

1.1 Latar belakang

Komunikasi antara manusia dan mesin dengan metode komunikasi menggunakan *keyboard*, *remote*, *mouse* ataupun semacamnya terbukti belum optimal, hal tersebut seperti dikemukakan oleh Ryszard dalam penelitiannya yang berjudul *speech in human system interaction*. Ryszard juga mengatakan bahwa metode yang paling optimal untuk proses komunikasi antara manusia dan mesin adalah melalui komunikasi ucapan, karena bagi kebanyakan orang, ucapan adalah metode yang paling alami dan yang paling nyaman untuk berkomunikasi. Dalam sistem kontrol, komunikasi dengan ucapan terkadang sangatlah penting bagi orang lanjut usia, orang dengan kebutuhan khusus (kecuali tuna wicara), ataupun pasien dengan penyakit tertentu (Ryszard, 2010).

Pentingnya komunikasi antara manusia dan mesin mulai diteliti oleh Carniege Mellon University (CMU) dalam 5 dekade terakhir melalui pengenalan ucapan otomatis (*Automatic Speech Recognition*). CMU kemudian bekerja sama dengan beberapa pihak untuk membangun sistem pengenal ucapan yang diberi nama CMUSphinx. CMUSphinx merupakan *toolkit open source* untuk pengenalan pengenalan ucapan secara otomatis. CMUSphinx terdiri dari serangkaian sistem pengenalan ucapan (Sphinx 2, Sphinx 3, Sphinx 4), pengenalan ucapan untuk sistem *embedded* (PocketSphinx), dan *trainer* untuk model akustik (SphinxTrain).

CMUSphinx dapat mendukung berbagai bahasa dengan merubah *language model* dengan menyesuaikan fonem pada masing-masing bahasa, seperti penelitian yang berjudul "Implementasi *Library* Pocketsphinx Untuk Pengenalan *Voice Command* Berbahasa Indonesia Secara *Offline"*. Menurut penelitian tersebut, *library* Pocketsphinx yang merupakan salah satu versi dari CMUSphinx dapat mengenali ucapan dengan berbahasa Indonesia dengan merubah fonem yang disesuaikan dengan pengucapan bahasa Indonesia (Hilman, 2014). Namun *library* PocketSphinx hanya sebatas mengenali ucapan saja, *library* PocketSphinx tidak mengerti maksud dari *user* ketika mengatakan sesuatu. Akibatnya, *library* PocketSphinx tidak dapat merespon ucapan dari *user*.

Dari permasalahan yang telah dipaparkan diatas, perlu diciptakan sistem yang mampu mengenali maksud ucapan dengan memanfaatkan *library* PocketSphinx, salah satunya dengan menggunakan metode *query matching* untuk menentukan respon ucapan dari *user*. Jika respon yang dihasilkan telah sesuai, dapat disimpulkan bahwa sistem mampu mengerti maksud dari ucapan yang dikatakan *user.* Metode *query matching* merupakan metode yang digunakan untuk mencari sebuah informasi, metode tersebut membandingkan *query* yang dicari dengan informasi acak yang telah disediakan (Elastic, 2016).

Berdasarkan uraian diatas, pada penelitian ini akan membahas bagaimana *library* PocketSphinx dapat mengenali maksud ucapan dengan implemetasi berupa sistem penjawab pertanyaan dengan menggunakan metode *query matching*. Dengan adanya penelitian ini, pengenal ucapan yang sebelumnya hanya mengenali ucapan, dapat mengerti maksud dari ucapan, sehingga dapat melakukan respon dengan baik.

1.2 Rumusan masalah

Berdasarkan latar belakang di atas, dapat dirumuskan beberapa masalah sebagai berikut:

1. Bagaimana cara merancang pengenal ucapan menggunakan *library* PocketSphinx?
2. Bagaimana cara *library* PocketSphinx mengenali maksud ucapan agar dapat menentukan respon jawaban?
3. Bagaimana pengaruh metode *query matching* yang diterapkan pada sistem penjawab pertanyaan berbahasa Indonesia dengan pengenalan ucapan menggunakan *library* PocketSphinx?

1.3 Tujuan

Penelitian ini bertujuan untuk menciptakan sistem yang mampu mengenali maksud ucapan, sehingga dapat digunakan sebagai sarana antarmuka dengan memanfaatkan metode *query matching* dan *library* PocketSphinx.

1.4 Manfaat

Manfaat penelitian ini yaitu mengetahui efektifitas metode *query matching* apabila diterapkan pada sistem penjawab pertanyaan dengan pengenalan ucapan berbahasa Indonesia menggunakan *library* PocketSphinx, sehingga library PocketSphinx mampu memahami maksud ucapan *user* serta dapat meresponnya dengan baik.

1.5 Batasan masalah

Agar permasalahan yang dirumuskan dapat lebih terfokus, maka penelitian ini dibatasi dalam hal:

1. PocketSphinx digunakan sebagai *library* untuk mengenali ucapan yang digunakan sebagai kalimat pertanyaan.
2. Sistem hanya mengenali 20 kalimat pertanyaan untuk mencari ruangan di Fakultas Ilmu Komputer Universitas Brawijaya.
3. Sistem hanya dapat mengenali ucapan berbahasa Indonesia dalam bentuk ucapan manusia.
4. Jawaban untuk respon dari pertanyaan berupa teks yang ditampilkan melalui *terminal* di linux.

1.6 Sistematika pembahasan

Sistematika pembahasan ditunjukkan untuk memberikan gambaran dan uraian dari penyusunan tugas akhir secara garis besar yang meliputi beberapa bab, sebagai berikut.

BAB I : **Pendahuluan**
Bab ini merupakan dasar penyusunan skripsi ini yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan.

BAB II : **Landasan Kepustakaan**
Bab ini membahas tinjauan pustaka dan dasar teori yang digunakan untuk menunjang penulisan skripsi "Metode Query Matching untuk Sistem Penjawab Pertanyaan dengan Speech Recognition Memanfaatkan Library PocketSphinx".

BAB III : M**etodologi**
Bab ini menguraikan tentang metode dan langkah kerja yang terdiri dari studi literatur, analisis kebutuhan simulasi, perancangan sistem, implementasi dan analisis serta pengambilan kesimpulan.

BAB IV : **Persyaratan**
Bab ini menguraikan tentang segala kebutuhan dan melakukan gambaran tentang segala aktifitas yang dilakukan untuk merancang sistem yang dibuat.

BAB V : **Perancangan dan Implementasi**
Bab ini menguraikan proses implementasi dari dasar teori yang telah dipelajari sesuai analisis dan perancangan sistem.

BAB VI : **Pengujian dan Analisis**
Bab ini memuat hasil pengujian dan analisis terhadap sistem yang telah direalisasikan

BAB VII : **Penutup**
Bab ini memuat kesimpulan yang diperoleh dari pembuatan dan pengujian program, serta saran-saran untuk pengembangan lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

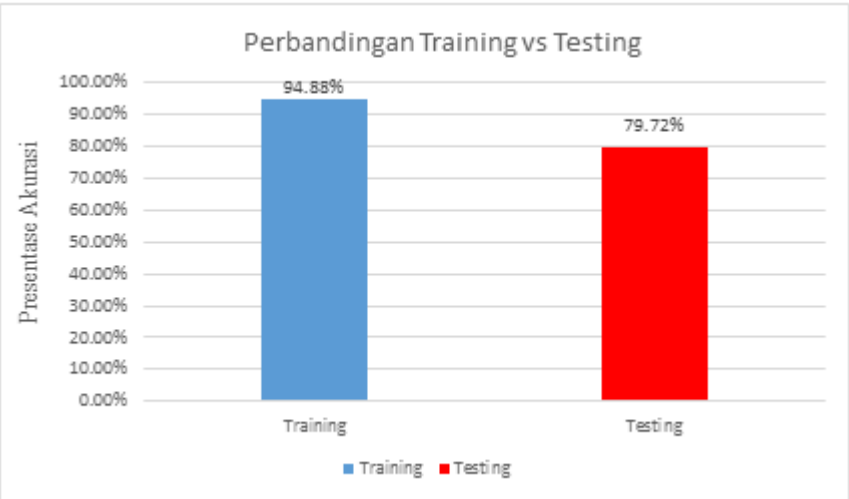
Bab ini berisi sumber pustaka yang membahas penelitian yang ada dan diusulkan. Serta dasar teori yang berisi uraian dan pembahasan tentang konsep, model, metode, atau sistem dari literatur ilmiah yang digunakan pada penelitian ini.

2.1 Sumber pustaka

Sumber pustaka membahas penelitian yang telah ada dan yang diusulkan. Pada penelitian ini kajian pustaka diambil dari beberapa penelitian yang pernah dilakukan untuk digunakan sebagai acuan dalam penyusunan kerangka penelitian.

2.1.1 Implementasi *Library* PocketSphinx Untuk Pengenalan *Voice Command* Berbahasa Indonesia Secara *Offline* Oleh Mokhammad Hilman Fatah

PocketSphinx merupakan *library* pengenalan ucapan yang mendukung bahasa Inggris, Cina, Perancis, Mandarin, Jerman dan Rusia. Penelitian yang dilakukan Mokhammad Hilman Fatah pada *library* PocketSphinx yaitu merancang sistem pengenalan ucapan berbahasa Indonesia, penelitian tersebut diimplemetasikan dalam perancangan sistem berbasis *offline voice control*. Hasil pengujian yang dilakukan, memaparkan bahwa tingkat akurasi pengenalan ucapan pada saat *training* PocketSphinx sebesar 94.88% dari total 254 kosakata berbahasa Indonesia, sedangkan pada saat *testing* didapatkan hasil pengujian sebanyak 79.72% dari total 25 kosakata berbahasa Indonesia seperti yang dijabarkan oleh Gambar 2.1.



Gambar 2.1 Grafik perbandingan akurasi *Training* dan *Testing*
Sumber: M. Hilman Fatah (2014)

Dari hasil penelitian "Implementasi *Library* PocketSphinx Untuk Pengenalan *Voice Command* Berbahasa Indonesia Secara *Offline*" oleh Mokhammad Hilman Fatah tersebut dapat digunakan sebagai acuan atau referensi dalam pemilihan PocketSphinx sebagai *library* yang digunakan untuk melakukan pengenalan ucapan bahasa Indonesia, karena PocketSphinx dapat mengenali bahasa Indonesia dengan keakuratan cukup tinggi.

2.1.2 Rancang Bangun Sistem Pengelolaan Dokumen-dokumen Penting Menggunakan *Text Mining* Oleh Ahmad Hatta A. dkk.

Text mining adalah suatu proses yang bertujuan untuk menemukan informasi atau tren terbaru yang sebelumnya tidak terungkap, dengan memproses dan menganalisa data dalam jumlah besar. *Teks mining* mengasosiasikan satu bagian teks dengan yang lainnya berdasarkan aturan-aturan tertentu. Tujuan dari *text mining* yaitu mencari kata-kata yang dapat mewakili apa yang ada dalam dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen.

Dalam penelitian yang dilakukan oleh Ahmad Hatta A. dkk, mereka mentransformasikan kartu berformat kertas menjadi format digital. Dalam penelitian tersebut, *text mining* digunakan untuk memudahkan sistem dalam mendapatkan informasi didalam dokumen serta untuk mengelola dokumen sehingga dapat dikelompokkan data digital tersebut sesuai dengan tipe masing-masing. Metode *text mining* yang diimplementasikan pada penelitian tersebut, dapat dimanfaatkan oleh penulis untuk mengolah kata-kata berupa pertanyaan, sehingga dapat ditemukan kelompok pertanyaannya dan ditampilkan jawabannya.

2.1.3 *Question Answering System* Berbasis *Clustering* Pada Buku Pedoman Ptiik Dengan Menggunakan Algoritma Levenshtein Distance Oleh Deppy Rangga Maulana

Question Answering System (*QAS*) merupakan cabang ilmu dari *Information Retrieval* (*IR*) atau sistem temu kembali informasi. *IR* merupakan alat yang dapat membantu pencarian informasi dengan memberikan koleksi informasi yang sesuai dengan kebutuhan pengguna. *QAS* adalah proses interaktif antara manusia dan komputer yang meliputi pemahaman terhadap kebutuhan informasi pengguna dan menampilkan respon yang lebih efektif. Dalam penelitiannya, Deppy Rangga Maulana memanfaatkan metode klastering untuk mempersempit daerah pencarian untuk meningkatkan efektifitas *IR*. Penelitian tersebut menggunakan dokumen buku pedoman PTIIK sebagai data yang akan digunakan sebagai respon atas pertanyaan yang disampaikan oleh *user*.

QAS pada penelitian tersebut dapat diadopsi untuk digunakan pada pengenalan ucapan menggunakan *library* PocketSphinx yang selama ini hanya mengenali ucapan tetapi tidak memahami maksudnya. Dengan memanfaatkan *QAS*, ucapan berupa pertanyaan akan diidentifikasi maksudnya dengan memilih

kandidat jawaban yang seharusnya menjadi respon dari ucapan pertanyaan tersebut.

2.1.4 *Architecture of ProMe Instant Question Answering System* Oleh Guangzhi Zhang Dkk

Penelitian yang dilakukan oleh Guangzhi Zhang adalah modifikasi dari *Question Analysis System* (*QAS*) yang dikombinasikan dengan *search engine* dan *instant messenger.* Pada penelitian tersebut, seluruh pertanyaan dicari jawabannya dengan 3 metode. Pertama, jawaban akan dicari menggunakan algoritma TF-IDF, jika masih belum ditemukan, pertanyaan di*submit* pada *search engine* untuk mendapatkan jawabannya melalui pencarian informasi dari halaman web, dan jika masih belum ditemukan atau jawaban tidak memuaskan, pertanyaan akan dikirim pada orang lain melalui *instant messenger*.

Dalam penelitian tersebut, *QAS* memakai metode keyword extraction, ekstraksi dilakukan dengan memberi nilai pada tiap-tiap kata, sehingga pencarian jawaban didasarkan pada frekuensi munculnya keyword. Berikut adalah rumus untuk memberi nilai pada kata:

$$len_i = \frac{l_i}{L}$$

Dimana l_i adalah indikasi nilai dari kata ke i atau biasa dianggap normalisasi nilai, sedangkan L adalah banyaknya seluruh kata dalam kalimat.

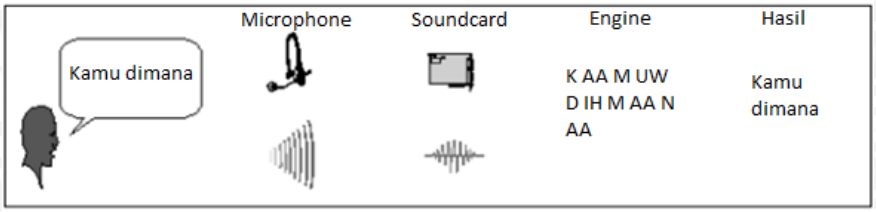
Rumus untuk pemberian nilai tersebut dapat dimanfaatkan pada program *query matching* untuk mencari bobot nilai pada masing-masing *query* atau kata sehingga setiap pertanyaan dapat ditemukan kelompoknya dan dihasilkan jawabannya.

2.2 Dasar teori

Dasar teori berisi penjelasan mengenai bahan, konsep, objek yang dibutuhkan dalam penyusunan penelitian ini.

2.2.1 Pengenalan ucapan

Pengenalan ucapan (*speech recognition*) merupakan suatu teknik yang memungkinkan sistem untuk menerima masukan berupa ucapan. Kata-kata tersebut diubah menjadi sinyal digital dengan mengubah gelombang ucapan menjadi sekumpulan angka kemudia disesuaikan dengan kode-kode tertentu dan dicocokan dengan pola yang tersimpan dalam suatu perangkat. Hasil identifikasi kata yang diucapkan dapat ditampilkan dalam bentuk tulisan atau dapat dibaca oleh perangkat teknologi (CMU, 1996).



Gambar 2.2 Dasar Pengenalan Ucapan

Sumber: M. Hilman Fatah (2014)

Gambar 2.2 merupakan ilustrasi dasar pengenalan ucapan yang diolah oleh perangkat komputer. Ucapan diucapkan melalui *microphone* kemudian diterima oleh *soundcard* komputer dan diolah oleh aplikasi pengenalan ucapan, sehingga dapat menghasilkan ucapan dalam bentuk teks (Feng, 2014).

Banyaknya kosakata dari sistem pengenalan ucapan mempengaruhi kompleksitas, parameter pelatihan dan akurasi pengenalan. Beberapa aplikasi pengenal ucapan hanya memerlukan beberapa kata, sedangkan yang lainnya memerlukan kamus yang sangat besar. Terdapat 4 macam kosakata berdasarkan jumlahnya, yaitu:

a. Kosakata ukuran kecil (*small vocabulary*) yang terdiri dari puluhan kata.
b. Kosakata ukuran sedang (*medium vocabulary*) yang terdiri dari ratusan kata.
c. Kosakata ukuran besar (*large vocabulary*) yang terdiri dari ribuan kata.
d. Kosakata ukuran sangat besar (*very large vocabulary*) yang terdiri dari puluhan ribu kata (CMU, 1996)

2.2.2 PocketSphinx

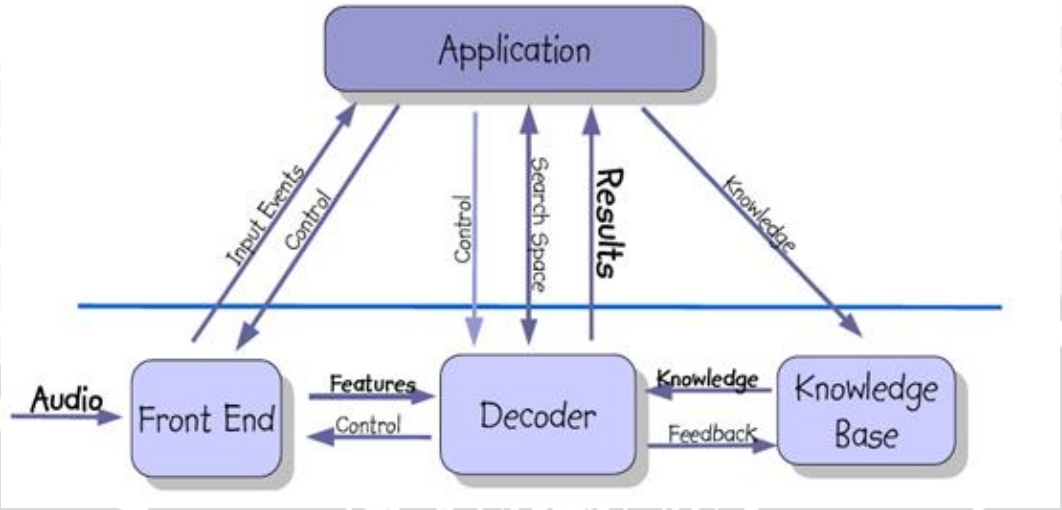
Sejak 5 dekade terakhir, Carniege Mellon University mengembangkan *library* pengenalan ucapan yang diberi nama CMUSphinx. Ada beberapa versi *library* pengenalan ucapan yang dikembangkan oleh CMU yaitu Sphinx2, Sphinx3, Sphinx4, dan PocketSphinx.

PocketSphinx merupakan pengenal ucapan versi *mobile application* dari CMUSphinx yang dirancang oleh Carniege Mellon University, sehingga PocketSphinx dapat digunakan untuk perangkat *mobile* dan sistem *embedded* karena didukung dengan kelebihan komputasi yang cepat dan ringan untuk mengenali ucapan seseorang. Dengan mengandalkan respon yang sangat cepat dan konsumsi sumber daya yang minim sehingga memungkinkan pengembangan pengenalan ucapan cocok digunakan untuk aplikasi *desktop*, komando dan kontrol serta pendiktean. Selain itu, *engine* ini dapat berguna bagi perangkat tertanam dengan *fixed-point ariphmetics* yang berhasil digunakan pada iPhone, perangkat Nokia Maemon dan Windows Mobile. CMUSphinx juga menyediakan fasilitas untuk para penggunanya untuk memodifikasi *library* agar dapat digunakan untuk mengenali beberapa bahasa (CMU, 2015).

Metode yang digunakan dalam PocketSphinx yaitu Hidden Markov Model. Proses pembelajaran unit-unit ucapan 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*.

2.2.2.1 Arsitektur PocketSphinx



Gambar 2.3 Arsitektur *Library* CMUSphinx
Sumber: http://cmusphinx.sourceforge.net

Gambar 2.3 merupakan arsitektur umum CMUSphinx yang terdapat komponen umum untuk menunjang hasil yang presisi pada saat pengujian aplikasi. Komponen tersebut terdiri dari *front end*, *decoder*, basis pengetahuan dan aplikasi itu sendiri.

Front end bertugas 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.

Basis pengetahuan 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.

Decoder berfungsi sebagai komponen utama pada aplikasi pengenalan ucapan. *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* (Hilman, 2014).

2.2.3 Query Matching

Query matching adalah metode untuk mencari informasi dalam bidang apapun berdasarkan *query* yang ditentukan sebelumnya. Metode ini merupakan salah satu pencarian teks tingkat tinggi yang membandingkan suatu *query* untuk

mencari informasi yang dibutuhkan. Ada 4 tahap yang dilakukan dalam *query matching* untuk mendapatkan mendapatkan informasi yaitu:

1. *Check the field type*

   Pada tahap ini dilakukan analisa pada seluruh dokumen informasi yang lengkap.

2. *Analyze the query string*

   Tahap ini melakukan analisa untuk menentukan *query* atau informasi yang ingin dicari dalam seluruh dokumen informasi.

3. *Find match docs*

   Tahap ini akan membandingkan *query* yang ditentukan sebelumnya dengan seluruh dokumen informasi yang ada.

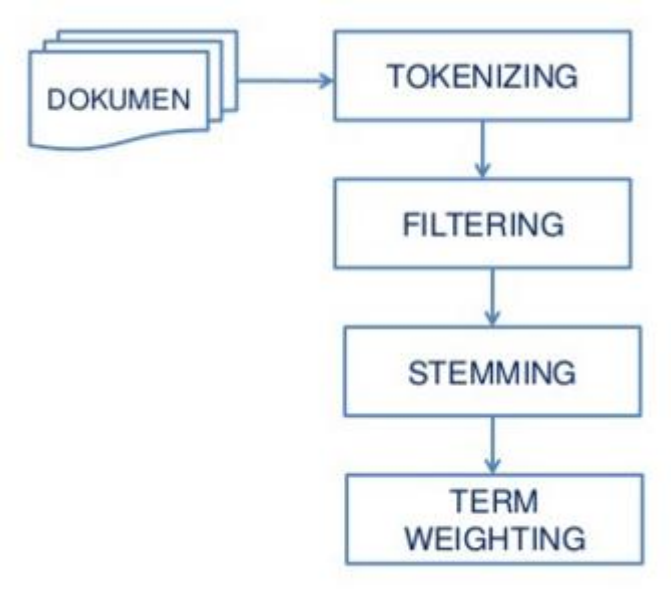
4. *Score each doc*

   Setelah membandingkan *query* pada seluruh dokumen informasi yang ada, kemudian dilakukan perhitungan untuk mencari bobot nilai yang paling besar dari setiap dokumen informasi. Dokumen informasi dengan bobot nilai paling besar adalah informasi yang dicari berdasarkan *query* yang telah ditentukan sebelumnya (Elastic, 2016).

2.2.4 Text Mining

Text mining merupakan penerapan konsep dan teknik *data mining* untuk mencari pola dalam teks. Proses analisa teks berguna untuk mencari informasi yang bermanfaat untuk tujuan tertentu. Proses *text mining* memerlukan beberapa tahapan karena data berupa teks memiliki karakteristik yang lebih kompleks dari pada data biasa.

Berdasarkan ketidakaturan struktur data teks, maka proses teks mining memerlukan beberapa tahapan awal yang pada intinya adalah mempersiapkan agar teks dapat diubah menjadi lebih terstruktur. Proses yang dilakukan adalah *preprocessing*, yaitu memproses kalimat agar menjadi *query-query* yang siap untuk dilakukan pembobotan. Berikut tahap untuk melakukan *preprocessing* yang dijelaskan pada gambar 2.4 (Raymond, 2006).



Gambar 2.4 *Preprocessing*
Sumber: (Dodik Suseno, 2014)

a. Tokenizing

Tahap ini memotong setiap kata pada teks dan mengubah semua huruf pada dokumen menjadi huruf kecil. *Tokenizing* melakukan pemotongan kata berdasarkan spasi pada kalimat (dokumen). Jadi hasil dari tahap *tokenizing* adalah kata-kata yang merupakan penyusun kalimat/*string* yang dimasukkan.

b. Stopword Removal / Filtering

Pada tahap ini dilakukan proses *filter* atau penyaringan kata hasil dari proses *tokenizing*, dimana kata yang tidak relevan dibuang. Proses ini menggunakan pendekatan *stoplist*. Beberapa kata yang masuk *stoplist* antara lain: yang, ini, itu, dengan, dari dan sebagainya.

c. Stemming

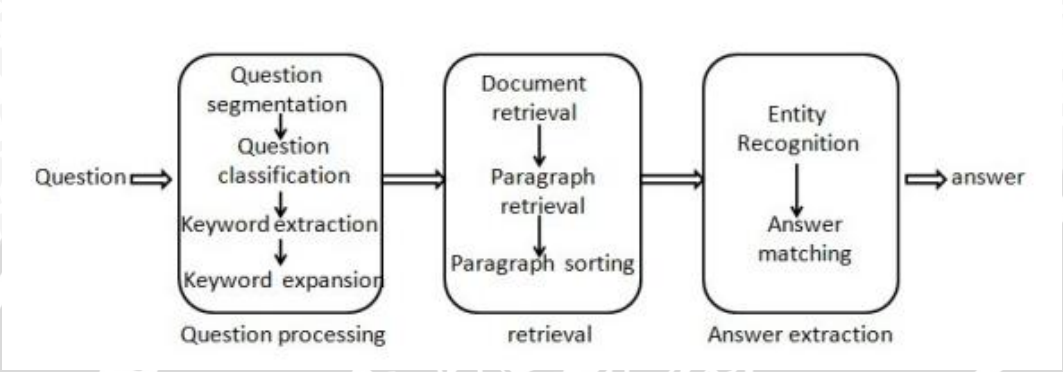
Dalam morfologi bahas Indonesia dikenal adanya 3 imbuhan yaitu awalan (prefiks), sisipan, dan akhiran (sufiks). Untuk tahap *stemming* pada penelitian ini hanya akan menghilangkan awalan dan akhiran, sehingga akan didapatkan kata dasar (Ahmad, 2010).

2.2.5 Keyword extraction

Keyword extraction merupakan salah satu teknik pada *Question Answering (QA)* yang melakukan analisa pada pertanyaan dengan menentukan beberapa kata untuk dijadikan keyword. Sedangkan *Question Answering (QA)* adalah salah satu sistem yang menggabungkan teknologi dari *natural language processing, information retrieval* (informasi temu kembali), *semantic analysis* dan *artificial intelligence* (kecerdasan buatan). *QA* berbeda dengan mesin pencari (*search engine*), *QA* hanya memberi respon dalam bentuk yang lebih spesifik dan alami, yaitu dalam bentuk *natural language sentence*. *QA* tidak mengembalikan respon berupa daftar dokumen, yang kemudian harus disaring lagi oleh user

untuk menentukan apakah dokumen-dokumen tersebut mengandung jawaban atas pertanyaan. *QA* memberikan teks singkat atau bahkan frase sebagai jawabannya.

Sebagai contoh, apabila user ingin mencari dimana mantan presiden Amerika Serikat, George Washington lahir, *user* dapat memberi input pertanyaan seperti "dimana George Washington lahir" pada *QA system*. Dan *output* dari pertanyaan berupa potongan teks atau frasa "Wesmoreland Country" sebagai jawaban. Berikut adalah arsitektur dari *QA* yang dijabarkan pada Gambar 2.5 (Guangzhi, 2013).



Gambar 2.5 Arsitektur Question Answering (QA) System
Sumber: (Guangzhi Zhang, 2013)

Dalam Arsitek *QA* yang dijabarkan dalam gambar 2.5 tersebut, tahap *keyword extraction* bertugas untuk mencocokkan kata yang sama antara dokumen latih dengan dokumen yang sudah dilatih. Untuk menentukan kelompok dokumen, masing-masing dokumen kemudian ditemukan bobot nilainya, kemudian diurutkan berdasarkan probabibiltas kebenarannya.

BAB 3 METODOLOGI

Bab ini membahas konsep teoritik berbagai metode yang dipilih sebagai metode yang digunakan dalam penilitian

3.1 Metode penelitian

Metode penelitian berisi prosedur atau tahapan yang ditempuh agar sistem yang dibangun menghasilkan respon dari pertanyaan berupa ucapan. Jika dapat memberikan respon dengan baik, dapat dikatakan sistem ini mengerti maksud ucapan. Berikut ini merupakan tahapan yang dilakukan untuk membangun sistem.

3.1.1 Studi literatur

Dalam perancangan penelitian, perlu dilakukan studi literatur yang menjelaskan dasar teori yang digunakan sebagai penunjang dan pendukung untuk membangun sistem pengenalan ucapan yang dapat mengerti maksud ucapan. Adapun yang digunakan sebagai bahan studi literatur untuk dapat merancang sistem meliputi:

1. Pengenalan ucapan
2. PocketSphinx
3. *Query Matching*
4. *Text mining*
5. *Keyword Extraction*

3.1.2 Analisis kebutuhan sistem

Analisis kebutuhan dilakukan untuk mengidentifikasi apa saja yang akan menjadi fitur dari sistem ini, sehingga sistem mampu memberikan respon dari pertanyaan yang berupa ucapan dengan memanfaatkan *library* PocketSphinx. Analisis kebutuhan juga dilakukan untuk mengetahui kondisi lapangan, sehingga dapat diketahui implementasi perangkat lunak dan perangkat keras untuk penelitian ini, agar sistem pengenalan ucapan dapat mengenali maksud ucapan dengan baik dan metode *query matching* dapan memberi respon dengan benar.

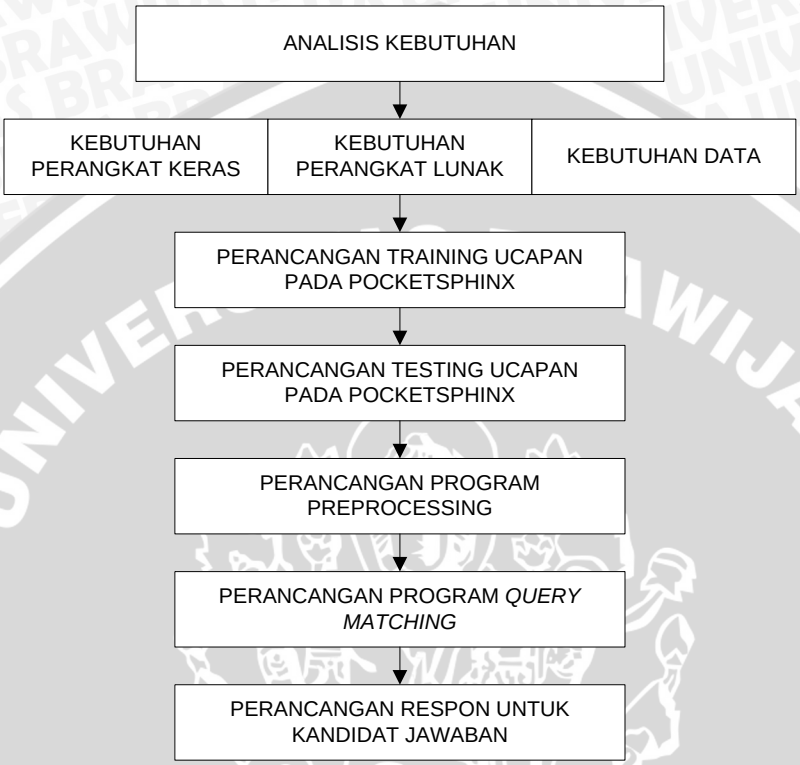
3.1.3 Diagram aktifitas

Diagram aktifitas bertujuan untuk menggambarkan representasi aktifitas dari masing-masing sub-sistem agar sistem ini dapat berjalan sesuai dengan analisis kebutuhan yang peneliti harapkan. Berikut adalah sub-sistem yang perlu dirancang diagram aktifitasnya:

1. Perancangan pengenalan ucapan dengan *library* PocketSphinx
2. Program pemrosesan text dengan teknik *preprocessing*
3. Program pencarian respon atau jawaban menggunakan metode *query matching*

3.1.4 Perancangan sistem

Setelah semua kebutuhan diperoleh dari tahap analisis tahap selanjutnya adalah perancangan perangkat lunak. Perancangan perangkat lunak berdasarkan pada diagram aktifitas yang telah dibuat. Berikut ini ialah tahapan-tahapan perancangan sistem pada penelitian ini yang dijelaskan pada Gambar 3.1.



Gambar 3.1 Perancangan sistem secara umum

Gambar 3.1 merupakan tahapan-tahapan yang dilakukan untuk merancang sistem sehingga sistem penjawab pertanyaan ini mampu memberikan respon dari pertanyaan yang berupa ucapan. Tahap pertama pada perancangan adalah *training* ucapan untuk melatih ciri suara dari rekaman ucapan. Setelah itu akan dilakukan proses testing untuk *speech to text* yang disimpan dalam *file txt*. Selanjutnya teks akan diproses menggunakan teknik *preprocessing* untuk dilakukan pengelompokkan menggunakan teknik *query matching*, setelah kelompok dokumen ditemukan, maka respon akan dihasilkan.

3.1.5 Implementasi

Implementasi sistem mengacu pada perancangan perangkat lunak yang dimulai dengan penjabaran spesifikasi lingkungan perancangan perangkat lunak. Implementasi perangkat lunak pada sistem ini dibagi menjadi 3 tahap yaitu:

1. Implementasi *Speech To Text*
2. Implementasi Program *Preprocessing*
3. Implementasi Program *Query Matching*

3.1.6 Pengujian

Pengujian pada penelitian ini dilakukan untuk mengetahui fungsi dari masing-masing perangkat lunak yang diimplementasikan apakah dapat berjalan dengan baik sesuai dengan analisis kebutuhan sistem. Pengujian pada sistem dilakukan dengan mengacu pada perancangan sistem, yang meliputi:

1. Pengujian *speech to text* pada *library* PocketSphinx untuk menghasilkan teks yang sesuai dengan ucapan.
2. Pengujian metode *query matching* untuk menghasilkan respon yang benar berdasarkan teks pertanyaan yang telah dibuat.
3. Pengujian keseluruhan sistem untuk menghasilkan respon dari pertanyaan berupa ucapan.

3.1.7 Analisis

Analisis dilakukan setelah mengetahui hasil dari 3 pengujian yang telah dilakukan untuk menarik kesimpulan dari penelitian yang dilakukan. Hasil pengujian dianalisis untuk menentukan apakah sistem dapat menghasilkan respon dengan baik sehingga dapat dikatakan sebagai sistem pengenalan ucapan yang dapat mengenali maksud ucapan sesuai dengan tujuan yang diharapkan.

3.1.8 Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi dan pengujian perangkat lunak yang dibangun telah selesai. Kesimpulan diambil dari hasil pengujian dan analisis terhadap sistem yang dibangun. Kesimpulan diambil untuk mengetahui kesesuaian *library* PocketSphinx untuk mengenali ucapan dan program pencarian respon dengan metode *query matching* terhadap perancangan yang telah dibuat. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi dalam implementasi dan penulisan dalam penelitian serta sebagai acuan untuk pengembangan sistem selanjutnya.

BAB 4 PERSYARATAN

Bab ini menjelaskan mengenai seluruh kebutuhan untuk menunjang penelitian, serta menjelaskan tahapan-tahapan aktifitas yang dilakukan oleh user, perangkat keras, maupun perangkat lunak dalam pengerjaan penelitian.

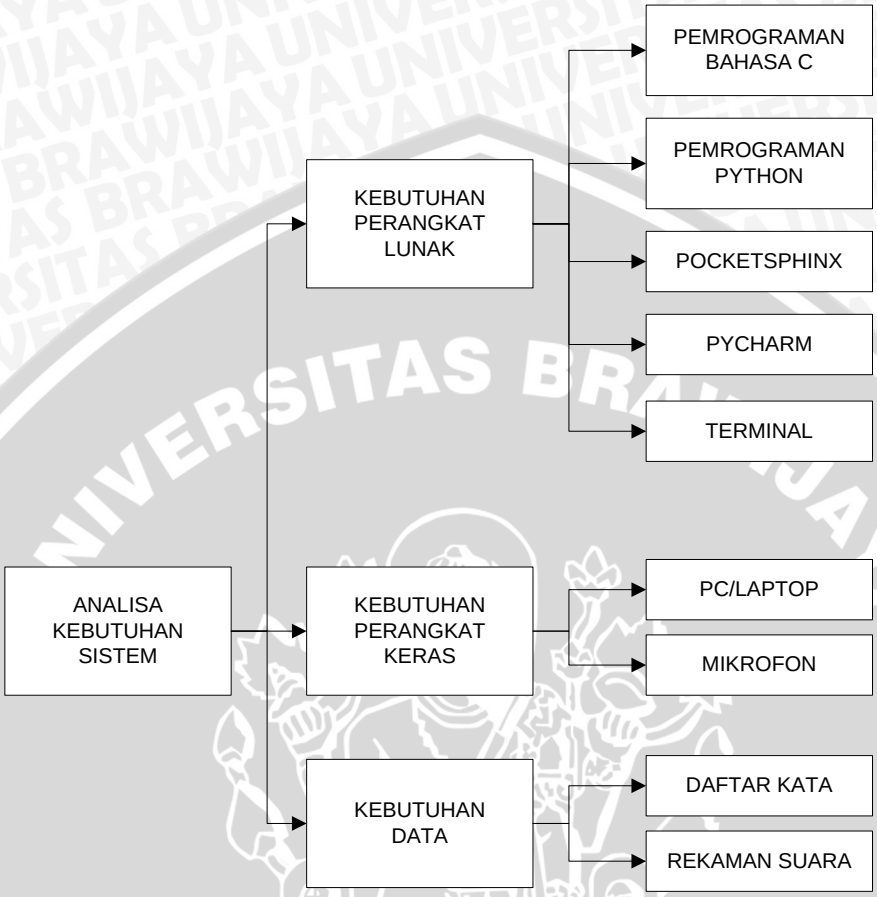
4.1 Analisis kebutuhan sistem

Dalam sistem penjawab pertanyaan ini, ada beberapa fitur yang ingin peneliti capai sehingga sehingga sistem ini mampu memberikan respon dari pertanyaan yang berupa ucapan dalam bahasa Indonesia. Yang pertama adalah *speech to text* untuk merubah pertanyaan yang berupa ucapan dalam bahasa Indonesia menjadi bentuk teks dengan menggunakan *library* pocketSphinx. Untuk memenuhi fitur tersebut, dibutuhkan *library* PocketSphinx untuk melakukan *training* ucapan, selain itu dibutuhkan juga data seluruh kata, fonem dalam bahasa Indonesia, dan rekaman ucapan yang digunakan dalam pertanyaan. *Microphone* diperlukan untuk melakukan rekaman ucapan, sehingga rekaman yang menghasilkan *noise* yang kecil, sehingga *training* ucapan akan memberikan akurasi yang tinggi untuk menghasilkan *acoustic model*. Setelah proses *training* selesai, akan dirancang program *testing* ucapan. *Testing* ucapan bertujuan untuk mencoba apakah ucapan yang diucapkan menghasilkan ketepatan yang tinggi dalam merubah ucapan kedalam bentuk teks. Program *testing* ucapan dirancang dengan menggunakan bahasa C, tujuan menggunakan bahasa C yaitu karena menyesuaikan *library* PocketSphinx yang peneliti gunakan. Perancangan pada program *speech to text* akan dilakukan pada notebook dengan sistem operasi Linux, hal ini berdasarkan kemudahan sistem operasi Linux terhadap hubungan dengan mesin (*embedded system*).

Fitur yang kedua adalah penjawab pertanyaan, teks pertanyaan diambil dari *speech to text* yang kemudian diproses menggunakan teknik *preprocessing*. Setelah semua teks terproses, teks dikelompokkan dokumennya dengan menggunakan metode *query matching*. Untuk memenuhi fitur tersebut, peneliti merancang program penjawab pertanyaan ini dengan menggunakan pemrograman python yang dilakukan melalui *IDE* Pycharm, karena python memiliki tata bahasa yang mudah dipelajari dan memiliki aturan *layout source* yang memudahkan untuk pengecekan, pembacaan kembali dan penulisan ulang *source code* tersebut.

Selain analisis kebutuhan untuk perancangan, dibutuhkan juga analisis kebutuhan untuk melakukan testing pada keseluruhan sistem. Agar sistem ini memberikan respon yang benar dengan tingkat akurasi yang baik, maka dibutuhkan *microphone* untuk user jika akan menanyakan pertanyaan. *Microphone* akan mempengaruhi pengenalan ucapan karena dapat meminimalkan noise pada ucapan yang masuk pada *library* PocketSphinx, sehingga proses pada fitur *speech to text* akan menghasilkan teks yang sesuai dengan akurat. Selain itu, dibutuhkan juga media untuk menampilkan jawaban atau respon yang dijadikan keluaran pada sistem ini. Respon ditampilkan melalui

terminal yang tersedia pada sistem operasi Linux, jadi sistem ini dijalankan pada *notebook* dengan sistem operasi Linux. Gambaran umum mengenai seluruh kebutuhan sistem pada penelitian ini dapat dilihat pada Gambar 4.1.



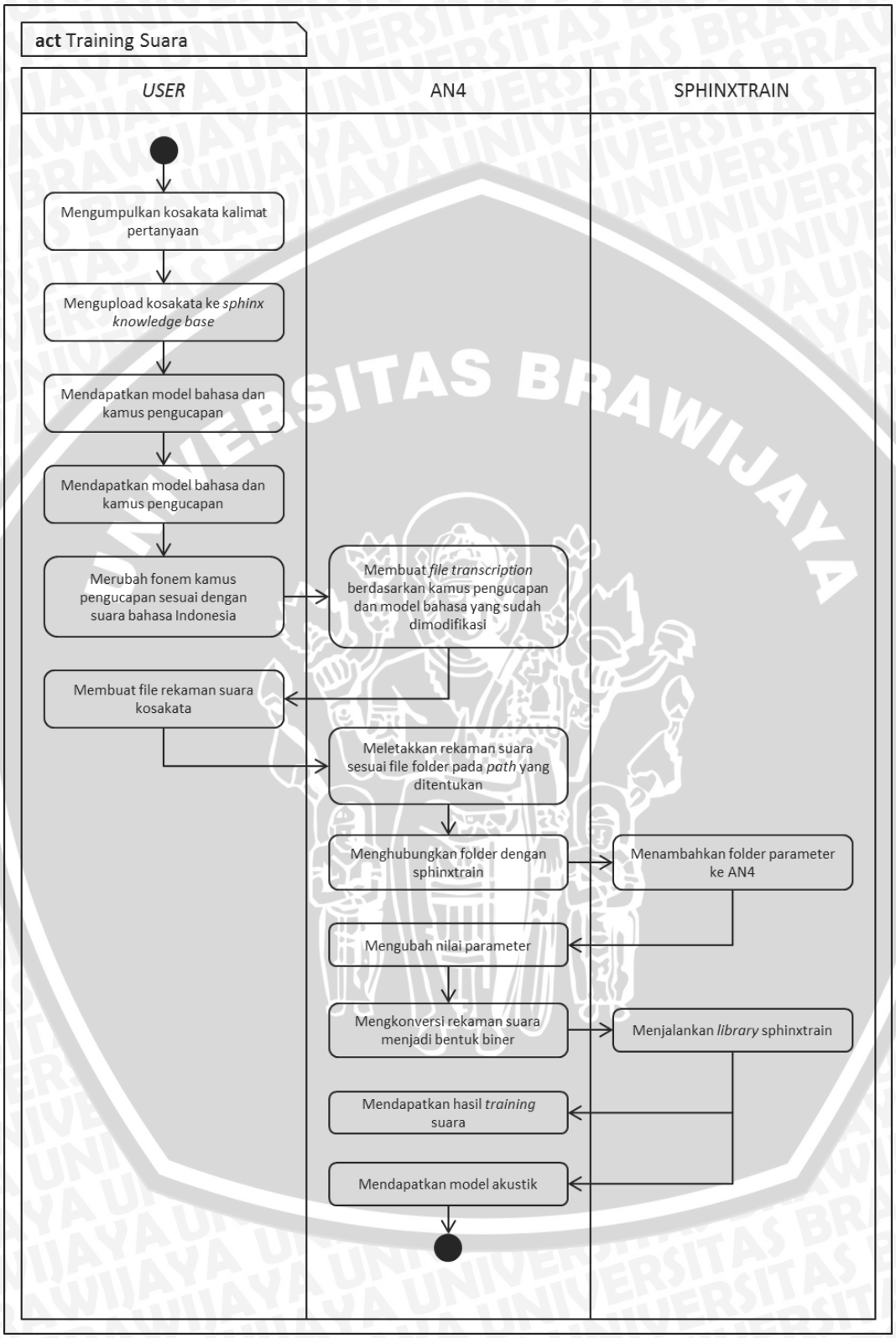
Gambar 4.1 Diagram kebutuhan sistem

Gambar 4.1 merupakan seluruh kebutuhan yang peneliti gunakan untuk merancang dan menjalankan sistem penjawab pertanyaan dengan menggunakan *speech recognition*. Analisis diatas diatas dirancang berdasarkan fitur yang ingin dicapai pada masing-masing sub-sistem, sehingga sistem dapat menghasilkan respon yang benar sesuai dengan pertanyaan.

4.2 Diagram aktivitas

Diagram aktivitas merupakan representasi dari seluruh tahapan sistem saat melakukan aktivitas perancangan perangkat lunak. Diagram aktivitas mendeskripsikan aliran kerja dari perilaku sistem, sehingga dapat membantu memahami proses sistem secara keseluruhan. Untuk merancang sistem ini, ada 4 aktivitas yang dilakukan yaitu *training* ucapan, *testing* ucapan, *preprocessing*, dan program *query matching*. Dibawah ini adalah penjabaran mengenai diagram aktifitas yang dilakukan oleh penulis.

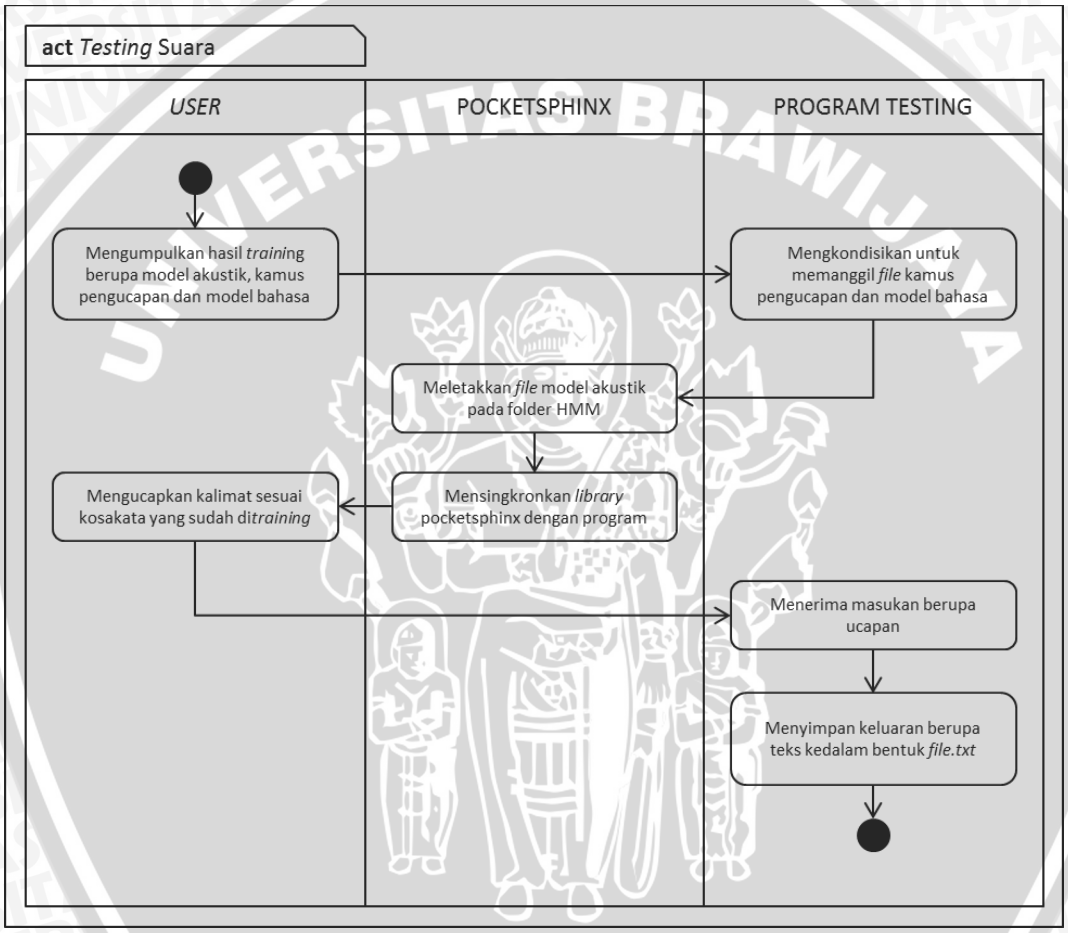
4.2.1 Diagram aktifitas *training* ucapan



Gambar 4.2 Diagram aktifitas untuk *training* ucapan

Gambar 4.2 menjabarkan proses yang terjadi pada saat implementasi *training* ucapan yang dijadikan masukan pertanyaan. *Training* ucapan menggunakan *library* SphinxTrain dan database AN4 sebagai tempat rekaman file ucapan dan *file transcription* atau daftar kosakata. Peneliti menggunakan rekaman ucapan berformat .wav yang disesuaikan dengan *file transcription* dan menyiapkan kamus pengucapan yang disesuaikan dengan fonem pengucapan bahasa Indonesia. Tujuan dari dilakukannya *training* ucapan adalah untuk mendapatkan model akustik atau ciri suara pengguna.

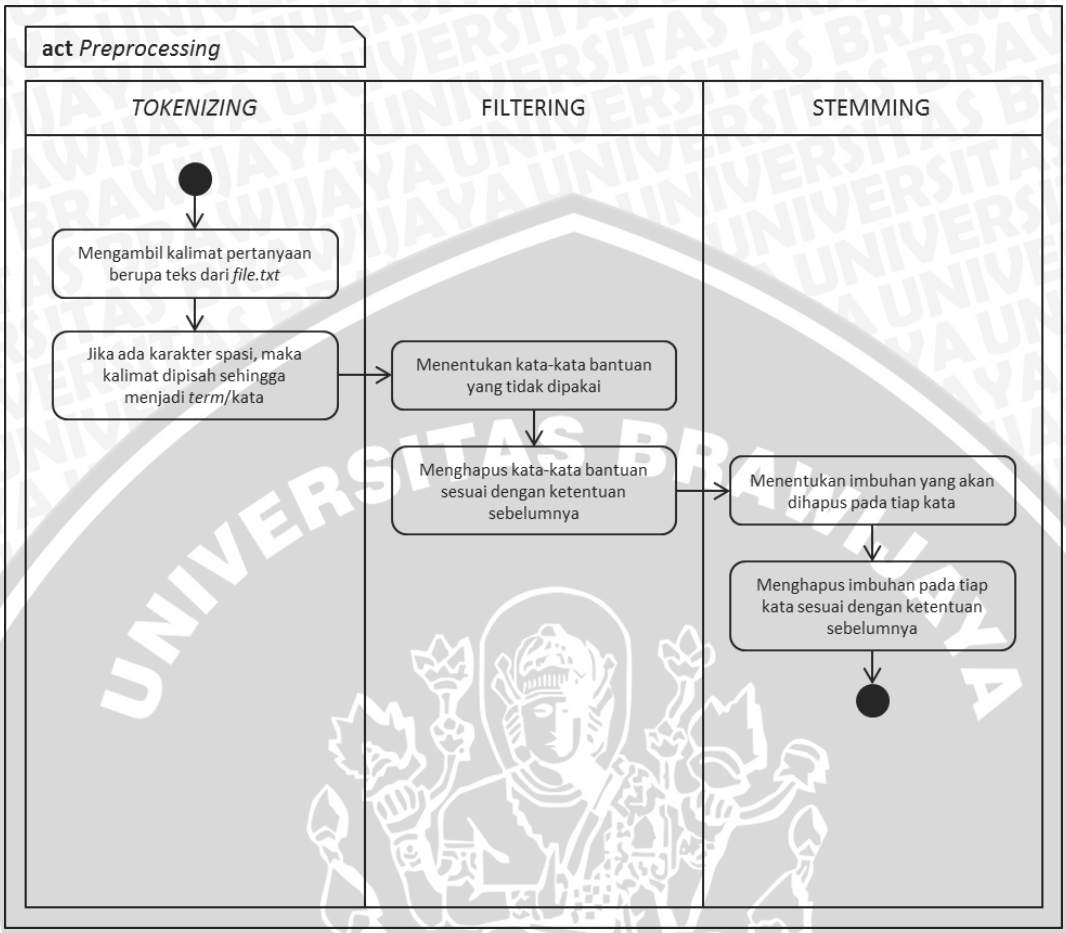
4.2.2 Diagram aktifitas *testing* ucapan



Gambar 4.3 Diagram aktifitas *testing* ucapan

Gambar 4.3 menjabarkan proses untuk melakukan *testing* ucapan bahasa Indonesia yang digunakan sebagai kalimat pertanyaan. *Testing* ucapan dilakukan dengan menggunakan *library* PocketSphinx dan program *testing*. Program *testing* mengkondisikan menyimpan keluaran berupa teks kedalam *file* berformat .txt untuk selanjutnya diolah oleh program klasifikasi pada aktifitas selanjutnya.

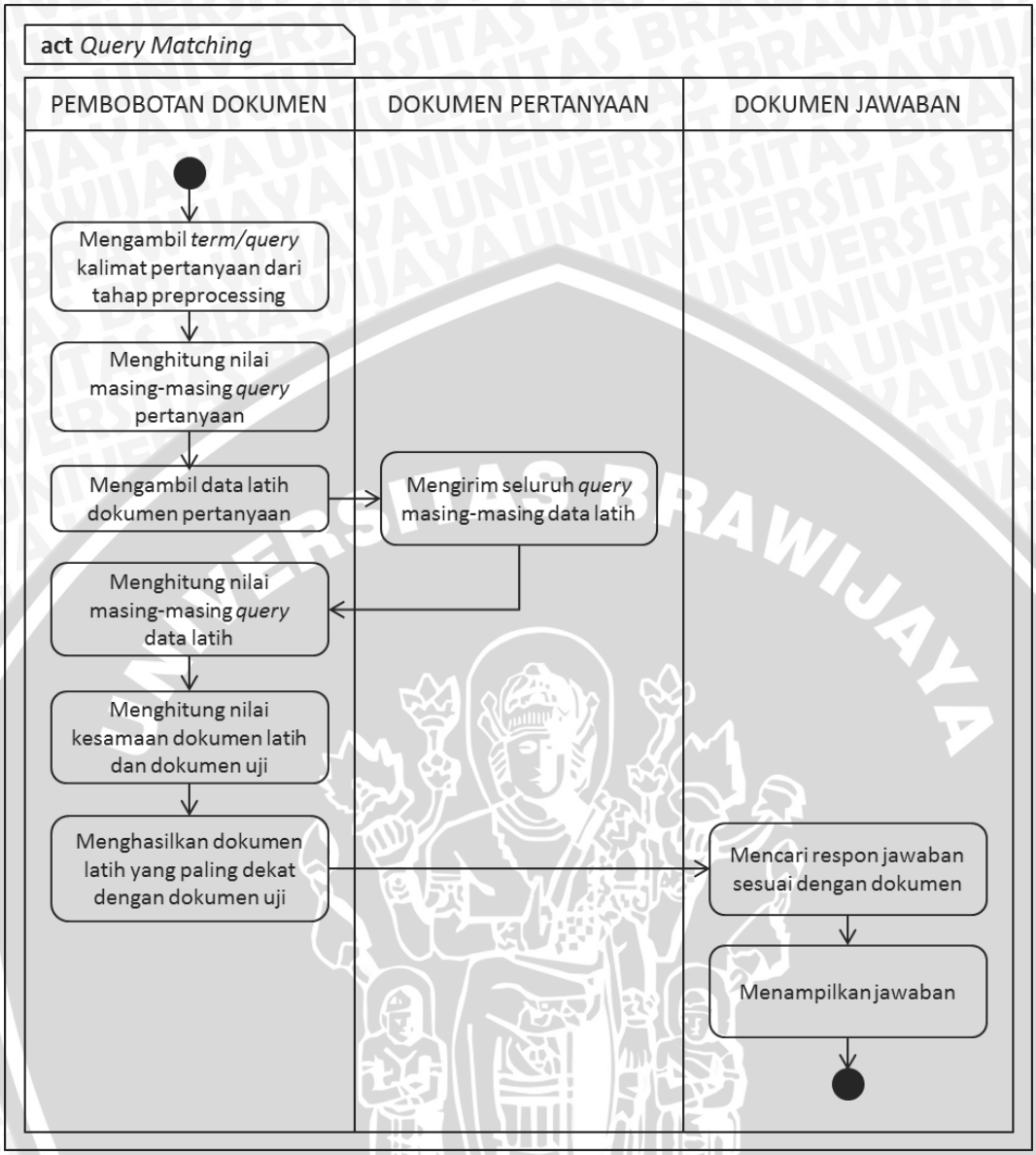
4.2.3 Diagram aktifitas *preprocessing*



Gambar 4.4 Diagram aktifitas *preprocessing*

Gambar 4.4 menjabarkan mengenai pengolahan kata yang dinamakan *preprocessing*. Pada tahap ini, dilakukan proses pemisahan kata, menghapus kata-kata bantuan, dan menghapus imbuhan.

4.2.4 Diagram aktifitas program *query matching*



Gambar 4.5 Diagram aktifitas program *query matching*

Gambar 4.5 menjabarkan tahapan yang dilakukan untuk mencari respon atau jawaban dari pertanyaan. Pada tahap ini, dilakukan proses perhitungan nilai dokumen, mencari nilai kesamaan, dan menampilkan respon jawaban.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

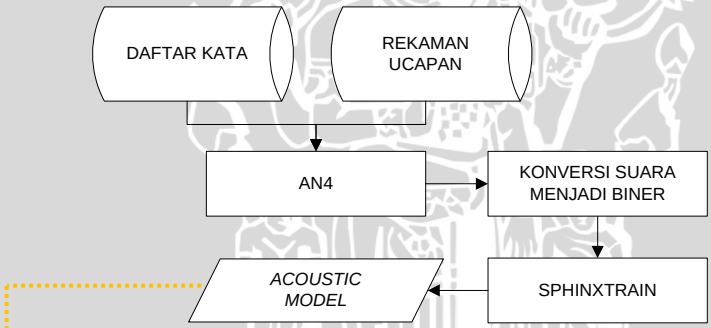
Bab ini membahas proses penerapan berbagai metode dan prinsip yang bertujuan untuk mendefinisikan sistem yang dibangun. Pada tahap ini dilakukan perancangan sistem pengenalan ucapan menggunakan *library* PocketSphinx untuk merubah kalimat pertanyaan yang berupa ucapan ucapan menjadi teks yang nantinya dapat diolah pada tahap *preprocessing* dan dilanjutkan dengan *query matching* untuk menentukan respon dari pertanyaan.

5.1 Perancangan

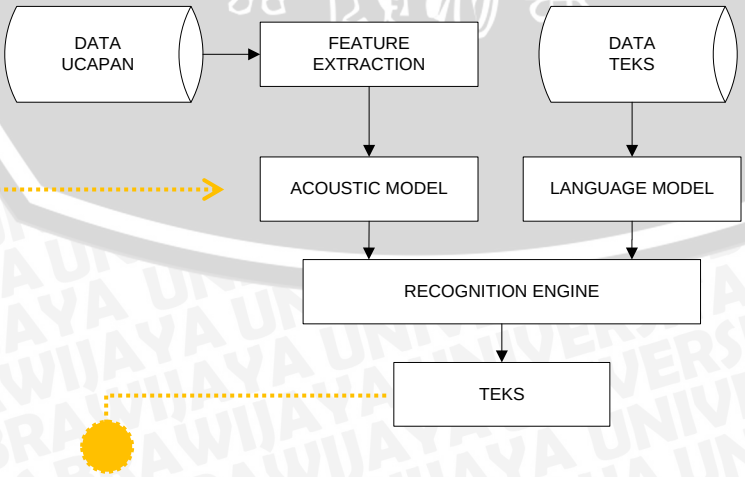
Penelitian ini membahas mengenai penerapan metode *query matching* untuk menentukan respon dari pertanyaan yang berupa ucapan dengan menggunakan *library* pengenal ucapan PocketSphinx. Untuk membangun sistem tersebut, perlu dilakukan tahap perancangan yang terdiri dari:

5.1.1 Diagram blok sistem

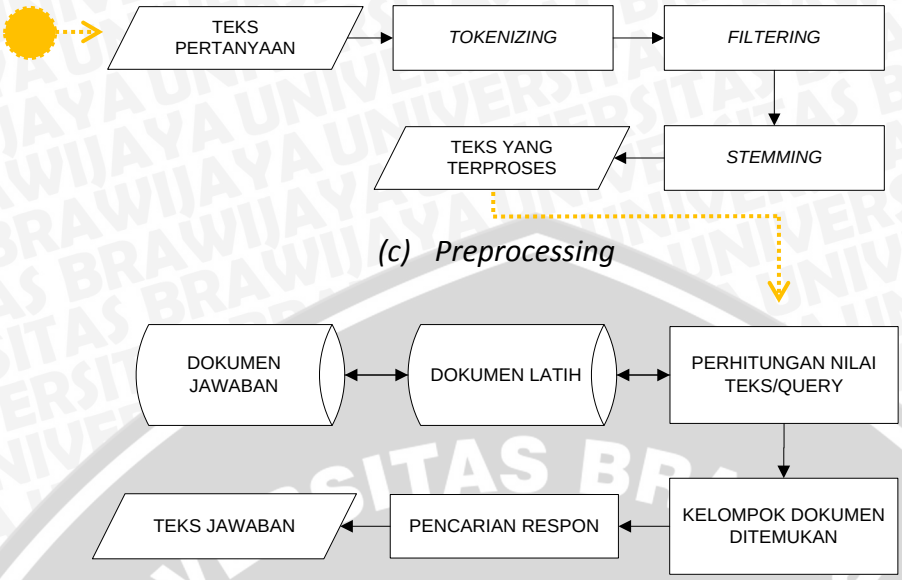
Dari analisis kebutuhan sistem yang telah dirancang, perlu dibuat diagram blok agar proses perancangan sistem terarah dan terstruktur. Perancangan sistem secara keseluruhan dapat digambarkan melalui diagram blok seperti pada Gambar 5.1.



(a) *Training* ucapan pada PocketSphinx



(b) *Testing* ucapan pada PocketSphinx



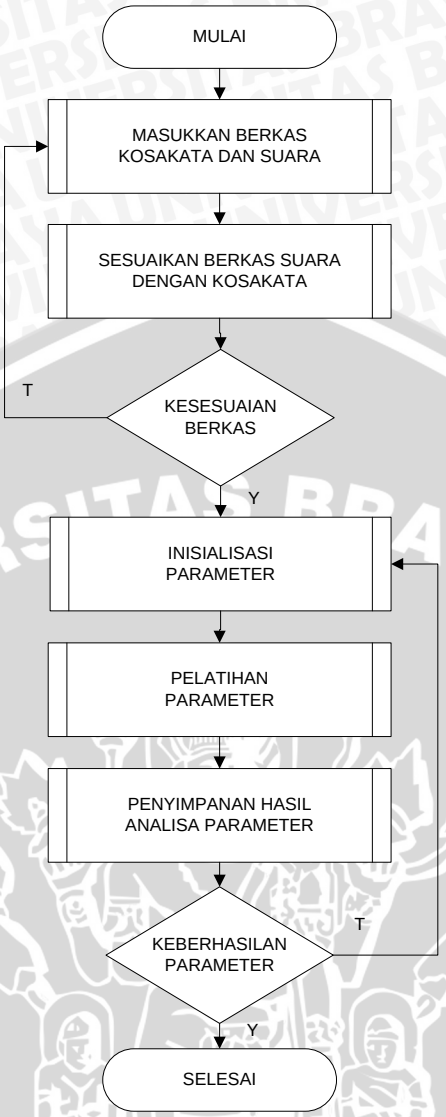
Gambar 5.1 Diagram blok sistem

Gambar 5.1 merupakan penjabaran mengenai tahapan-tahapan pada masing-masing sub-sistem sehingga sistem dapat memberikan respon jawaban yang benar dari pertanyaan yang berupa ucapan. Pada diagram blok yang telah dibuat, menggambarkan bahwa pertanyaan berupa ucapan adalah sebagai *input* untuk kemudian dilakukan pengenalan ucapan untuk ditampilkan berupa teks melalui *library* PocketSphinx. Pertanyaan berupa teks kemudian di-*preprocessing* untuk menjadi teks yang siap untuk diproses pada tahap *query matching*.

Sebelumnya, dokumen latih berupa daftar pertanyaan yang sudah dikelompokkan dan dokumen jawaban berupa daftar jawaban telah dirancang. Kedua dokumen tersebut saling terhubung antara pertanyaan dan jawabanya, sedangkan teks yang terproses dianggap sebagai dokumen baru. *Query matching* membandingkan antara dokumen latih dan dokumen baru untuk mencari kata atau *query* yang cocok. Hasil dari perhitungan nilai teks digunakan untuk menentukan kelompok dokumen dengan menggunakan teknik *keyword extraction*. Jika sudah ditemukan kelompok dokumennya, maka akan dihasilkan jawaban yang sudah ditentukan sebelumnya.

5.1.2 Perancangan *training* pada PocketSphinx

Training ucapan pada PocketSphinx dilakukan untuk mengenali berbagai ciri ucapan manusia dalam pengucapan bahasa Indonesia. *Training* ucapan akan mempengaruhi akurasi dalam *testing* pengenalan ucapan bahasa Indonesia, semakin baik *training* yang dilakukan, maka semakin akurat pula pengenalan ucapan untuk mengubahnya ke dalam teks. Pada Gambar 5.2 berikut merupakan tahap perancangan *training* ucapan yang dilakukan.



Gambar 5.2 Diagram alir perancangan *training* ucapan

Gambar 5.2 merupakan alur pada saat *training* ucapan yang bertujuan untuk melatih ciri suara pada teks, *training* ucapan akan menghasilkan *acoustic model* yang nantinya dipakai untuk *testing* ucapan. Dibawah ini merupakan penjabaran mengenai tahapan-tahapan yang dijalankan pada saat training ucapan menggunakan *library* PocketSphinx.

a. Pengumpulan berkas kosakata dan ucapan

Peneliti mengumpulkan berkas ucapan dan kosakata yang digunakan untuk melakukan *training* pada PocketSphinx, berkas yang digunakan antara lain:

1. *Phonetic dictionary* dengan ekstensi .dic
2. *Phoneset file* dengan ekstensi .phone
3. *Transcript file* dengan ekstensi .transcription
4. Daftar *file* untuk training dengan ekstensi .fileids
5. *Filler dictionary* dengan ekstensi .fillers
6. *File* ucapan dengan format wav 16 kHz, 16 bit, mono.

b. Penyesuaian berkas ucapan dan kosakata

Berkas dalam rekaman ucapan disesuaikan dengan daftar kosakata yang digunakan untuk pengenalan ucapan dalam bahasa Indonesia. Ketika berkas yang digunakan tidak sesuai satu sama lain, proses *training* akan gagal.

c. Inisialisasi parameter

Pada tahap ini akan dilakukan inisialisasi parameter berdasarkan kosakata yang diimplementasikan dalam pengenalan ucapan. Parameter yang digunakan pada program yaitu *lowerf, uppers, nfft* dan *nfilt*. Pemilihan parameter ini digunakan sebagai dasar perhitungan untuk hasil latihan kosakata yang sudah diimplementasikan pada program.

d. Pelatihan parameter

Pelatihan parameter dilakukan untuk mendapatkan estimasi *error* terkecil, pelatihan parameter dilakukan dengan cara mengubah nilai *nfft/nfilt* dan *upperf/lowerf* sehingga menghasilkan akurasi terkecil pada *Word Error Rate* (*WER*).

e. Penyimpanan hasil analisa parameter

Penyimpan parameter diperlukan karena berfungsi untuk proses pengenalan kata. Setelah menghasilkan parameter dengan nilai *error* terkecil, parameter tersebut disimpan dan di-*push* bersama *language model* untuk pengenalan *speech to text*.

5.1.3 Perancangan program *testing* pada PocketSphinx

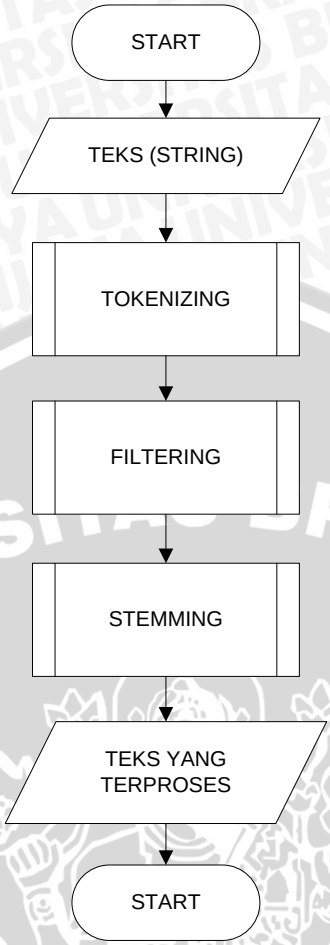
Perancangan program *testing* berguna untuk mencoba pengenalan ucapan bahasa Indonesia sekaligus mengetahui ketepatan pengenalan ucapan yang diubah ke dalam teks melalui *library* PocketSphinx. Ucapan yang dimasukkan disesuaikan dengan kamus pengucapan berdasarkan fonem bahasa Indonesia. Proses masukan ucapan dipengaruhi oleh hasil model akustik yang didapat dari hasil *training* ucapan. Dari hasil *training* ucapan tersebut menghasilkan model parameter yang digunakan sebagai optimasi terhadap pengenalan ucapan. Proses yang dilalui dalam mengenali ucapan sehingga menghasilkan keluaran berupa teks mempunyai beberapa tahapan sebagai berikut.

Gambar 5.3 Diagram alir perancangan program *testing*

Pada Gambar 5.3 menunjukkan melalui program pengenal ucapan, user akan mengucapkan kata atau kalimat dalam bahasa Indonesia. Ucapan tersebut dicocokkan dengan hasil yang didapat dari *training* pada SphinxTrain, jika pengenalan ucapan saat *testing* akurat, maka program *testing* akan menampilkan teks dari ucapan tersebut dengan benar.

5.1.4 Perancangan program *preprocessing*

Perancangan *preprocessing* bertujuan untuk memproses kalimat berupa teks yang didapat dari pengenalan ucapan melalui program *testing* pada PocketSphinx. Berikut tahapan dalam melakukan proses *preprocessing* yang dijelaskan pada Gambar 5.4.

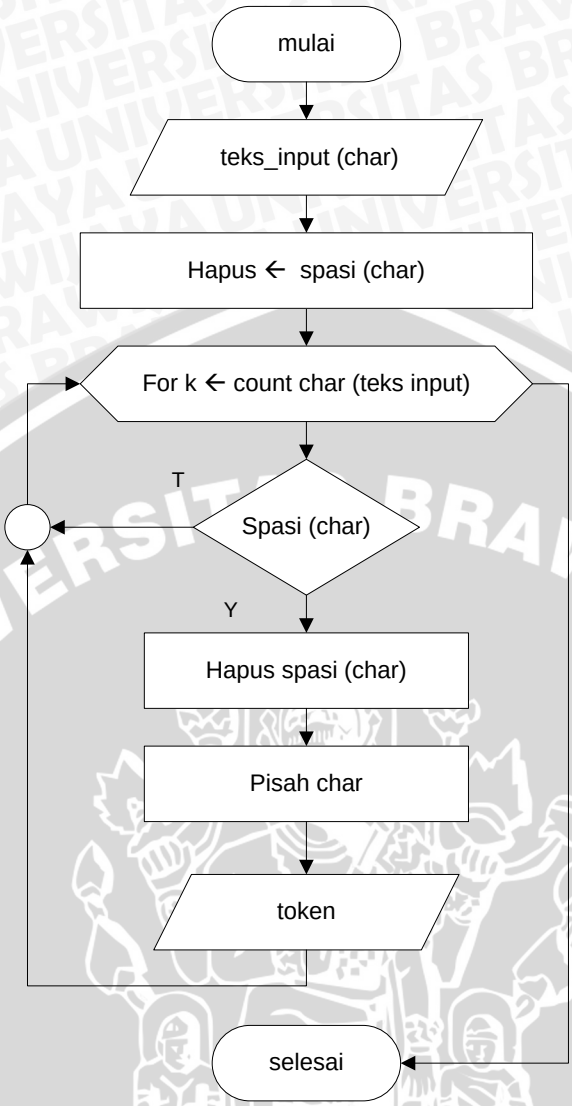


Gambar 5.4 Diagram alir *preprocessing*

Pada Gambar 5.4 diatas menjelaskan langkah yang dilakukan pada saat *preprocessing* kalimat pertanyaan yang didapat dari pengenalan ucapan menggunakan PocketSphinx. Kalimat tersebut dipisah setiap kata dengan cara menghapus karakter spasi pada tahap *tokenizing*, maka akan didapatkan potongan-potongan kata. Dari potongan kata tersebut, kata bantuan misalkan yang, dengan, ini, itu, dan sebagainya akan dihapus pada tahap *filtering*. Proses terakhir adalah menghapus imbuhan yang bertujuan untuk mencari kata dasar pada tahap *stemming*. Setelah ketiga proses tersebut dilakukan, maka dihasilkan kata-kata penting pada kalimat yang digunakan untuk pengklasifikasian dokumen berupa kalimat.

5.1.4.1 Tokenizing

Proses pertama yang dilakukan untuk *preprocessing* adalah memecah kalimat menjadi *token*/kata berdasarkan spasi yang biasa disebut dengan *tokenizing*. Berikut diagram alir pada proses *tokenizing* yang ditunjukkan pada Gambar 5.5.

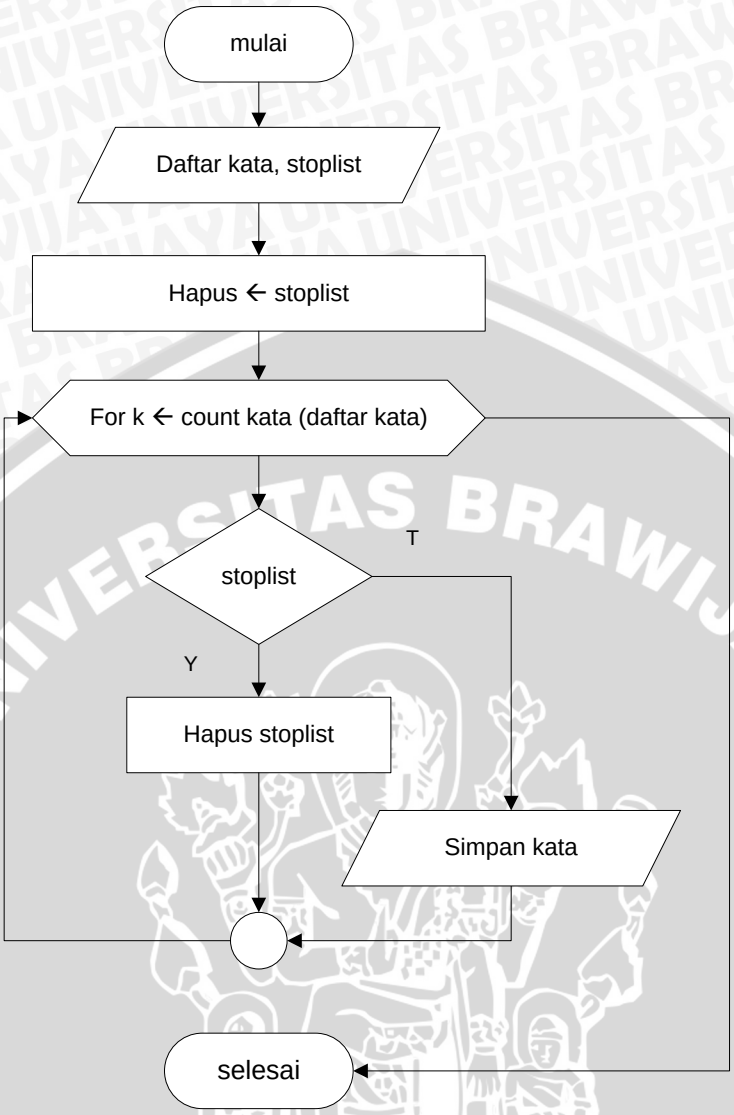


Gambar 5.5 Diagram alir *tokenizing*

Gambar 5.5 menjelaskan menegenai tahap *tokenizing* untuk melakukan pemisahan kalimat menjadi token-token. Masukan teks diambil dari file .txt pada tahap *speech to text* atau pengenal ucapan. Pemisahan token berdasarkan pada karakter spasi, jadi setiap karakter pada kalimat akan diperiksa dari karakter awal hingga karakter akhir, jika ada karakter spasi, maka kalimat akan dipisah menjadi token.

5.1.4.2 Stopword removal/filtering

Stopword removal atau *filtering* adalah proses untuk menghilangkan kata-kata bantuan dari token yang telah dihasilkan pada tahap *tokenizing* sebelumnya. Gambar 5.6 menunjukkan diagram alir *stopword removal*.

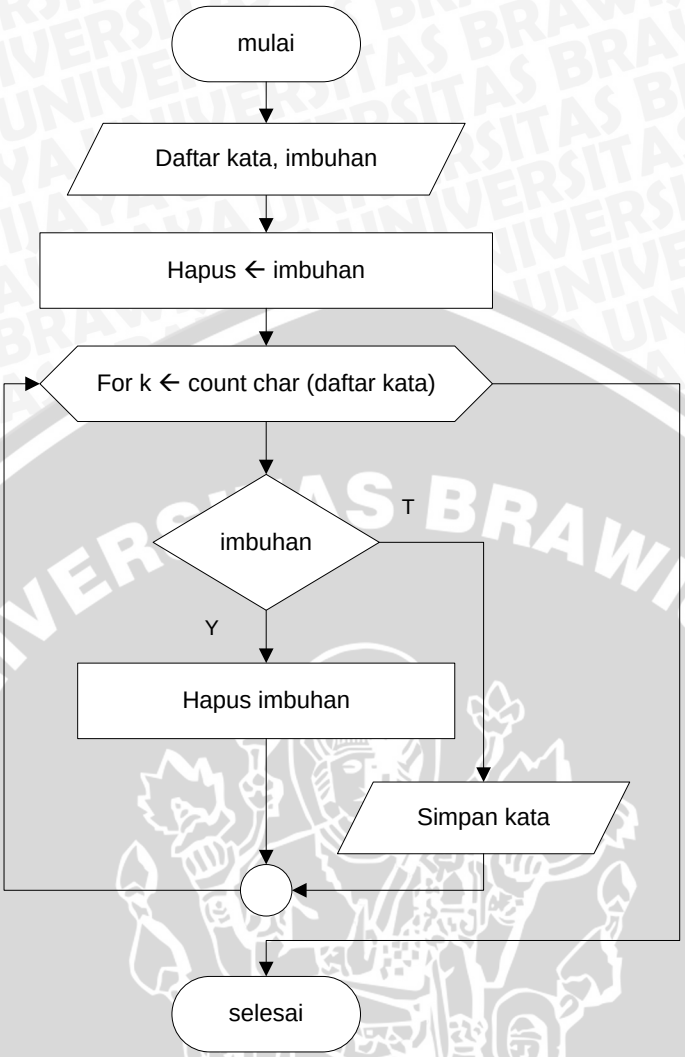


Gambar 5.6 *Stopword Removal*

Gambar 5.6 menjelaskan tentang tahap *stopword removal* yang bertujuan untuk menghilangkan kata-kata bantuan seperti yang, ini, itu, dan, dengan, dan sebagainya. Masukan pada tahap ini diambil dari hasil *tokenizing* yang sebelumnya dilakukan, selain itu diperlukan pendefinisian kata bantuan yang nantinya akan dihapus. Setiap kata (token) akan diperiksa, jika ada kata yang dianggap sebagai kata bantuan, maka kata tersebut akan dihapus.

5.1.4.3 Stemming

Stemming dilakukan untuk mendapatkan kata dasar dari sebuah term dengan cara menghapus imbuhan dari term. Berikut diagram alir *stemming* yang ditunjukkan pada Gambar 5.7.



Gambar 5.7 Diagram alir *stemming*

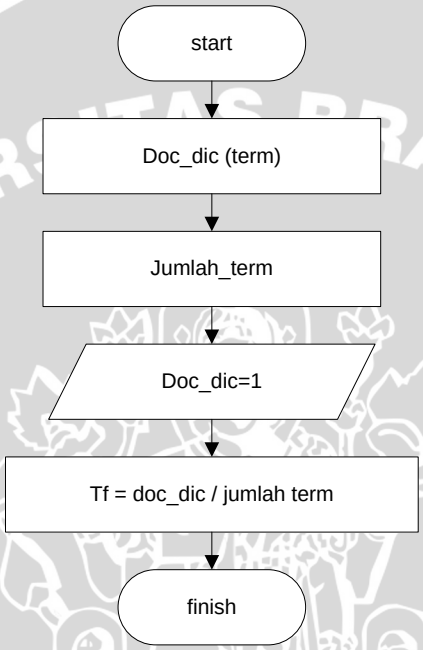
Gambar 5.7 menjelaskan proses yang dilakukan untuk menghapus imbuhan atau bisa dianggap juga merubah menjadi kata dasar, sehingga pada tahap ini diharapkan menghasilkan kata-kata penting yang siap untuk dilakukan proses pembobotan kata. Daftar kata diambil dari keluaran pada tahap yang sebelumnya yaitu *stopword removal,* selain itu diperlukan juga daftar imbuhan yang akan dihapus. Setiap kalimat akan diperiksa setiap karakter, jika dalam karakter ada yang dianggap sebagai imbuhan yang sebelumnya telah didefinisikan, maka imbuhan tersebut akan dihapus.

5.1.5 Perancangan program *query matching*

Perancangan program query matching bertujuan untuk mencari kelompok dokumen. Metode ini mengelompokkan dokumen berdasarkan kata yang sama atau cocok antara dokumen latih dan dokumen uji sehingga antar dokumen dapat menghasilkan nilai bobot nilai perbandingan antar dokumen. Untuk menghitung bobot nilai dokumen, peneliti menggunakan teknik *keyword extraction*, semakin banyak *query* yang cocok, maka semakin besar bobot

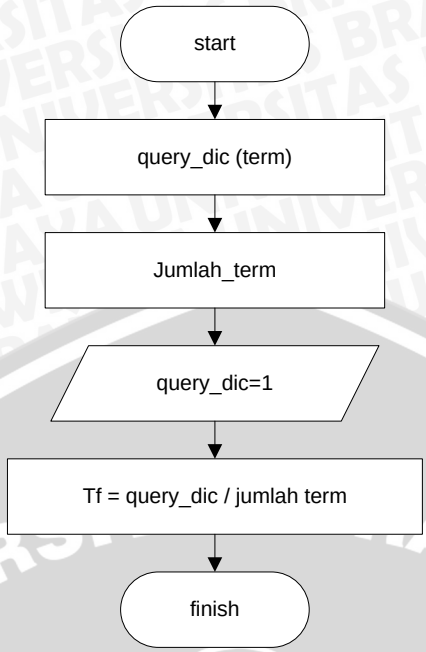
dokumen tersebut. Dokumen latih dengan nilai kedekatan paling besar adalah kelompok dimana dokumen baru diklasifikasikan. Pertanyaan didefinisikan sebagai dokumen baru, sedangkan pertanyaan yang sudah dikelompokkan sebelumnya didefinisikan sebagai dokumen latih.

Tahap pertama adalah melakukan perhitungan terhadap nilai *query* pada dokumen latih. Perhitungan *query* adalah dengan membagi nilai 1 dengan jumlah kata yang ada dalam dokumen. Nilai 1 adalah nilai normalisasi untuk sebuah dokumen atau kalimat. Gambar 5.8 menunjukkan diagram alir dalam proses perhitungan nilai *query* pada dokumen latih.



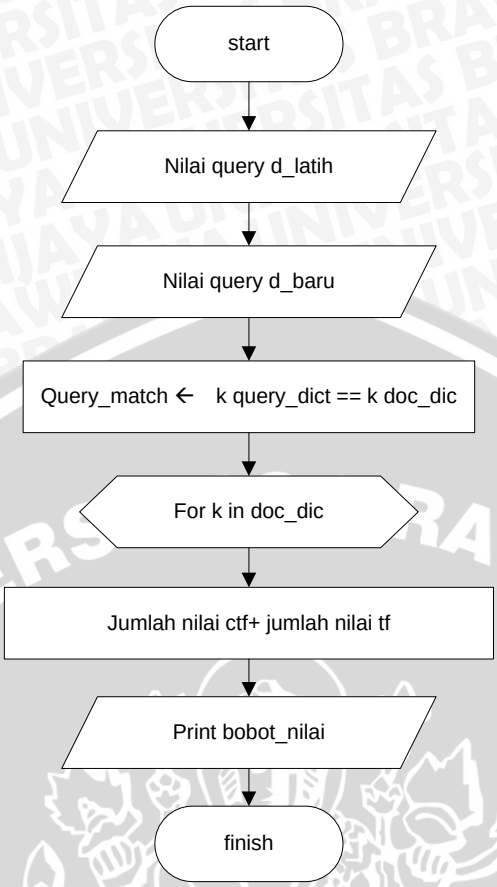
Gambar 5.8 Diagram alir perhitungan *query* dokumen latih

Gambar 5.8 merupakan tahapan untuk menghasilkan nilai query atau kata pada dokumen latih. Selanjunya akan dilakukan perhitungan nilai *query* pada dokumen baru, perhitungan yang dilakukan untuk mencari nilai *query* pada dokumen baru sama dengan perhitungan *query* pada dokumen latih. Berikut diagram alir pada proses perhitungan *query* pada dokumen baru yang ditunjukkan oleh Gambar 5.9.



Gambar 5.9 Diagram alir perhitungan *query* dokumen baru

Gambar 5.8 merupakan tahapan untuk menghasilkan nilai query atau kata pada dokumen latih. Untuk mencari nilai bobot kedekatan seluruh dokumen latih terhadap dokumen baru, maka dicari *query matching* atau kata yang cocok antara dokumen baru dengan tiap dokumen latih. Semakin banyak *query* yang cocok yang dimiliki dokumen latih terhadap dokumen baru, maka semakin besar pula bobot nilai dokumen latih. Setelah itu masing-masing nilai *query* dijumlahkan, maka dihasilkan bobot nilai kedekatan. Bobot nilai dokumen mempengaruhi pengelompokan dokumen, jadi dokumen latih yang mempunyai bobot nilai yang paling besar adalah dokumen dimana dokumen baru diklasifikasikan. Berikut diagram alir dari proses perhitungan bobot nilai dokumen yang dijelaskan pada Gambar 5.10.



Gambar 5.10 Diagram alir perhitungan bobot nilai dokumen

Gambar 5.10 menjabarkan proses untuk menghitung bobot nilai seluruh dokumen latih dengan membandingkan query pada dokumen baru terhadap seluruh dokumen latih.

5.1.6 Perancangan respon untuk kandidat jawaban

Perancangan respon untuk kandidat jawaban bertujuan untuk menentukan jawaban dari pertanyaan yang ditanyakan oleh *user*. Berikut adalah proses bagaimana sistem menentukan kandidat jawaban:

1. Penulis menentukan respon atau jawaban dari masing-masing dokumen latih atau dokumen pertanyaan.
2. Mencari bobot nilai seluruh dokumen latih yang telah dibandingkan dengan dokumen baru (pertanyaan).
3. Urutkan bobot nilai dokumen tersebut dari nilai yang paling besar ke nilai yang paling kecil.
4. Ambil satu dokumen latih dengan nilai kedekatan yang paling besar.
5. Tampilkan respon atau jawaban sesuai dengan dokumen pertanyaan.

5.1.7 Manualisasi pengelompokan dokumen menggunakan *query matching*

Perhitungan manual berfungsi untuk memberikan gambaran umum perancangan sistem yang dibangun. Contoh manualisasi setiap proses dari *query*

matching untuk menentukan jawaban dari pertanyaan yang ditanyakan user adalah sebagai berikut:

1. Input dokumen latih oleh peneliti, sedangkan dokumen baru adalah pertanyaan yang ditanyakan oleh user.
   a. Dokumen Baru
      d_baru : Ruangnya pak Eko ada dimana
   b. Dokumen Latih
      d_satu : Dimanakah ruang Pak Adarul
      d_dua : Dimanakah ruang Pak Eko
      d_tiga : Dimanakah ruang Pak Gembong
      d_empat : Dimanakah ruang Pak Wijaya
      d_lima : Dimanakah ruang akademik

2. *Tokenizing*
   a. Dokumen Baru
      d_baru : Ruangnya, pak, Eko, ada, dimana
   b. Dokumen Latih
      d_satu : Dimanakah, ruang, Pak, Adarul
      d_dua : Dimanakah, ruang, Pak, Eko
      d_tiga : Dimanakah, ruang, Pak, Gembong
      d_empat : Dimanakah, ruang, Pak, Wijaya
      d_lima : Dimanakah, ruang, akademik

3. *Stopword Removal / Filtering*
   a. Dokumen Baru
      d_baru : Ruangnya, pak, Eko, dimana
   b. Dokumen Latih
      d_satu : Dimanakah, ruang, Pak, Adarul
      d_dua : Dimanakah, ruang, Pak, Eko
      d_tiga : Dimanakah, ruang, Pak, Gembong
      d_empat : Dimanakah, ruang, Pak, Wijaya
      d_lima : Dimanakah, ruang, akademik

4. *Stemming*
   a. Dokumen Baru
      d_baru : Ruang, pak, Eko, mana
   b. Dokumen Latih
      d_satu : mana, ruang, Pak, Adarul
      d_dua : mana, ruang, Pak, Eko
      d_tiga : mana, ruang, Pak, Gembong
      d_empat : mana, ruang, Pak, Wijaya
      d_lima : mana, ruang, akademik

5. Pembobotan Dokumen
   a. Perhitungan nilai *query* pada dokumen baru

d_baru = 1 / jumlah kata
= 1 / 4 = 0,25

b. Perhitungan nilai *query* pada dokumen latih
d_satu = 1 / jumlah kata
= 1 / 4 = 0,25
d_dua = 1 / 4 = 0,25
d_tiga = 1 / 4 = 0,25
d_empat = 1 / 4 = 0,25
d_lima = 1 / 3 = 0,33333333333

c. Perhitungan bobot nilai dokumen latih yang telah dibandingkan dengan dokumen baru.

Dalam dokumen d_baru mempunyai 3 kata yang sama dengan dokumen latih, yaitu kata "dimana, ruang, pak", sehingga perhitungan bobot kedekatan menjadi seperti berikut:

d_satu = (0,25 x 3) + (0,25 x 3) = 1,5
d_dua = (0,25 x 4) + (0,25 x 4) = 2,0
d_tiga = (0,25 x 3) + (0,25 x 3) = 1,5
d_empat = (0,25 x 3) + (0,25 x 3) = 1,5
d_lima = (0,25 x 2) + (0,33333 x 2) = 1,16666

5.2 Implementasi

Tahap Implementasi berisi tentang pelaksanaan atau tindakan yang dilakukan penulis dari rencana yang sudah disusun sesuai dengan perancangan pada bab sebelumnya.

5.2.1 Spesifikasi sistem

Hasil dari analisis kebutuhan dan perancangan perangkat lunak yang telah dijelaskan pada tahap analisis kebutuhan sistem menjadi dasar untuk dilakukan implementasi menjadi sebuah sistem penjawab pertanyaan otomatis menggunakan ucapan yang dapat berfungsi sesuai dengan kebutuhan. Spesifikasi sistem diimplementasikan pada spesifikasi perangkat keras dan perangkat lunak.

5.2.1.1 Spesifikasi perangkat keras

Untuk mengimplementasikan pengenalan ucapan yang digunakan sebagai sistem penjawab pertanyaan menggunakan *library* PocketSphinx diperlukan perangkat keras dengan spesifikasi yang dijelaskan pada Tabel 5.1 berikut.

Tabel 5.1 Spesifikasi Perangkat Keras

Nama komponen	Spesifikasi
System Model PC	Toshiba Satellite
Processor	Intel(R) Core(TM) i3 CPU M370 @ 2.40 GHz x2

Memory	2.0 GiB
Disk	33.5 GB
OS Type	32-bit
Kelengkapan	- *Headphone Jack* - 1 *Microphone Input*

Tabel 5.1 merupakan penjabaran mengenai spesifikasi selruh perangkat keras yang digunakan untuk merancang sistem ini. Penggunaan *microphone* berfungsi dalam melakukan perekaman data ucapan untuk pertanyaan sebagai alat penunjang saat proses *training* ucapan, serta untuk *testing* program penjawab pertanyaan, sehingga akan meminimalkan *noise* yang masuk. Adapun spesifikasi dari *microphone* ditunjukkan pada Tabel 5.2.

Tabel 5.2 Spesifikasi *microphone*

Nama	Philips SBC MD150
Batas Frekuensi	*Headphone*: 8 Hz – 20KHz *Microphone*: 8 Hz – 20KHz

Tabel 5.2 menjabarkan spesifikasi mengenai microphone yang dipakai untuk proses perekaman data ucapan dan proses testing ucapan maupun keseluruhan sistem

5.2.1.2 Spesifikasi perangkat lunak

Spesifikasi dari perangkat lunak yang digunakan dalam mengimplementasikan pengenalan ucapan untuk digunakan sebagai sistem penjawab pertanyaan menggunakan *library* PocketSphinx adalah sebagai berikut:

Tabel 5.3 Spesifikasi Perangkat Lunak

Nama komponen	**Spesifikasi**
Sistem Operasi	Ubuntu 14.04 LTS
Bahasa pemrograman	C, Python, Perl
Library	PocketSphinx versi 0.7, SphinxBase versi 0.7, SphinxTrain versi 1.07
Aplikasi Pendukung	GNOME Terminal 3.4.1.1, Gedit 3.4.1, Pycharm, Python 2.7.3, Perl 5, AN4, Audacity Voice Recorder

Tabel 5.3 menjabarkan mengenai seluruh perangkat lunak yang peneliti gunakan untuk mendukung implementasi sistem penjawab pertanyaan menggunakan *speech recognition* ini

5.2.2 Batasan-batasan implementasi

Beberapa batasan dalam implementasi pada penelitian ini adalah sebagai berikut:

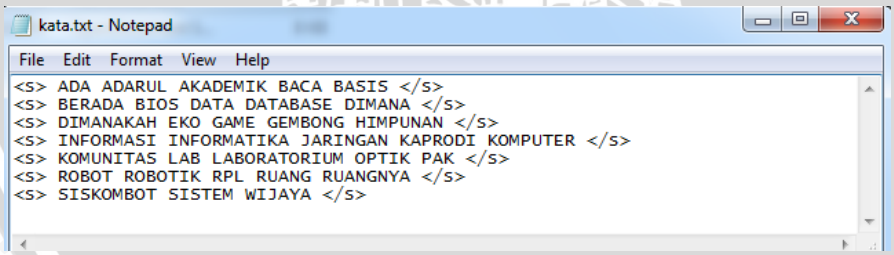
1. *Interface* sistem ini menggunakan terminal yang ada di Ubuntu 14.04 LTS.
2. Implementasi pengenalan ucapan *library* PocketSphinx menggunakan bahasa pemrograman C.
3. Implementasi program *query matching* menggunakan pemrograman python.

5.2.3 Implementasi *training* pada PocketSphinx

Training ucapan pada PocketSphinx dilakukan untuk mengenali ciri ucapan dari beberapa warna ucapan manusia yang dijadikan sebagai pertanyaan. Ada beberapa *software* yang harus di install terlebih dahulu, antara lain:

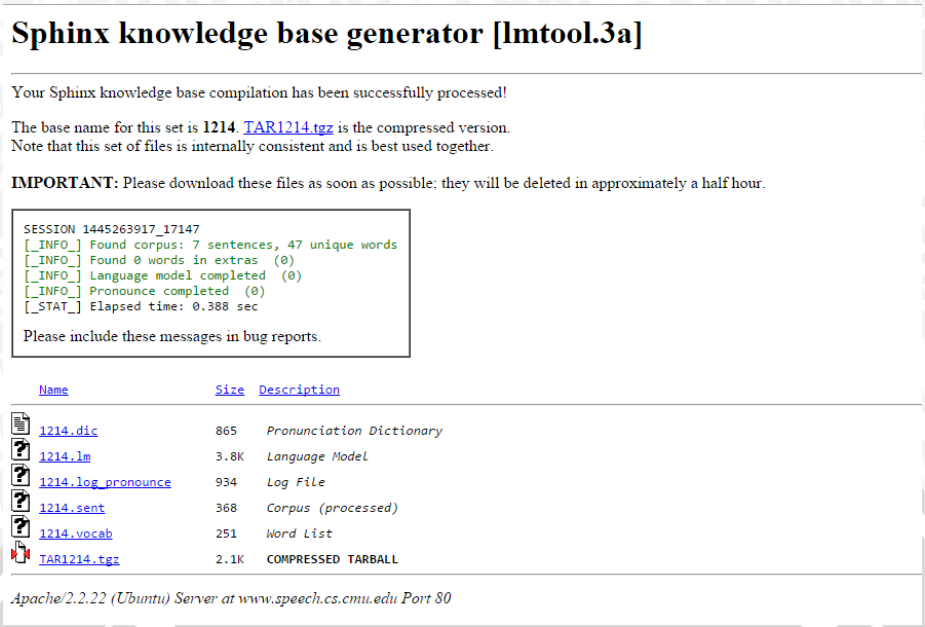
1. SphinxBase
2. SphinxTrain
3. PocketSphinx

Fungi dari melakukan *training* ucapan ialah agar hasil keluaran dari *training* dapat digunakan pada saat *testing program* pada PocketSphinx. Dalam implementasi ini, langkah pertama yang dilakukan adalah penulisan daftar kata yang digunakan sebagai kalimat pertanyaan untuk sistem penjawab otomatis. Daftar kata ditulis pada *text editor* dengan ekstensi .txt seperti dalam Gambar 5.11 berikut.



Gambar 5.11 Daftar kata pertanyaan

Selanjutnya daftar kata tersebut diunggah di *knowledge base tool* pada halaman URL: http://www.speech.cs.cmu.edu/tools/lmtool-new.html untuk menghasilkan model bahasa dan kamus pengucapan. Berikut adalah hasil dari *knowledge base tool* yang ditampilkan pada Gambar 5.12.



Sphinx knowledge base generator [lmtool.3a]

Your Sphinx knowledge base compilation has been successfully processed!

The base name for this set is **1214**. TAR1214.tgz is the compressed version.
Note that this set of files is internally consistent and is best used together.

IMPORTANT: Please download these files as soon as possible; they will be deleted in approximately a half hour.

```
SESSION 1445263917_17147
[_INFO_] Found corpus: 7 sentences, 47 unique words
[_INFO_] Found 0 words in extras  (0)
[_INFO_] Language model completed  (0)
[_INFO_] Pronounce completed  (0)
[_STAT_] Elapsed time: 0.388 sec
```

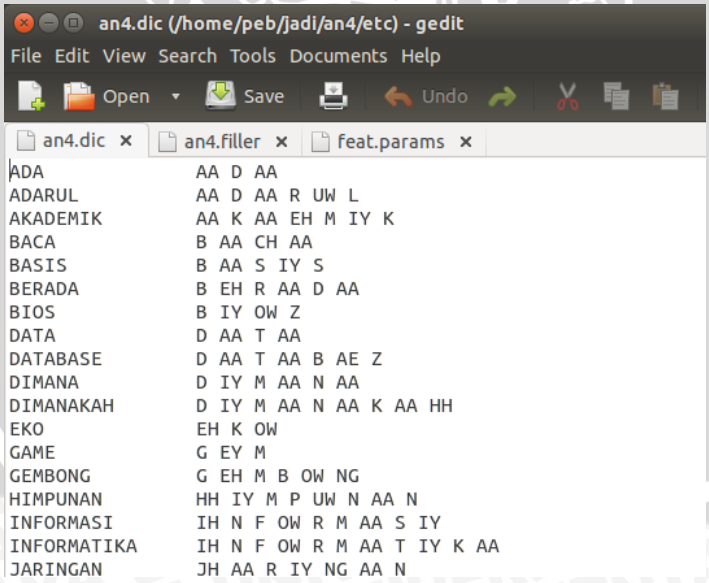
Please include these messages in bug reports.

Name	Size	Description
1214.dic	865	*Pronunciation Dictionary*
1214.lm	3.8K	*Language Model*
1214.log_pronounce	934	*Log File*
1214.sent	368	*Corpus (processed)*
1214.vocab	251	*Word List*
TAR1214.tgz	2.1K	COMPRESSED TARBALL

Apache/2.2.22 (Ubuntu) Server at www.speech.cs.cmu.edu Port 80

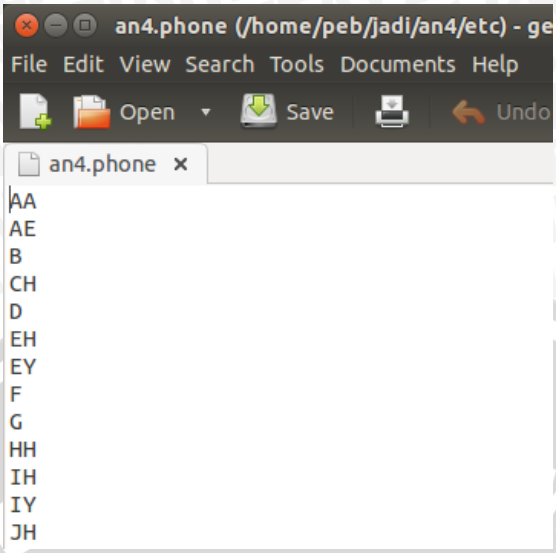
Gambar 5.12 *Knowledge base tool*

Gambar 5.12 merupakan penjabaran setelah proses untuk menghasilkan model bahasa melalui *tool* yang tersedia pada halaman web CMUSphinx. Dari proses yang dilakukan di *knowledge base tool* dihasilkan *file dictionary* yang berisi pengucapan fonem dari masing-masing kata, namun fonem masih disesuaikan dengan pengucapan bahasa Inggris. Untuk digunakan dalam pengenalan ucapan bahasa Indonesia, fonem harus disesuaikan dengan pengucapan bahasa Indonesia seperti pada Gambar 5.13 berikut.



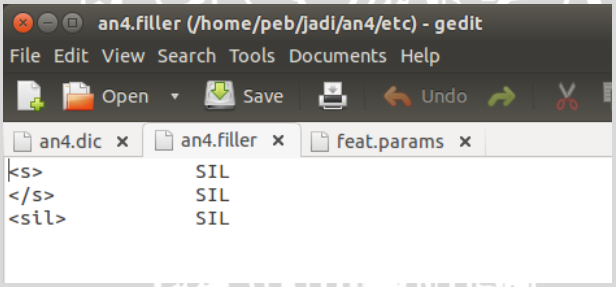
Gambar 5.13 Isi kamus pengucapan

Berdasarkan *file dictionary*, implementasi selanjutnya adalah membentuk *file phone* yang merupakan daftar fonem yang telah disesuaikan dengan pengucapan bahasa Indonesia. Daftar fonem dapat dilihat pada Gambar 5.14.



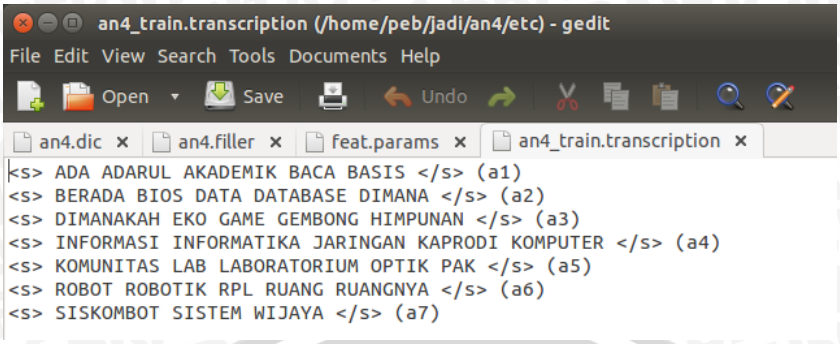
Gambar 5.14 Isi file phone

Setelah membuat *file dictionary* dan *phone*, dibutuhkan juga *file fillers* yang digunakan pada saat keadaan sepi. *File fillers* berguna untuk memberi jeda antar kalimat pertanyaan, sehingga diartikan sebagai *silent* atau sepi. Adapun format penulisan secara *default* pada *file fillers* diisi dengan *"SIL"* yang dapat dilihat pada Gambar 5.15.



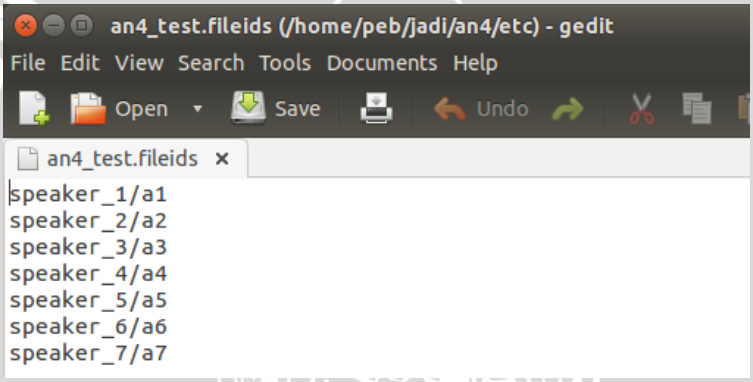
Gambar 5.15 Isi file *filler*

Selanjutnya adalah merancang *file transcription* yang berisi daftar kata yang digunakan sebagai pertanyaan. Isi *file transcription* disesuaikan dengan nama *file* ucapan yang berformat *.wav* seperti yang ditampilkan pada Gambar 5.16.



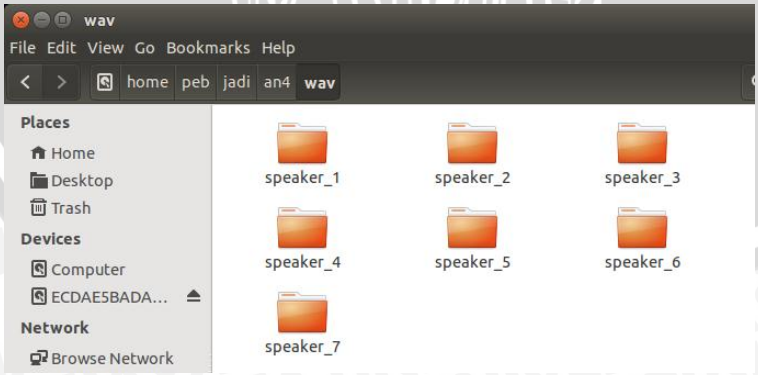
Gambar 5.16 Isi file *transcription*

Dalam daftar ucapan terdiri dari suara 4 responden, hal tersebut dimaksudkan agar pengenal ucapan semakin akurat untuk mengenali ucapan manusia yang berbeda-beda. Peletakan *folder file* ucapan juga harus ditentukan, maka dibuatlah *file fileids* yang merupakan *path*/penunjuk *file* ucapan yang akan dilakukan *training*. Dalam *file fields* harus dicantumkan nama *folder* dan nama *file* ucapan yang berformat *wav*.



Gambar 5.17 Isi *file fileids*

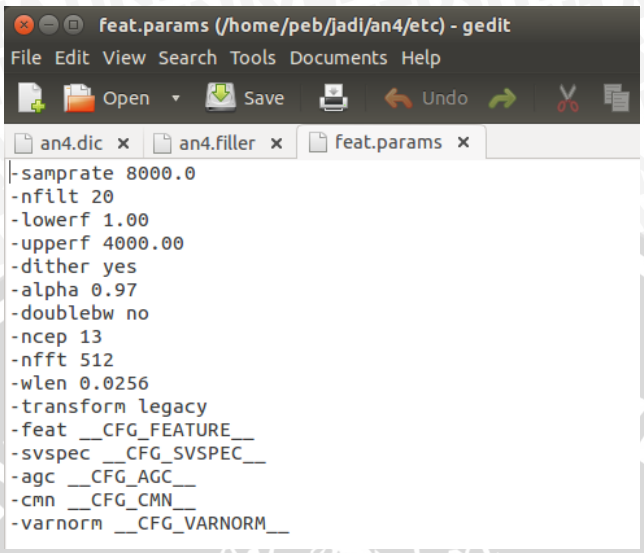
Gambar 5.17 menjelaskan isi dari *file fields* yang berisi *path* berkas dari file rekaman ucapan.



Gambar 5.18 Folder yang rekaman ucapan

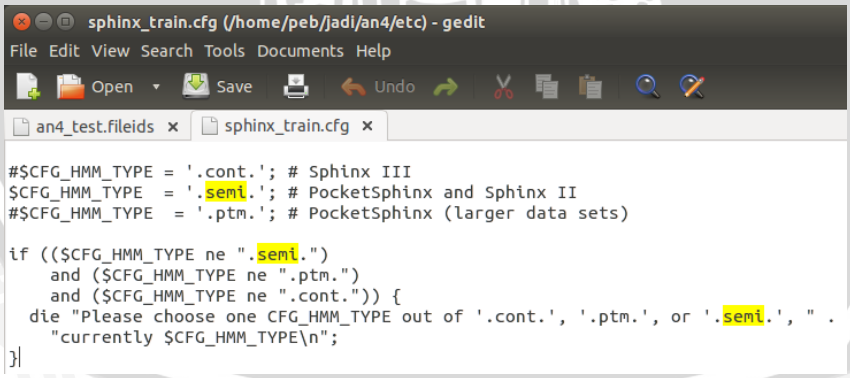
Gambar 5.18 merupakan penjabaran dari *folder* yang berisi daftar ucapan. Setelah semua *file* yang dilakukan untuk *training* terpenuhi, akan dilakukan pengaturan spesifikasi dari ucapan untuk mengenali ucapan manusia

dalam bahasa Indonesia, pengaturan tersebut dirancang dalam *file feat.params*. Pengaturan yang peneliti gunakan disesuaikan berdasarkan referensi dari *website* CMUSphinx yang dijabarkan pada Gambar 5.19.



Gambar 5.19 Isi file *feat.params*

Berdasarkan tutorial yang diberikan oleh website CMUSphinx, untuk pembuatan *training acoustic model*, terdapat 3 tipe model jenis perekaman, yakni *continuous* (*cont*), *semi-continuous* (*semi*), dan *phonetically tied mixtures* (*ptm*). Model *ptm* digunakan jika jumlah data perekaman yang cukup banyak. Model *semi* digunakan jika jumlah data perekaman sedikit dan membutuhkan kecepatan akses saat proses *testing*. Model *cont* digunakan jika jumlah data perekaman berada di antara model *ptm* dan *semi*. Pada implementasi saat ini digunakan model *semi-continuous* model seperti dalam Gambar 5.20.



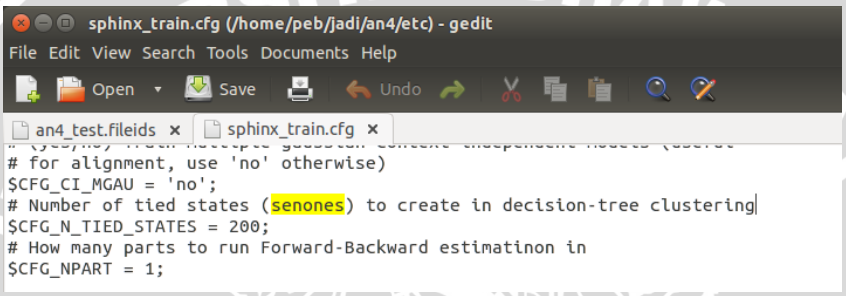
Gambar 5.20 Pengaturan Model *semi*

Gambar 5.20 merupakan pengaturan pada *sphinx_train.cfg* berdasarkan jumlah data kata yang digunakan. Parameter pengaturan yang digunakan terdiri dari *senone* dan *density*. *Senone* dan *density* adalah pengaturan untuk jumlah kata atau *vocabulary* yang di-*training* pada PocketSphinx. Gambar 5.21 adalah penjelasan mengenai pengaturan nilai *senone* dan *density* yang ditetapkan oleh CMUSphinx.

Vocabulary	Hours in db	Senones	Densities	Example
20	5	200	8	Tidigits Digits Recognition
100	20	2000	8	RM1 Command and Control
5000	30	4000	16	WSJ1 5k Small Dictation
20000	80	4000	32	WSJ1 20k Big Dictation
60000	200	6000	16	HUB4 Broadcast News
60000	2000	12000	64	Fisher Rich Telephone Transcription

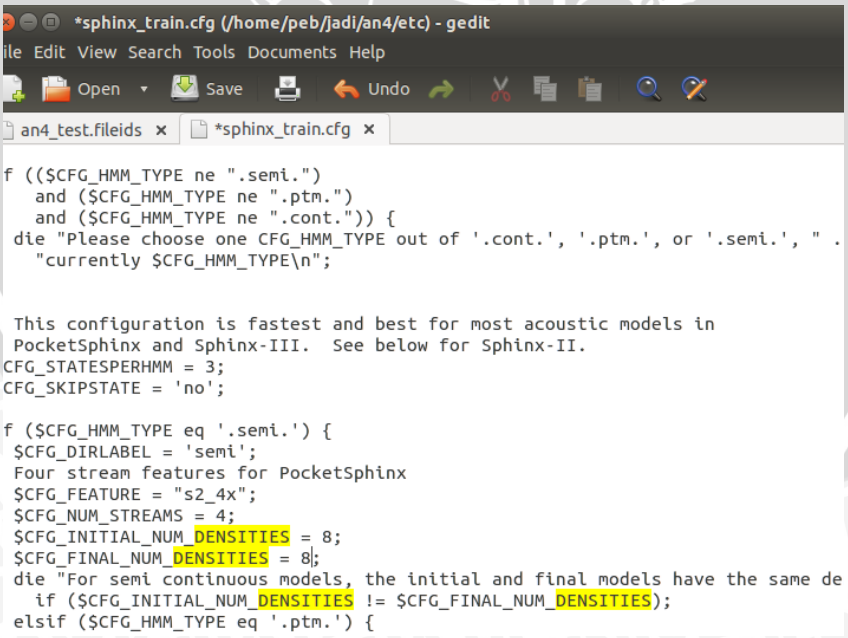
Gambar 5.21 *Density dan Senone*

Peneliti menggunakan pengaturan *senone 200* dan *density 8*, pengaturan tersebut berdasarkan jumlah kata untuk *training* tidak begitu banyak, sehingga menggunakan pengaturan yang minimal. Berikut adalah penjabaran mengenai nilai *senone* dan *density* yang ditampilkan pada Gambar 5.22 dan Gambar 5.23.



Gambar 5.22 Jumlah *Senones*

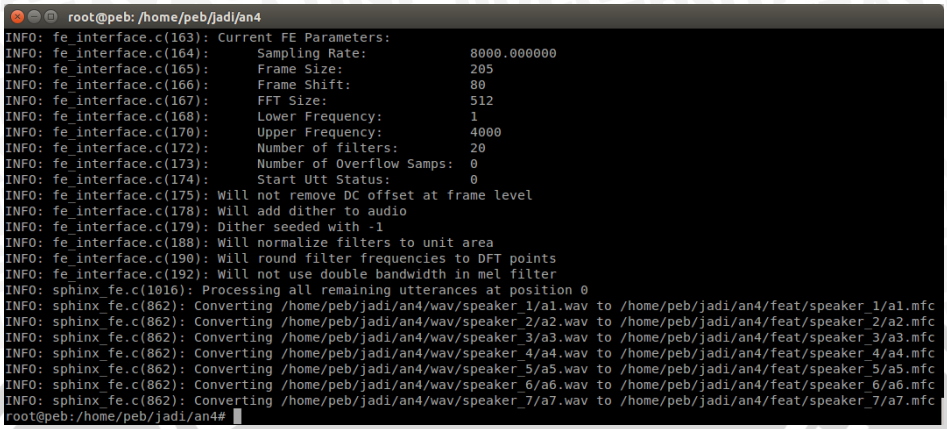
Gambar 5.22 merupakan pengaturan *senone* yang peneliti gunakan sesuai dengan jumlah kata yang dipakai.



Gambar 5.23 Besar *Density*

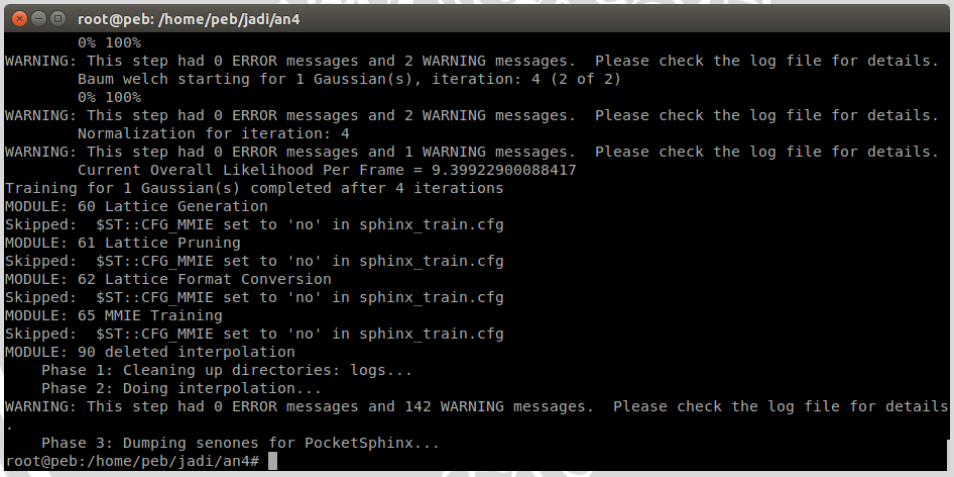
Gambar 5.23 merupakan pengaturan *density* yang peneliti gunakan sesuai dengan jumlah kata yang dipakai. Setelah melakukan konfigurasi terhadap nilai-

nilai parameter jumlah kata, maka akan dilakukan konversi file ucapan dengan format *wav* menjadi *mfc* dalam bentuk *binary* dengan perintah "*./scripts_pl/make_feats.pl –ctl etc/an4_train .fileids*". Hasil konversi *file* dapat dilihat pada Gambar 5.24.



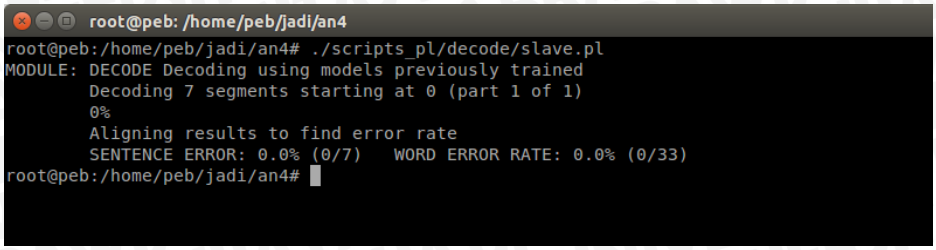
Gambar 5.24 Proses Pembentukan *File mfc*

Tahap terakhir adalah melakukan proses *delete interpolation* dengan perintah "*./scripts.pl/RunAll.pl*", *delete interpolation* merupakan teknik untuk *N-gram smoothing*. Berikut proses *training* yang ditampilkan pada Gambar 4.25.



Gambar 5.25 Proses *Delete Interpolation*

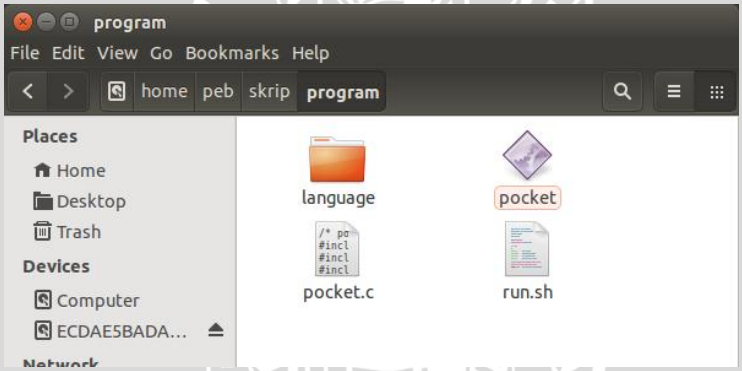
Gambar 5.25 merupakan penjabaran mengenai proses Hasil dari *training* ucapan menggunakan *library* PocketSphinx dapat dilihat melalui modul *decode* untuk melihat estimasi kualitas berdasarkan *Word Error Rate* dan *Sentence Error*. Dari 33 kata yang peneliti pakai untuk digunakan pada sistem penjawab pertanyaan mempunyai *Word Error Rate* sebesar 0% dan *Sentence Error* sebesar 0%. Kesimpulan dari proses *training* ucapan melalui modul *decode* mempunyai akurasi sebesar 100%. Berikut hasil *training* yang ditampilkan pada Gambar 5.26.



Gambar 5.26 Hasil *Decode*

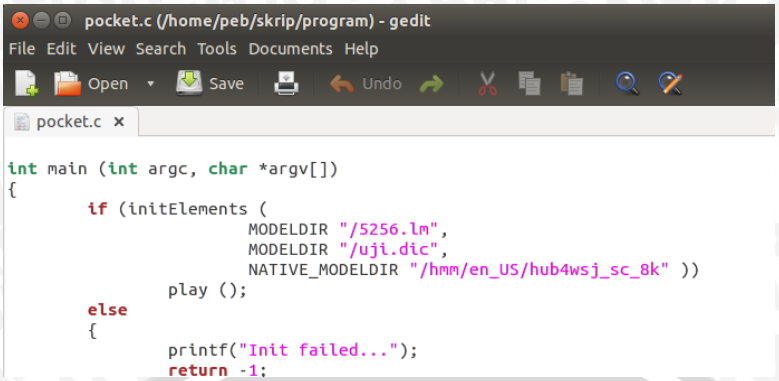
5.2.4. Implementasi program *testing* pada PocketSphinx

Implementasi ini merupakan hasil dari implementasi *training* ucapan dengan *library* SphinxTrain. Dalam melakukan *testing* pengenalan ucapan dibutuhkan *dictionary file*, *language model* dan *acoustic model* yang dihasilkan dari proses *training. Dictionary file* dan *language model* ditempatkan pada *folder* yang sama dengan *program testing*, sedangkan *acoustic model* didapat di *folder "model_parameters"* yang berada pada *file AN4*, lalu pilihlah *folder* "an4.cd_semi 200" yang isi filenya akan digunakan sebagai isi dari *folder* "hmm" pada *library* PocketSphinx. Berikut adalah hasil pengambilan datanya yang dapat dilihat pada Gambar 5.27.



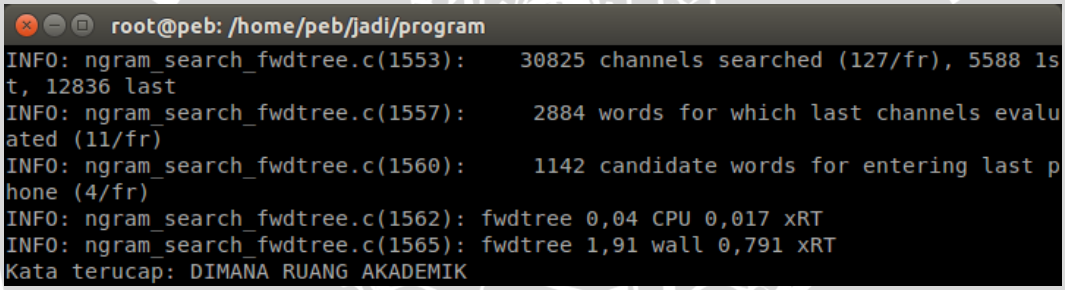
Gambar 5.27 Kebutuhan File untuk *Testing Program*

Dari beberapa kebutuhan *file* untuk *testing program* tersebut, maka dapat dipanggil seperti *listing code* pada Gambar 5.28 dibawah ini.



Gambar 5.28 *Listing Code*

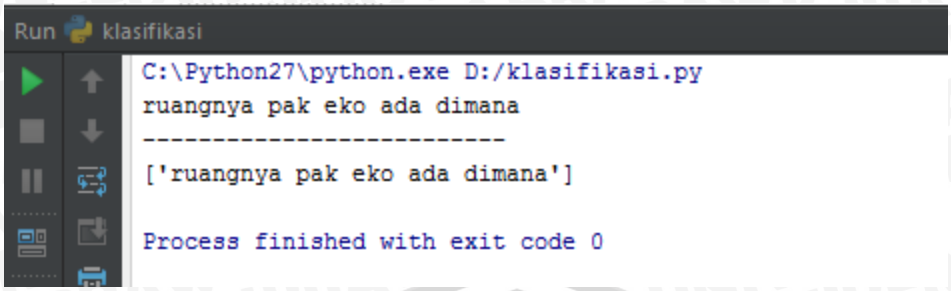
Setelah semua *file* dan *listing code* terpenuhi, selanjutnya adalah melakukan percobaan untuk pengenalan ucapan yang digunakan sebagai pertanyaan. *Testing* dilakukan dengan menjalankan *file run.sh* dan mengucapkan kata dengan menggunakan *microphone*, sinyal ucapan kemudian diekstraksi fiturnya dan *decode* secara otomatis dengan *library* PocketSphinx. Hasil dari pengenalan ucapan berupa teks yang dapat dilihat pada Gambar 5.29.



Gambar 5.29 Hasil *Testing Program*

5.2.5. Implementasi program *preprocessing*

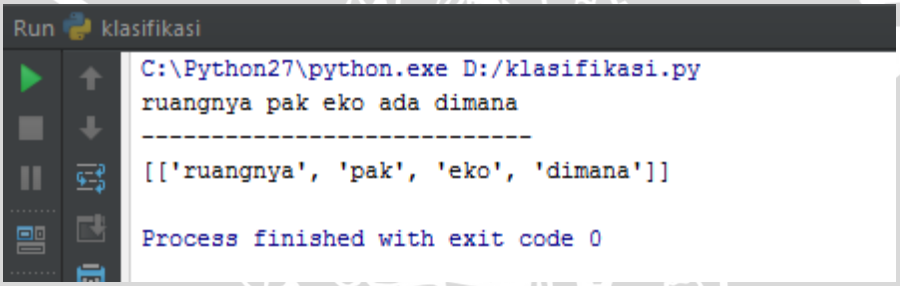
Implementasi *preprocessing* untuk mengolah teks kalimat pertanyaan yang diucapkan melalui *testing program* pada *library* PocketSphinx. Teks yang dihasilkan oleh *testing program* tidak akan ditampilkan, tetapi akan disimpan dalam *file* berformat *txt*. *File txt* hanya menyimpan satu kalimat pertanyaan, jika ada kalimat pertanyaan baru, maka kalimat pertanyaan yang lama terhapus secara otomatis. Teks pertanyaan dalam *file txt* kemudian dilakukan *preprocessing*, tahap pertama akan dilakukan pemisahan kalimat menjadi kata atau *tokenizing*. Pemisahan dilakukan dengan memotong kata berdasarkan karakter spasi dalam kalimat pertanyaan. Berikut adalah hasil dari *tokenizing* yang ditampilkan pada Gambar 5.30.



Gambar 5.30 Hasil *Tokenizing* Teks

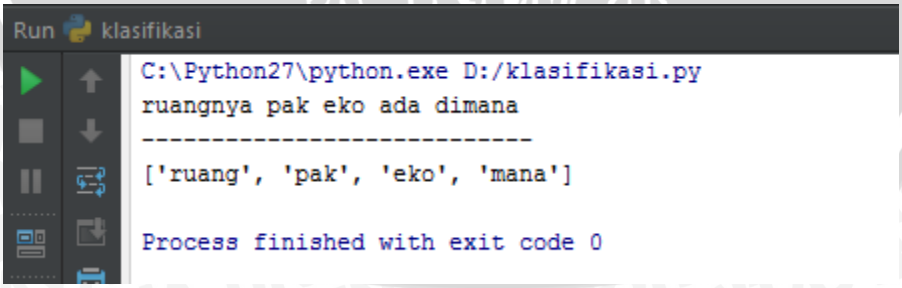
Teks pertama pada Gambar 5.30 merupakan teks pertanyaan dari *file txt* yang dihasilkan *testing program* PocketSphinx, sedangkan teks kedua adalah hasil dari *stemming*.

Setelah menjadi potongan kata, akan dihapus kata bantuan yang tidak dibutuhkan dalam pembobotan dokumen atau biasa disebut *stopword removal* atau *filtering*, kata bantuan yang dihapus seperti yang, ini, itu, adalah, ada, dan sebagainya. Berikut adalah hasil dari *stopword removal* yang ditampilkan pada Gambar 5.31.



Gambar 5.31 Hasil *Stopword Removal*

Kata “ada” dianggap sebagai kata bantuan yang tidak dibutuhkan untuk pembobotan dokumen, maka kata tersebut akan dihapus. Setelah dihasilkan kata-kata penting maka dilanjutkan dengan tahapan pemotongan imbuhan atau *stemming* dan menghasilkan kata dasar seperti yang ditampilkan pada Gambar 5.32.



Gambar 5.32 Hasil *Stemming*

Dalam program *stemming,* diperlukan penghapusan imbuhan awal dan imbuhan akhir. Keterangan dalam Gambar 5.32 menampilkan kata “ruangnya” dan dihapus imbuhan “–nya” sehingga menjadi kata dasar “ruang”, juga dalam kata “dimana” menghapus imbuhan “di-“ sehingga menjadi kata dasar “mana”.

Akhirnya kalimat pertanyaan tersebut menjadi kata-kata dasar yang siap diolah untuk mencari bobot nilai masing-masing dokumen.

5.2.6. Implementasi program q*uery matching*

Implementasi program *query matching* digunakan untuk menentukan dimana dokumen baru yang berupa pertanyaan dikelompokkan. Dokumen baru tersebut dicocokkan berdasarkan *query* yang cocok terhadap dokumen latih. Dokumen baru adalah kalimat pertanyaan dari user yang sudah diolah pada tahap *preprocessing*, sedangkan dokumen latih adalah daftar kata-kata yang digunakan sebagai pertanyaan untuk menanyakan ruang, seperti yang dijabarkan pada Gambar 5.33 berikut.

```
def add_document():
    table.addDocument("d_satu", ["mana", "ruang", "pak", "adarul"])
    table.addDocument("d_dua", ["mana", "ruang", "pak", "eko"])
    table.addDocument("d_tiga", ["mana", "ruang", "pak", "gembong"])
    table.addDocument("d_empat", ["mana", "ruang", "pak", "wijaya"])
    table.addDocument("d_lima", ["mana", "ruang", "akademik"])
```

Gambar 5.33 Dokumen Latih

Dari dokumen baru yang didapatkan dari tahap *preprocessing* dicari bobot nilai dokumennya terhadap dokumen latih dengan menggunakan *keyword extraction*. Perhitungan tersebut berdasarkan query yang cocok antar masing-masing dokumen, sehingga didapatkan bobot nilai masing-masing dokumen latih seperti pada Gambar 5.34 berikut.

```
Run tfidf
C:\Python27\python.exe D:/tfidf.py
[['d_satu', 1.5], ['d_dua', 2.0], ['d_tiga', 1.5], ['d_empat', 1.5], ['d_lima', 1.1666666666666665]]

Process finished with exit code 0
```

Gambar 5.34 Hasil Bobot Nilai Dokumen Latih

Setelah bobot nilai masing-masing dokumen latih diketahui, program akan melakukan sorting atau pengurutan bobot nilai dari terbesar ke terkecil. Proses sorting bertujuan untuk mengambil satu dokumen yang muncul pada urutan pertama yang akan dijadikan kelompok untuk dokumen baru. Hasil dari proses sorting bobot nilai dokumen latih dapat dilihat pada Gambar 5.35 berikut.

```
Run tfidf
C:\Python27\python.exe D:/tfidf.py
[[2.0, 'd_dua'], [1.5, 'd_tiga'], [1.5, 'd_satu'], [1.5, 'd_empat'], [1.1666666666666665, 'd_lima']]

Process finished with exit code 0
```

Gambar 5.35 Hasil *Sorting* Bobot Nilai

Berdasarkan hasil dari sorting bobot nilai dokumen, maka dokumen latih d_dua adalah dokumen dimana dokumen baru di kelompokkan.

5.2.7. Implementasi program respon kandidat jawaban

Setelah dokumen baru ditemukan kelompok dokumennya, yaitu dokumen d_dua, tahap selanjutnya yaitu implementasi respon untuk jawaban yang bertujuan untuk menampilkan jawaban dari pertanyaan yang ditanyakan oleh user. Gambar 5.36 berikut merupakan daftar jawaban yang digunakan sebagai respon dari pertanyaan yang ditanyakan oleh user.

```
def jawaban():
    table.jawaban("1", ["Gedung A Lantai 1 Ruang A1.6.3"])
    table.jawaban("2", ["Gedung C Ruang C1.12"])
    table.jawaban("3", ["Gedung C Ruang C1.19"])
    table.jawaban("4", ["Gedung C Ruang C1.12"])
    table.jawaban("5", ["Gedung G"])
```

Gambar 5.36 Daftar Jawaban

Sebelumnya, peneliti sudah menentukan kandidat jawaban untuk masing-masing dokumen latih seperti yang dijelaskan pada Gambar 5.37 berikut.

```
hasil = test[0]
if hasil[1] == '1':
    print table.tampil(0)
elif hasil[1]== '2':
    print table.tampil(1)
elif hasil[1]== '3':
    print table.tampil(2)
elif hasil[1]== '4':
    print table.tampil(3)
elif hasil[1]== '5':
    print table.tampil(4)
```

Gambar 5.37 Program Penentuan Kandidat jawaban

Dokumen baru yang dikelompokkan dengan dokumen “d_dua” akan menampikan isi dari dokumen “2” sebagai respon atau jawaban dari pertanyaan yang ditanyakan. Maka hasil dari pertanyaan “Ruangnya Pak Eko ada dimana” akan menghasilkan jawaban berupa “Gedung C Ruang C1.12”. Berikut hasil keluaran dari program *query match* yang ditampilkan pada Gambar 5.38.

```
root@peb: /media/peb/ECDAE5BADAE580E6/KULIAH/SK
pertanyaan : ['ruangnya pak eko ada dimana\n']
--------------------
jawabannya =
['Gedung C Ruang C1.12']
None
```

Gambar 5.38 Hasil Keluaran Program *Query Match*

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini dilakukan proses pengujian dan analisis dari metode *query matching* untuk sistem penjawab pertanyaan dengan *speech recognition* pada *library* Pocketsphinx. Pengujian dalam penelitian ini terdiri dari tiga proses uji coba sistem. Pengujian pertama adalah pengujian *speech to text* dari pengenalan suara menggunakan *library* PocketSphinx. Pengujian yang kedua adalah pengujian terhadap respon yang dihasilkan dengan menggunakan perhitungan sensifisitas dan spesifisitas dari metode *query matching*. Dan pengujian yang ketiga adalah pengujian keseluruhan sistem. Tujuan dilakukannya tiga pengujian tersebut adalah untuk mengetahui apakah semua sub-sistem dapat bekerja sesuai dengan yang diinginkan.

Sedangkan analisis dilakukan untuk mendapatkan kesimpulan dari hasil pengujian pada yang telah dilakukan.

6.1. Pengujian pengenalan ucapan

Pengujian pengenalan ucapan adalah pengujian *speech to text* yang dilakukan untuk mengetahui tingkat akurasi PocketSphinx dalam merubah ucapan menjadi teks. Pengujian dilakukan dengan 3 responden yang mengucapkan kata yang sudah dilatih sebelumnya. Masing-masing responden mengucapkan kata yang sama sebanyak 10 kali, sehingga didapatkan 30 kali pengujian terhadap masing-masing kata yang digunakan sebagai penyusun kalimat tanya yang digunakan. Tujuan dilakukan pengujian ini untuk menentukan kinerja pengenalan ucapan terhadap suara yang berbeda. Berikut hasil pengujian pengenalan ucapan tiap kata yang dijabarkan oleh Tabel 6.1 berikut.

Tabel 6.1 Pengujian pengenalan ucapan

No	Kata	Akurasi per kata
1	ada	83,33%
2	adarul	6,66%
3	akademik	80%
4	baca	63,33%
5	basis	90%
6	berada	43,33%
7	bios	100%
8	data	90%
9	database	33,3%
10	dimana	100%
11	dimanakah	96,66%
12	eko	100%
13	game	90%
14	gembong	100%
15	himpunan	100%
16	informasi	100%
17	informatika	100%

No	Kata	Akurasi per kata
18	Jaringan	80%
19	Kaprodi	100%
20	Computer	33,33%
21	Komunitas	100%
22	Lab	90%
23	laboratorium	86,66%
24	Optic	83,33%
25	Pak	63,33%
26	Robot	70%
27	Robotic	70%
28	Rpl	56,66%
29	Ruang	73,33%
30	Ruangnya	96,66%
31	Siskombot	0%
32	Sistem	90%
33	Wijaya	93,33

Dari pengujian pengenalan ucapan untuk masing-masing kata, menghasilkan rata-rata akurasi sebesar 77.67% dari seluruh responden. Dari responden 1, didapatkan tingkat akurasi sebesar 82,12%, responden 2 sebesar 75,15%, dan responden 3 sebesar 75, 75%. Seluruh hasil pengujian pengenalan ucapan yang dilakukan penulis terdapat dilampiran

6.2. Pengujian respon pertanyaan

Pengujian respon pertanyaan dilakukan dengan menggunakan tes sensitivitas dan spesifisitas untuk mengetahui ketepatan program *query matching* dalam mengelompokkan dokumen untuk menghasilkan jawaban yang benar. Pengujian dilakukan dengan memberikan pertanyaan berupa teks, karena pada pengujian ini hanya menguji metode *query matching* yang sudah dibangun. Pengujian dilakukan dengan menggunakan teknik sensitivitas dan spesifisitas. Sensitivitas adalah seberapa besar prosentase program menghasilkan jawaban salah jika diberi masukan berupa pertanyaan yang salah pula. Sedangkan spesifisitas adalah seberapa besar persentase program dalam menghasilkan jawaban yang benar jika diberi masukan berupa pertanyaan yang benar pula. Berikut adalah rumus untuk menghitung sensitifitas dan spesifisitas: (Wen Zhu, 2010)

$$Sensitivitas = \frac{Positive\ True}{\text{Positive True} + \text{Negative False}}\ x\ 100\%$$

$$Spesifisitas = \frac{Negative\ True}{\text{Positive False} + \text{Negative True}}\ x\ 100\%$$

POSITIVE TRUE Pertanyaan Benar - Jawaban Benar	**POSITIVE FALSE** Pertanyaan Benar - Jawaban Salah
NEGATIVE FALSE Pertanyaan Salah - Jawaban Benar	**NEGATIVE TRUE** Pertanyaan Salah - Jawaban Salah

Gambar 6.1 Penjabaran tentang rumus sensitivitas dan spesifisitas

Gambar 6.1 merupakan penjabaran mengenai parameter-parameter yang digunakan untuk menghitung nilai sensitivitas dan spesifisitas. Dalam Pengujian sensitivitas dan spesifisitas, diperlukan data pertanyaan yang benar dan pertanyaan yang salah. Pada pengujian pertama, dilakukukan dengan memberikan 20 kalimat pertanyaan yang benar. Dari 20 kalimat pertanyaan tersebut, masing-masing pertanyaan diuji dengan 3 variasi pertanyaan, jadi total ada 60 pertanyaan yang diujikan. Tujuan dari penggunaan variasi pertanyaan adalah untuk menentukan kinerja program jika diberikan pertanyaan yang berbeda tetapi dengan maksud yang sama. Sedangkan pengujian kedua yaitu memberikan 20 pertanyaan yang salah, dengan jumlah pertanyaan yang sama seperti sebelumnya.

Contoh:

Positif True

Pertanyaan: Dimana ruang Akademik?
Jawaban: Gedung G

Positif False

Pertanyaan: Dimana ruang Akademik?
Jawaban: Gedung C Ruang C 1.12

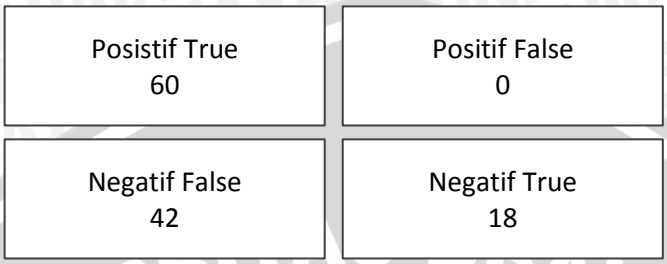
Negatif True

Pertanyaan: Dimana ruang Akademiknya Sistem Komputer?
Jawaban: Gedung C Ruang C 1.12

Negatif False

Pertanyaan: Dimana ruang Akademiknya Sistem Komputer?
Jawaban: Gedung G

Berdasarkan pengujian yang dilakukan, dari seluruh (60 kalimat) pertanyaan benar yang diujikan, tidak satupun menghasilkan respon yang salah. Sedangkan saat diuji dengan memberikan 60 pertanyaan salah, program pengelompokan dokumen dengan metode *query matching* menghasilkan jawaban yang salah sebanyak 18, sisanya menghasilkan jawaban yang benar seperti yang dijabarkan pada Gambar 6.2.



Gambar 6.2 Hasil pengujian Sensitivitas dan Spesifisitas

Dari hasil pengujian yang dilakukan, menghasilkan nilai seperti yang dijabarkan pada gambar 6.2 diatas. Sehingga dari nilai tersebut, kemudian dihitung sesuai dengan rumus yang sudah ditentukan seperti berikut:

$$Sensitivitas = \frac{60}{60+42}\ x\ 100\% = 58{,}82\%$$

$$Spesifisitas = \frac{18}{0+18}\ x\ 100\% = 100\%$$

6.3. Pengujian keseluruhan Sistem

Pengujian keseluruhan sistem dilakukan untuk mengetahui akurasi sistem untuk memberikan respon dari pertanyaan yang ditanyakan user melalui ucapan. Pengujian menggunakan 3 responden yang menanyakan 20 pertanyaan dengan banyaknya perulangan ucapan sebanyak 2 kali, jadi masing-masing responden mengucapkan pertanyaan sebanyak 40 kali. Masing-masing responden menggunakan kalimat pertanyaan dengan variasi yang berbeda-beda satu sama lain, namun tetap dengan maksud yang sama. Tujuannya yaitu untuk mengetahui akurasi sistem untuk memberikan respon atau jawaban yang benar dari pertanyaan berupa ucapan.

Dari pengujian yang telah dilakukan seluruh responden, didapatkan hasil pengujian yang dijabarkan pada Tabel 6.2.

Tabel 6.2 Hasil pengujian keseluruhan sistem

	Responden 1	Responden 2	Responden 1
Respon Benar	33	32	34
Respon Salah	7	8	6
Prosentase	82,5%	80%	85%
Rata-rata	82,5%		

6.4. Analisis dan evaluasi hasil program

Setelah melihat hasil pengujian pada pengenalan ucapan menggunakan *library* pocketshinx yang hanya mempunyai tingkat akurasi sebesar 77,67%, dapat disimpulkan bahwa terdapat beberapa faktor yang dapat mempengaruhi *library* PocketSphinx dalam menangkap sinyal suara. Hal tersebut disebabkan oleh beberapa faktor yaitu:

1. Karena *library* PocketSphinx tidak mendukung pengenalan ucapan dalam bahasa Indonesia, sehingga fonem pada kosakata harus mengikuti pola fonem bahasa inggris.
2. Perekaman suara tidak semua dilakukan di ruang rekaman khusus.
3. Terdapat beberapa kosakata yang pengucapannya hasmpir mirip berdasarkan fonemnya, sehingga sulit dikenali dengan baik oleh library.

Pengujian respon pertanyaan mempunyai akurasi sebesar 100%, artinya program *query matching* yang dibangun sangat cocok untuk pengelompokan teks dengan teks yang sedikit. Hal tersebut karena teknik *keyword extraction* menentukan kelompok berdasarkan kata yang dijadikan sebagai *query*, jadi program *query matching* hanya memilih kata yang penting untuk mewakili suatu pertanyaan. Meskipun PocketSphinx tidak dapat mengenali ucapan secara sempurna, program *query matching* tetap dapat menentukan kelompok dengan benar asalkan *query* yang ditentukan dapat dikenali oleh CMUSphinx.

Secara umum, meskipun pada pengujian pengenalan ucapan hanya mendapatkan akurasi sebesar 77,67%, tetapi pada pengujian respon pertanyaan saat diberi pertanyaan yang benar menghasilkan keakuratan respon sebesar 100%. Sehingga setelah dirata-rata, dari hasil pengujian terhadap keseluruhan sistem didapatkan tingkat akurasi sistem dalam mengenali maksud ucapan sebesar 82,5%. Meningkatnya akurasi pada pengujian keseluruhan sistem disebabkan karena sangat akuratnya metode *query matching* untuk mencari informasi atau respon dari pertanyaan.

Dengan hasil pengujian tersebut, penulis yakin bahwa Pocketsphinx yang sebelumnya hanya mampu mengenali ucapan, dapat dikembangkan lebih baik lagi untuk mengenali maksud ucapan, sehingga dapat melakukan interaksi dengan manusia ditopik yang lebih luas, tidak hanya seputar ruang di Fakultas Ilmu Komputer saja.

BAB 7 PENUTUP

Pada bab ini dipaparkan kesimpulan dan saran yang dapat diambil setelah melakukan pengujian terhadap sistem yang sudah dibangun.

7.1 Kesimpulan

Kesimpulan yang dapat diambil setelah melakukan pengujian terhadap penerapan metode *query matching* untuk sistem penjawab pertanyaan dengan *speech recognition* menggunakan *library* PocketSphinx antara lain:

1. Pengenalan ucapan pada sistem ini menggunakan *library* PocketSphinx, PocketSphinx merupakan salah satu *library* pengenalan ucapan dari CMUSphinx yang dapat diimplementasikan pada perangkat *mobile* dan sistem *embedded*.
2. Diperlukan pengubahan fonem kosakata disesuaikan dengan pengucapan orang Indonesia, sehingga pengenalan ucapan dapat mengenali ucapan dalam Bahasa Indonesia.
3. Untuk merancang sistem pengenalan ucapan yang dapat mengerti ucapan, dapat memanfaatkan metode *query macthing* yang bertujuan untuk mencari informasi berdasarkan kelompok dokumen, ketika kelompok dokumen sudah ditemukan, maka dihasilkan respon. Dapat dikatakan sistem pengenalan ucapan ini mampu berinteraksi dengan *user* jika dapat menghasilkan respon yang benar.
4. Sistem ini mampu memberikan jawaban yang benar saat diberikan pertanyaan berupa ucapan dalam bahasa Indonesia dengan persentase sebesar 82,5%. Kelemahan didominasi oleh pengenalan suara pada *library* PocketSphinx yang hanya dapat menghasilkan teks yang sesuai dengan ucapan sebesar 77,67%. Kelemahan tersebut disebabkan karena terbatasnya *library* PocketSphinx yang belum mendukung pengenalan ucapan dalam bahasa Indonesia, sehingga fonem pengucapan kata harus menyesuaikan pengucapan bahasa inggris.

7.2 Saran

Saran yang dapat diberikan unutk pengembangan penelitian penerapan metode query matching untuk sistem penjawab pertanyaan dengan speech recognition menggunakan library PocketSphinx antara lain:

1. Lebih banyak memberikan ciri suara manusia atau rekaman ucapan pada saat melakukan *training*, sehingga akan meningkatkan akurasi dalam mengenali berbagai suara manusia yang berbeda-beda.
2. Perekaman suara maupun saat testing suara lebih baik menggunakan *microphone* yang mempunyai kualitas bagus, hal itu akan

mempengaruhi pengenalan ucapan karena suara dari *microphone* akan terdengar lebih jelas.

3. Pada penelitian saat ini, masih diimplementasikan diperangkat laptop, untuk pengembangan penelitian lebih lanjut dapat diimplementasikan di perangkat mikrokomputer seperti Raspberry Pi.

DAFTAR PUSTAKA

Carnegie Mellon University, 1996. What is Speech recognition?. Tersedia di: <http:// www.speech.cs.cmu.edu/comp.speech/Section6/Q6.1.html> [diakses 4 Agustus 2015]

Carniege Mellon University, 2015. Tersedia di: http:// cmusphinx.sourceforge.net/wiki/tutorialPocketSphinx/> [diakses 4 Agustus 2015]

Deppy Rangga Maulana, 2014. Question Answering System Berbasis Clustering Pada Buku Pedoman Ptiik Dengan Menggunakan Algoritma Levenshtein Distance. Universitas Brawijaya, Malang, Indonesia.

Elasticsearch. Tersedia di: https://www.elastic.co/guide/en/elasticsearch/ guide/current/match-query.html [diakses 20 januari 2016]

Feng L. C., 2014. Understanding the CMUSphinx Based Recognition System. Department of Computer Science, National Chengchi University.

Guangzhi Zhang dkk, 2013. The Architecture of ProMe Instant Question Answering System. Beijing Normal University, Beijing, China.

HA. Ahmad dkk, 2010. Rancang Bangun Sistem Pengelolaan Dokumen-Dokumen Penting Menggunakan Text Mining. Jurusan Teknik Informatika, Politeknik Negeri Surabaya, Surabaya.

Jialun Lin , Xiaoling Li & Yuan Jiao, 2010. Text Categorization Research Based on Cluster Idea. Hainan Medical College.

Mokhammad Hilman Fatah, 2014. Implementasi *Library* PocketSphinx Untuk Pengenalan *Voice Command* Berbahasa Indonesia Secara *Offline.* Universitas Brawijaya, Malang, Indonesia.

Oracle, 2015. Data Mining Concept, Tersedian di: <http://docs.oracle.com/cd/ B28359_01/datamine.111/b28129/classify.htm> [diakses 30 Otober 2015]

Raymond J. Mooney, 2006. Machine Learning Text Categorization. University of Texas

Ryszard. T., 2010. Speech In Human System Interaction. AGH University of Science and technology, Department of Automatics, Krakow, Polandia.

Wen Zhu, 2010. Sensitivity, Specificity, Accuracy, Associated Confidence Interval and ROC Analisys with Practical SAS Implementation. Octagon Research Solution.

LAMPIRAN A PENGUJIAN SISTEM

A.1 Lampiran Pengujian pengenalan ucapan

No	Kata	Responden 1		Responden 2		Responden 3		Akurasi per kata
		Pengulangan	Dikenali	Pengulangan	Dikenali	Pengulangan	Dikenali	
1	ada	10	8	10	7	10	10	83,33%
2	adarul	10	0	10	2	10	0	6,66%
3	akademik	10	9	10	5	10	10	80%
4	baca	10	10	10	0	10	9	63,33%
5	basis	10	9	10	8	10	10	90%
6	berada	10	7	10	1	10	5	43,33%
7	bios	10	10	10	10	10	10	100%
8	data	10	9	10	8	10	10	90%
9	database	10	2	10	4	10	4	33,3%
10	dimana	10	10	10	10	10	10	100%
11	dimanakah	10	10	10	10	10	9	96,66%
12	eko	10	10	10	10	10	10	100%
13	game	10	10	10	8	10	9	90%
14	gembong	10	10	10	10	10	10	100%
15	himpunan	10	10	10	10	10	10	100%
16	informasi	10	10	10	10	10	10	100%
17	informatika	10	10	10	10	10	10	100%
18	jaringan	10	6	10	9	10	9	80%
19	kaprodi	10	10	10	10	10	10	100%

20	komputer	10	7	10	3	10	0	33,33%
21	komunitas	10	10	10	10	10	10	100%
22	lab	10	8	10	9	10	10	90%
23	laboratorium	10	10	10	6	10	10	86,66%
24	optik	10	10	10	10	10	5	83,33%
25	pak	10	8	10	7	10	4	63,33%
26	robot	10	10	10	7	10	4	70%
27	robotik	10	9	10	10	10	2	70%
28	rpl	10	8	10	5	10	4	56,66%
29	ruang	10	5	10	10	10	7	73,33%
30	ruangnya	10	9	10	10	10	10	96,66%
31	siskombot	10	0	10	0	10	0	0%
32	sistem	10	8	10	9	10	10	90%
33	wijaya	10	9	10	10	10	9	93,33
Akurasi tiap responden		82,12%		75,15%		75,75%		
Akurasi per kata		77,67%						
Rata-rata Akurasi tiap responden		77,67%						

A.2 Pengujian respon pertanyaan

A.2.1 Pengujian dengan pertanyaan benar

No	Pertanyaan	Jawaban	Hasil Respon
1	Dimana ruang Pak Adarul	Gedung A Lantai 1 Ruang A 1.6.3	√
	Ruangnya Pak Adarul ada dimana	Gedung A Lantai 1 Ruang A 1.6.3	√
	Dimanakah ruang Pak Adarul berada	Gedung A Lantai 1 Ruang A 1.6.3	√
2	Dimana ruang Pak Eko	Gedung C Ruang C 1.12	√
	Ruangnya Pak Eko ada dimana	Gedung C Ruang C 1.12	√
	Dimanakah ruang Pak Eko berada	Gedung C Ruang C 1.12	√
3	Dimana ruang Pak Gembong	Gedung C Ruang C 1.9	√
	Ruangnya Pak Gembong ada dimana	Gedung C Ruang C 1.9	√
	Dimanakah ruang Pak Gembong berada	Gedung C Ruang C 1.9	√
4	Dimana ruang Pak Wijaya	Gedung C Ruang C 1.12	√
	Ruang Pak Wijaya ada dimana	Gedung C Ruang C 1.12	√
	Dimanakah ruang Pak Wijaya berada	Gedung C Ruang C 1.12	√
5	Dimana ruang Akademik	Gedung G	√
	Ruang Akademik ada dimana	Gedung G	√
	Dimanakah ruang Akademik berada	Gedung G	√
6	Dimana Lab Siskombot	Gedung C Ruang C 1.13	√
	Lab Siskombot ada dimana	Gedung C Ruang C 1.13	√
	Dimanakah Lab Siskombot berada	Gedung C Ruang C 1.13	√
7	Dimana Lab Game	Gedung C Ruang C 1.7	√
	Lab Game ada dimana	Gedung C Ruang C 1.7	√

	Dimanakah Lab Game berada	Gedung C Ruang C 1.7	√
8	Dimana Lab RPL	Gedung B Ruang B 1.4	√
	Lab RPL ada dimana	Gedung B Ruang B 1.4	√
	Dimanakah Lab RPL berada	Gedung B Ruang B 1.4	√
9	Dimana Lab Jaringan Komputer	Gedung B Ruang B 1.9	√
	Lab Jaringan Komputer ada dimana	Gedung B Ruang B 1.9	√
	Dimanakah Lab Jaringan Komputer berada	Gedung B Ruang B 1.9	√
10	Dimana Lab Basis Data	Gedung C Ruang C 1.5	√
	Lab Basis Data ada dimana	Gedung C Ruang C 1.5	√
	Dimanakah Lab Basis Data berada	Gedung C Ruang C 1.5	√
11	Dimana ruang himpunan Informatika	Gedung D Ruang D 1.1	√
	Ruang himpunan Informatika ada dimana	Gedung D Ruang D 1.1	√
	Dimanakah ruang himpunan Informatika berada	Gedung D Ruang D 1.1	√
12	Dimana ruang himpunan Sistem Komputer	Gedung D Ruang D 1.6	√
	Ruang himpunan Sistem Komputer ada dimana	Gedung D Ruang D 1.6	√
	Dimanakah ruang himpunan Sistem Komputer berada	Gedung D Ruang D 1.6	√
13	Dimana ruang himpunan Sistem Informasi	Gedung D Ruang D 1.6	√
	Ruang himpunan Sistem Informasi ada dimana	Gedung D Ruang D 1.6	√
	Dimanakah ruang himpunan Sistem Informasi berada	Gedung D Ruang D 1.6	√
14	Dimana ruang Kaprodi Informatika	Gedung A Lantai 1 Ruang A 1.6.1	√
	Ruang Kaprodi Informatika ada dimana	Gedung A Lantai 1 Ruang A 1.6.1	√
	Dimanakah ruang Kaprodi Informatika berada	Gedung A Lantai 1 Ruang A 1.6.1	√
15	Dimana ruang Kaprodi Sistem Komputer	Gedung A Lantai 1 Ruang A 1.6.3	√
	Ruang Kaprodi Sistem Komputer ada dimana	Gedung A Lantai 1 Ruang A 1.6.3	√
	Dimanakah ruang Kaprodi Sistem Komputer berada	Gedung A Lantai 1 Ruang A 1.6.3	√

16	Dimana ruang Kaprodi Sistem Informasi	Gedung A Lantai 1 Ruang A 1.6.2	√
	Ruang Kaprodi Sistem Informasi ada dimana	Gedung A Lantai 1 Ruang A 1.6.2	√
	Dimanakah ruang Kaprodi Sistem Informasi berada	Gedung A Lantai 1 Ruang A 1.6.2	√
17	Dimana ruang Optik	Gedung D Ruang D 1.8	√
	Ruang Optik ada dimana	Gedung D Ruang D 1.8	√
	Dimanakah ruang Optik berada	Gedung D Ruang D 1.8	√
18	Dimana ruang Komunitas Robot	Gedung D Ruang D 1.10	√
	Ruang Komunitas Robot ada dimana	Gedung D Ruang D 1.10	√
	Dimanakah ruang Komunitas Robot berada	Gedung D Ruang D 1.10	√
19	Dimana ruang Bios	Gedung D Ruang D 1.3	√
	Ruang Bios ada dimana	Gedung D Ruang D 1.3	√
	Dimanakah ruang Bios berada	Gedung D Ruang D 1.3	√
20	Dimana ruang Baca	Gedung A Lantai 2 Ruang A 2.20	√
	Ruang Baca ada dimana	Gedung A Lantai 2 Ruang A 2.20	√
	Dimanakah ruang Baca berada	Gedung A Lantai 2 Ruang A 2.20	√

A.2.2 Pengujian dengan pertanyaan salah

No	Pertanyaan	Jawaban	Hasil Respon
1	Ruang himpunan Pak Adarul ada dimana	Gedung A Lantai 1 Ruang A 1.6.3	√
	Lab Jaringannya Pak Adarul dimana	Gedung A Lantai 1 Ruang A 1.6.3	√
	Dimana himpunan mahasiswa RPL Pak Adarul	Gedung A Lantai 1 Ruang A 1.6.3	√
2	Ruang Baca Pak Eko dimana	Gedung C Ruang C 1.12	√
	Dimana ruang Komunitas Robot Pak Eko	Gedung C Ruang C 1.12	√

	Dimana ruang Sistem Komputer Pak Eko	Gedung C Ruang C 1.12	√
3	Dimana Optik Pak Gembong	Gedung C Ruang C 1.9	√
	Dimana Lab Baca Pak Gembong	Gedung C Ruang C 1.9	√
	Ruang Baca Pak Gembong dimana	Gedung C Ruang C 1.9	√
4	Dimana mahasiswa Informatika Pak Wijaya	Gedung C Ruang C 1.12	√
	Dimana Kaprodi Sistem Informasi Pak Wijaya	Gedung A Lantai 1 Ruang A 1.6.2	X
	Dimana ruang Akademiknya Pak Wijaya	Gedung C Ruang C 1.12	√
5	Dimana ruang Akademiknya Sistem Komputer	Gedung G	√
	Ruang Akademik Kaprodi Informatika dimana	Gedung A Lantai 1 Ruang A 1.6.1	X
	Ruang Akademik Lab Siskombot dimana	Gedung G	√
6	Lab Siskombot mahasiswa Sistem Informasi dimana	Gedung D Ruang D 1.6	X
	Lab Siskombot Komunitas Robot dimana	Gedung C Ruang C 1.13	√
	Dimanakah Lab Siskombot Kaprodi Informatika	Gedung A Lantai 1 Ruang A 1.6.1	X
7	Lab Game Komunitas Robot ada dimana	Gedung C Ruang C 1.7	√
	Dimana Lab Game mahasiswa Sistem Komputer	Gedung D Ruang D 1.6	X
	Dimanakah Lab Game mahasiswa Informatika	Gedung C Ruang C 1.7	√
8	Lab RPL mahasiswa Informatika dimana	Gedung B Ruang B 1.4	√
	Lab RPL Jaringan Komputer ada dimana	Gedung B Ruang B 1.4	√
	DimanaLab RPL himpunan Informatika	Gedung B Ruang B 1.4	√
9	Lab jaringan Komunitas Robot ada dimana	Gedung D Ruang D 1.10	X
	Dimana Lab Jaringan mahasiswa Informatika	Gedung D Ruang D 1.1	X
	Lab Jaringan Komputer Sistem dimana	Gedung B Ruang B 1.9	√
10	Lab Basis Data Sistem Informasi ada dimana	Gedung C Ruang C 1.5	√
	Lab Database Kaprodi Informatika ada dimana	Gedung A Lantai 1 Ruang A 1.6.2	X
	Lab basis data komunitas robot dimana	Gedung C Ruang C 1.5	√

11	Komunitas Jaringan himpunan Informatika dimana	Gedung D Ruang D 1.1	√
	Himpunan Kaprodi Informatika ada dimana	Gedung A Lantai 1 Ruang A 1.6.1	X
	himpunan sistem informatika ada dimana	Gedung D Ruang D 1.10	X
12	Dimana himpunan Komputer Lab Jaringan	Gedung D Ruang D 1.10	X
	Himpunan Komunitas Komputer dimana	Gedung D Ruang D 1.6	√
	lab himpunan sistem komputer jaringan dimana	Gedung B Ruang B 1.9	X
13	Himpunan Sistem Informasi baca dimana	Gedung D Ruang D 1.6	√
	Dimana himpunan Sistem Informasi Lab Data	Gedung D Ruang D 1.6	√
	dimana himpunan sistem informasi kaprodi	Gedung A Lantai 1 Ruang A 1.6.2	X
14	kaprodi Informatika baca dimana	Gedung A Lantai 1 Ruang A 1.6.1	√
	Dimana Kaprodi Informatika Akademik	Gedung A Lantai 1 Ruang A 1.6.1	√
	kaprodi informatika pak gembong	Gedung C Ruang C 1.9	X
15	Kaprodi Sistem Komputer Pak Adharul dimana	Gedung A Lantai 1 Ruang A 1.6.3	√
	Dimana ruang himpunan Kaprodi Sistem Komputer	Gedung A Lantai 1 Ruang A 1.6.3	√
	sistem komputer kaprodinya dimana	Gedung A Lantai 1 Ruang A 1.6.3	√
16	Dimana Lab himpunan Kaprodi Sistem Informasi	Gedung A Lantai 1 Ruang A 1.6.2	√
	Kaprodi Sistem Informasi ruang himpunannya dimana	Gedung A Lantai 1 Ruang A 1.6.2	√
	dimanakah kaprodi sistem informasi pak gembong	Gedung A Lantai 1 Ruang A 1.6.2	√
17	Dimanakah ruangnya Optik Pak Gembong	Gedung C Ruang C 1.9	X
	Ruangnya Optik himpunan Informatika dimana	Gedung D Ruang D 1.8	√
	optik ruang himpunannya pak eko	Gedung C Ruang C 1.12	X
18	Dimana Lab Komunitas Robot mahasiswa Informatika	Gedung D Ruang D 1.10	√
	Komunitasnya himpunan robotika dimana	Gedung D Ruang D 1.10	√
	komunitas robot mahasiswa komputer dimana	Gedung D Ruang D 1.10	√
19	Bios Kaprodinya dimana	Gedung D Ruang D 1.3	√

	Bios ruang bacanya dimana	Gedung A Lantai 2 Ruang A 2.20	X
	dimana ruang bios pak gembong	Gedung C Ruang C 1.9	X
20	Ruang Baca himpunan mahasiswa dimana	Gedung A Lantai 2 Ruang A 2.20	√
	Dimana ruang baca Lab Siskombot	Gedung A Lantai 2 Ruang A 2.20	√
	ruang bacanya komunitas robot dimana?	Gedung A Lantai 2 Ruang A 2.20	√

A.3 Pengujian keseluruhan sistem

A.3.1 Responden A

No	Pertanyaan yang Ditanyakan	Hasil Pengenalan Suara	Jawaban	
1	pak adarul ruangnya dimana	Data ada robot ruangnya dimana	Gedung D Ruang D 1.10	X
		Ada ruangnya dimana	Gedung G	X
2	pak eko ruangnya dimana	Pak eko ruangnya dimana	Gedung C Ruang C 1.12	√
		Eko ruangnya dimana	Gedung C Ruang C 1.12	√
3	pak gembong ruangnya dimana	Gembong ruangnya dimana	Gedung C Ruang C 1.9	√
		Gembong ruangnya dimana	Gedung C Ruang C 1.9	√
4	pak wijaya ruangnya dimana	Wijaya ruangnya dimana	Gedung C Ruang C 1.12	√
		Wijaya ruangnya dimana	Gedung C Ruang C 1.12	√
5	dimana ruang akademik	Dimana ruang akademik	Gedung G	√
		Dimana ruang akademik	Gedung G	√
6	dimana lab siskombot berada	Dimana robot berada	Gedung D Ruang D 1.10	X
		Dimana sistem lab	Gedung B Ruang B 1.4	X

7	dimana lab game berada	Dimana game	Gedung C Ruang C 1.7	√
		Dimana game	Gedung C Ruang C 1.7	√
8	dimana lab rpl berada	Dimana rpl game	Gedung B Ruang B 1.4	√
		Dimana rpl	Gedung B Ruang B 1.4	√
9	dimana lab jaringan berada	Dimana jaringan	Gedung B Ruang B 1.9	√
		Dimana jaringan	Gedung B Ruang B 1.9	√
10	dimana lab basis data berada	Dimana lab sistem	Gedung B Ruang B 1.4	X
		Dimana lab basis bios	Gedung C Ruang C 1.5	√
11	ruang himpunan informatika dimana	ruang himpunan informatika dimana	Gedung D Ruang D 1.1	√
		ruang himpunan informatika dimana	Gedung D Ruang D 1.1	√
12	ruang himpunan sistem komputer dimana	Ruang himpunan sistem robot dimana	Gedung D Ruang D 1.6	√
		Ruang himpunan sstem komputer dimana	Gedung D Ruang D 1.6	√
13	ruang himpunan sistem informasi dimana	ruang himpunan sistem informasi dimana	Gedung D Ruang D 1.6	√
		ruang himpunan sistem informasi dimana	Gedung D Ruang D 1.6	√
14	kaprodi informatika ruangnya dimana	Kaprodi informatika ruang eko	Gedung A Lantai 1 Ruang A 1.6.1	√
		Kaprodi informatika ruang eko dimana	Gedung A Lantai 1 Ruang A 1.6.1	√
15	kaprodi sistem komputer ruangnya dimana	kaprodi sistem komputer ruangnya dimana	Gedung A Lantai 1 Ruang A 1.6.3	√
		kaprodi sistem komputer ruangnya dimana	Gedung A Lantai 1 Ruang A 1.6.3	√
16	kaprodi sistem informasi ruangnya dimana	kaprodi sistem informasi ruangnya dimana	Gedung A Lantai 1 Ruang A 1.6.2	√
		kaprodi sistem informasi ruangnya dimana	Gedung A Lantai 1 Ruang A 1.6.2	√
17	ruang optik ada dimana	Ruang optik dimana	Gedung D Ruang D 1.8	√
		Ruang optik dimana	Gedung D Ruang D 1.8	√

18	ruang komunitas robot ada dimana	ruang komunitas robot dimana	Gedung D Ruang D 1.10	√
		ruang komunitas robot dimana	Gedung D Ruang D 1.10	√
19	dimanakah ruang bios berada	dimanakah ruang bios berada	Gedung D Ruang D 1.3	√
		dimanakah ruang bios	Gedung D Ruang D 1.3	√
20	dimanakah ruang baca berada	Dimanakah ruang lab	Gedung G	X
		Dimanakah ruang gembong	Gedung C Ruang C 1.9	X

A.3.2 Responden B

No	Pertanyaan yang Ditanyakan	Hasil Pengenalan Suara	Jawaban	
1	ruangnya pak adarul ada dimana	ruangnya ada dimana	Gedung B Ruang B 1.9	X
		ruangnya dimana	Gedung D Ruang D 1.3	X
2	dimanakah ruangnya pak eko	dimana ruangnya eko	Gedung C Ruang C 1.12	√
		game ruangnya eko	Gedung C Ruang C 1.12	√
3	ruang pak gembong ada dimana	ruang gembong dimana	Gedung C Ruang C 1.9	√
		ruang gembong dimana	Gedung C Ruang C 1.9	√
4	ruangnya pak wijaya dimana	ruangnya wijaya ada dimana	Gedung C Ruang C 1.12	√
		ruangnya wijaya dimana	Gedung C Ruang C 1.12	√
5	dimana ruang akademik	dimana ruang akademik	Gedung G	√
		dimana ruang akademik	Gedung G	√
6	lab siskombot ada dimana	robot dimana	Gedung D Ruang D 1.10	X
		optik lab dimana	Gedung D Ruang D 1.8	X
7	dimana laboratorium game	dimana laboratorium game	Gedung C Ruang C 1.7	√

		dimana laboratorium game	Gedung C Ruang C 1.7	√
8	dimanakah laboratorium RPL	dimanakah laboratorium	Gedung B Ruang B 1.4	√
		dimanakah laboratorium rpl	Gedung B Ruang B 1.4	√
9	dimana lab jaringan komputer	dimana jaringan komputer	Gedung B Ruang B 1.9	√
		dimana game gembong lab	Gedung C Ruang C 1.7	X
10	dimanakah lab basis data	dimana lab bios data	Gedung D Ruang D 1.3	X
		dimana basis bios	Gedung D Ruang D 1.3	X
11	dimana himpunan informatika	dimana himpunan informatika	Gedung D Ruang D 1.1	√
		dimana himpunan informatika	Gedung D Ruang D 1.1	√
12	himpunan sistem komputer ada dimana	himpunan sistem komputer dimana	Gedung D Ruang D 1.6	√
		himpunan sistem komputer dimana	Gedung D Ruang D 1.6	√
13	dimana himpunan sistem informasi	dimana himpunan sistem informasi	Gedung D Ruang D 1.6	√
		dimana himpunan sistem informasi	Gedung D Ruang D 1.6	√
14	dimana ruang kaprodi informatika	dimana ruang kaprodi informatika	Gedung A Lantai 1 Ruang A 1.6.1	√
		dimana ruang kaprodi informatika	Gedung A Lantai 1 Ruang A 1.6.1	√
15	ruang kaprodi sistem komputer dimana	ruang kaprodi sistem komputer dimana	Gedung A Lantai 1 Ruang A 1.6.3	√
		ruang kaprodi sistem komputer dimana	Gedung A Lantai 1 Ruang A 1.6.3	√
16	kaprodi sistem informasi ruangnya dimana	kaprodi sistem informasi ruangnya dimana	Gedung A Lantai 1 Ruang A 1.6.2	√
		kaprodi sistem informasi ruangnya dimana	Gedung A Lantai 1 Ruang A 1.6.2	√
17	optik ruangnya dimana	optik ruangnya dimana	Gedung D Ruang D 1.8	√
		optik ruangnya dimana	Gedung D Ruang D 1.8	√
18	dimana ruang komunitas robotik	dimana ruang komunitas robotik	Gedung D Ruang D 1.10	√

		game ruang komunitas robotik	Gedung D Ruang D 1.10	√
19	dimana ruang bios	dimana ruang bios	Gedung D Ruang D 1.3	√
		dimana ruang bios	Gedung D Ruang D 1.3	√
20	dimana ruang baca	dimana ruang baca	Gedung A Lantai 2 Ruang A 2.20	√
		dimana ruang	Gedung B Ruang B 1.9	X

A.3.2 Responden C

No	Pertanyaan yang Ditanyakan	Hasil Pengenalan Suara	Jawaban	
1	Dimana ruang pak adarul	dimana ruang pak	Gedung C Ruang C 1.12	X
		dimana ruang pak	Gedung C Ruang C 1.12	X
2	Ruangnya pak eko ada dimana	ruangnya pak pak dimana	Gedung C Ruang C 1.12	√
		ruangnya eko pak dimana	Gedung C Ruang C 1.12	√
3	dimana ruang pak gembong	dimana ruang gembong	Gedung C Ruang C 1.9	√
		dimana ruang gembong	Gedung C Ruang C 1.9	√
4	dimanakah ruangnya pak wijaya	dimanakah ruangnya wijaya	Gedung C Ruang C 1.12	√
		dimanakah ruangnya	Gedung C Ruang C 1.12	√
5	ruang akademik ada dimana	ruang ada pak optik dimana	Gedung D Ruang D 1.8	X
		ruang akademik pak dimana	Gedung G	√
6	dimana lab siskombot	dimana lab sistem lab	Gedung B Ruang B 1.4	X
		dimana lab sistem robot	Gedung B Ruang B 1.4	X
7	lab game ada dimana	lab game eko dimana	Gedung C Ruang C 1.7	√
		lab game pak dimana	Gedung C Ruang C 1.7	√

8	lab rpl ada dimana	lab rpl pak dimana	Gedung B Ruang B 1.4	√
		lab rpl pak dimana	Gedung B Ruang B 1.4	√
9	lab jaringan komputer ada dimana	lab jaringan robot pak dimana	Gedung B Ruang B 1.9	√
		lab jaringan komputer pak dimana	Gedung B Ruang B 1.9	√
10	lab basis data ada dimana	lab basis data pak dimana	Gedung C Ruang C 1.5	√
		lab basis dimana	Gedung C Ruang C 1.5	√
11	ruang himpunan informatika ada dimana	ruang himpunan informatika dimana	Gedung D Ruang D 1.1	√
		ruang himpunan informatika dimana	Gedung D Ruang D 1.1	√
12	dimanakah ruang himpunan sistem komputer	dimanakah ruang himpunan sistem robot	Gedung D Ruang D 1.6	√
		dimanakah ruang himpunan sistem komputer	Gedung D Ruang D 1.6	√
13	ruang himpunan sistem informasi ada dimana	himpunan ruang sistem informasi eko dimana	Gedung D Ruang D 1.6	√
		himpunan ruang sistem informasi pak dimana	Gedung D Ruang D 1.6	√
14	ruang kaprodi informatika ada dimana	ruang kaprodi optik informatika baca dimana	Gedung A Lantai 1 Ruang A 1.6.1	√
		ruang kaprodi informatika dimana	Gedung A Lantai 1 Ruang A 1.6.1	√
15	dimanakah ruang kaprodi sistem komputer	dimanakah ruang kaprodi sistem komputer	Gedung A Lantai 1 Ruang A 1.6.3	√
		dimanakah ruang kaprodi sistem komputer	Gedung A Lantai 1 Ruang A 1.6.3	√
16	ruang kaprodi sistem informasi dimana	kaprodi sistem informasi dimana	Gedung A Lantai 1 Ruang A 1.6.2	√
		ruang eko game sistem informasi dimana	Gedung A Lantai 1 Ruang A 1.6.2	√
17	dimana ruang optik	dimana ruang optik	Gedung D Ruang D 1.8	√
		dimana ruang optik	Gedung D Ruang D 1.8	√
18	ruang komunitas robot ada dimana	ruang komunitas lab pak dimana	Gedung B Ruang B 1.4	X
		ruang komunitas robot dimana	Gedung D Ruang D 1.10	√

19	ruang bios ada dimana	ruang bios data dimana	Gedung D Ruang D 1.3	√
		ruang bios pak ruang	Gedung D Ruang D 1.3	√
20	ruang baca ada dimana	ruang baca pak dimana	Gedung A Lantai 2 Ruang A 2.20	√
		ruang baca pak dimana	Gedung A Lantai 2 Ruang A 2.20	√