



# **Pengolahan Bahasa Alami untuk Penerjemahan Keluhan Pasien pada Anamnesis dalam Bentuk Teks atau Suara**

Pradipta Yuwono

S. Kom.

Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer

Program Magister Teknik Informatika

Universitas Islam Indonesia

2018

## **Lembar Pengesahan Pembimbing**

### **Pengolahan Bahasa Alami untuk Penerjemahan Keluhan Pasien pada Anamnesis dalam Bentuk Teks atau Suara**

Pradipta Yuwono, S. Kom.

14917132

Yogyakarta, April 2018

Pembimbing I

Pembimbing II

Izzati Muhimmah, S.T., M.Sc., Ph.D.

dr. Isnatin Miladiyah, M. Kes.

## Lembar Pengesahan Penguji

### Pengolahan Bahasa Alami untuk Penerjemahan Keluhan Pasien pada Anamnesis dalam Bentuk Teks atau Suara

Pradipta Yuwono, S. Kom.

14917132

Yogyakarta, April 2018

Tim Penguji,

Izzati Muhimmah, S.T., M.Sc., Ph.D.

(Ketua)

---

dr. Isnatin Miladiyah, M. Kes.

(Anggota I)

---

Dr. Sri Kusumadewi, S.Si., MT.

(Anggota II)

---

## **Abstrak**

Anamnesis adalah wawancara dokter kepada pasiennya yang dilakukan untuk mendapatkan informasi medis dari pasien untuk membantu penegakan diagnostik. Salah satu isi dari anamnesis adalah keluhan pasien yang dapat disimpan dalam rekam medis elektronik. Data keluhan pasien yang ada pada rekam medis elektronik sering tertulis dalam format penulisan bebas yang kurang baku dan tidak terstruktur sehingga menyulitkan komputer untuk melakukan pemrosesan secara otomatis terhadap data tersebut. Penelitian yang dilakukan ini berusaha mengembangkan sebuah aplikasi yang dapat menerjemahkan keluhan pasien yang tertulis dalam format kurang baku dan tidak terstruktur tersebut ke bentuk yang lebih baku dan terstruktur menggunakan metode-metode pengolahan bahasa alami. Pendekatan pengolahan bahasa alami yang digunakan dalam penelitian ini adalah pendekatan pengolahan bahasa alami berbasis aturan dan statistik. Aplikasi yang dikembangkan selain mampu menerima masukan dalam bentuk teks juga mampu menerima masukan dalam bentuk suara baik dari ucapan secara langsung maupun dari rekaman suara. Aplikasi yang dikembangkan juga mampu mempelajari penerjemahan keluhan baru yang tidak terdapat pada *database default* dan fitur *speech recognition* nya mampu dilatih agar mampu mengenali suara *user* dengan lebih akurat di masa yang akan datang. Hasil pengujian terhadap aplikasi yang sudah dikembangkan menunjukkan bahwa aplikasi sudah mampu menerjemahkan keluhan pasien dengan tingkat akurasi dan kecepatan yang cukup memuaskan, karena dari lima orang dokter yang menguji aplikasi, tiga orang dokter setuju bahwa tingkat akurasi penerjemahan pada aplikasi tersebut sudah cukup baik dan dua orang dokter setuju bahwa kecepatan penerjemahan oleh aplikasi tersebut sudah cukup baik. Untuk fitur *speech recognition*, setelah diujicoba baik tingkat akurasi maupun kecepatannya masih belum cukup memuaskan karena dari lima orang dokter yang menguji aplikasi ini tidak ada satupun dokter yang setuju bahwa tingkat akurasi dan kecepatan fitur *speech recognition* pada aplikasi sudah cukup baik.

## **Kata kunci**

pengolahan bahasa alami, keluhan pasien, anamnesis, *speech recognition*

## **Abstract**

Anamnesis (history taking) is a physician's interview with his/her patient which is done to obtain medical informations from the patient to aid the diagnosis process. One of the contents of anamnesis are patient complaints that may be stored in an electronic medical record. Patient complaints data in electronic medical records are often written in a less standardized and unstructured format that makes them difficult for computers to process the data. The research undertaken attempts to develop an application that can translate complaints of patients written in a less standardized and unstructured format to a more standardized and structured format using natural language processing methods. The natural language processing approach used in this research is the natural language processing approach based on rules and statistics. The application is not only able to receive input in the text form but also able to receive input in the form of sound either from direct speech or from recorded sound. The application is also able to learn the translation of new complaints that do not exist in the default database and its speech recognition feature is also able to be trained to more accurately recognizes user's voices in the future. The test results of the application that has been developed show that the application has been able to translate complaints with quite satisfactory level of accuracy and speed, because from five doctors who had tested the application, three doctors agreed that the accuracy of translation was quite good and two doctors agreed that the speed of translation was good enough. For speech recognition feature, both the accuracy and the speed is not satisfactory because from five doctors who had tested this application none of the doctors agreed that the level of accuracy and speed of speech recognition feature in the application was good enough.

## **Keywords**

natural language processing, patient complaint, history taking, speech recognition

### **Pernyataan keaslian tulisan**

Dengan ini saya menyatakan bahwa tesis ini merupakan tulisan asli dari penulis, dan tidak berisi material yang telah diterbitkan sebelumnya atau tulisan dari penulis lain terkecuali referensi atas material tersebut telah disebutkan dalam tesis. Apabila ada kontribusi dari penulis lain dalam tesis ini, maka penulis lain tersebut secara eksplisit telah disebutkan dalam tesis ini.

Dengan ini saya juga menyatakan bahwa segala kontribusi dari pihak lain terhadap tesis ini, termasuk bantuan analisis statistik, desain survei, analisis data, prosedur teknis yang bersifat signifikan, dan segala bentuk aktivitas penelitian yang dipergunakan atau dilaporkan dalam tesis ini telah secara eksplisit disebutkan dalam tesis ini.

Segala bentuk hak cipta yang terdapat dalam material dokumen tesis ini berada dalam kepemilikan pemilik hak cipta masing-masing. Apabila dibutuhkan, penulis juga telah mendapatkan izin dari pemilik hak cipta untuk menggunakan ulang materialnya dalam tesis ini.

Yogyakarta, April 2018

Pradipta Yuwono, S. Kom.

### **Publikasi selama masa studi**

Yuwono, Pradipta. (2017). *Pengolahan Bahasa Alami untuk Penerjemahan Keluhan Pada Anamnesis dalam Bentuk Teks atau Suara*. Makalah dipresentasikan pada Seminar Nasional Teknologi Terapan UGM 2017.

### **Publikasi yang menjadi bagian dari tesis**

Seluruh publikasi yang dilakukan selama masa studi tersebut dikembangkan dan disempurnakan menjadi tesis ini.

### **Kontribusi yang diberikan oleh pihak lain dalam tesis ini**

Oceano Design berkontribusi dalam membantu pembuatan desain latar belakang pada jendela utama aplikasi.

Gambar *icon* untuk tombol mikrofon dan tombol *speaker* pada jendela utama aplikasi yang dikembangkan pada penelitian ini diperoleh dari situs [icons8.com](https://icons8.com).



## **Halaman Persembahan**

Tesis ini saya persembahkan kepada:

1. Bapak saya, Koesadiyuwono, dan Ibu saya, Istilah Handayani, yang selalu memberikan doa dan motivasi, serta membantu saya selama saya menempuh kuliah di Program Pascasarjana Teknik Informatika UII.
2. Dokter-dokter dan beberapa petugas kesehatan di Apotek Unisia Polifarma UII dan Rumah Sakit Sardjito Yogyakarta yang telah membantu saya dalam penyelesaian tesis ini.
3. Almamater saya, Universitas Islam Indonesia.

## **Kata Pengantar**

Assalamu'alaikum Wr.Wb. Alhamdulillahrabbi'l'amin, puji syukur senantiasa penulis panjatkan pada Allah SWT atas rahmat dan hidayah-Nya selama penyelesaian tesis ini, serta shalawat dan salam kepada Nabi Muhammad SAW beserta keluarga dan sahabat-sahabat-Nya. Tesis dengan judul "Pengolahan Bahasa Alami untuk Penerjemahan Keluhan Pasien pada Anamnesis dalam Bentuk Teks atau Suara" ini ditulis sebagai salah satu syarat untuk memperoleh gelar Magister Teknik Informatika pada Fakultas Teknologi Industri Universitas Islam Indonesia.

Penulis menyadari bahwa penyusunan Tesis ini tidak terlepas dari bimbingan, dorongan, dan bantuan baik material maupun non-material dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Ayahanda Koesadiyuwono dan Ibunda Istilah Handayani yang senantiasa memberikan dukungan semangat, perhatian, bantuan moril, bantuan materil, dan doa setiap saat.
2. Bapak Dr. R. Teduh Dirgahayu, ST, M.Sc., selaku Direktur Program Pascasarjana Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Ibu Izzati Muhimmah Ph.D., selaku Dosen Pembimbing I yang telah memberikan saran, kritik, serta bimbingan sehingga penulis dapat menyelesaikan penyusunan tesis dengan baik.
4. Ibu dr. Isnatin Miladiyah M.Kes., selaku Dosen Pembimbing II yang telah memberikan saran-saran, kritik serta bimbingan sehingga penulis dapat menyelesaikan penyusunan tesis dengan baik.
5. Dosen-dosen jurusan Teknik Informatika Program Pascasarjana UII yang telah memberikan ilmunya.
6. Dokter-dokter dan petugas kesehatan di Apotek Unisia Polifarma UII yang telah membagikan ilmunya serta mengujicoba dan memberikan penilaian pada aplikasi yang sudah dibuat.

Penulis menyadari bahwa penulisan Tesis ini masih tidak luput dari kekurangan, sehingga kritik dan saran yang membangun sangat dibutuhkan dan akan diterima dengan senang hati oleh penulis sebagai bahan evaluasi untuk peningkatan kualitas penulisan oleh penulis di masa mendatang. Akhir kata penulis berharap semoga penulisan tesis ini dapat bermanfaat bagi penulis maupun pembaca serta dapat menjadi amal ibadah yang diterima di sisi-Nya. Amin.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, April 2018

Penulis

## Daftar Isi

Lembar Pengesahan Pembimbing .....	i
Lembar Pengesahan Penguji.....	ii
Abstrak .....	iii
Abstract.....	iv
Pernyataan keaslian tulisan.....	v
Publikasi selama masa studi .....	vi
Kontribusi yang diberikan oleh pihak lain dalam tesis ini .....	vii
Kata Pengantar.....	ix
Daftar Isi .....	xi
Daftar Tabel.....	xiv
Daftar Gambar .....	xv
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah .....	3
1.4. Tujuan Penelitian .....	3
1.5. Manfaat Penelitian .....	3
1.6. Metodologi Penelitian.....	4
1.7. Sistematika Penulisan .....	6
<b>BAB II TINJAUAN PUSTAKA .....</b>	<b>7</b>
2.1. Penelitian Terdahulu .....	7
2.2. Anamnesis.....	9
2.3. Bahasa Alami.....	12
2.3.1. Tata Bahasa .....	14

2.3.2.	<i>Parsing</i> .....	18
2.4.	Pengolahan Bahasa Alami .....	20
2.4.1.	Tingkatan Analisis Pengolahan Bahasa Alami .....	21
2.4.2.	Pendekatan Pengolahan Bahasa Alami .....	23
2.4.3.	Aplikasi Pengolahan Bahasa Alami .....	24
2.4.4.	<i>Speech Synthesis</i> .....	26
2.4.5.	<i>Speech recognition</i> .....	28
BAB III METODOLOGI PENELITIAN .....		33
3.1.	Pengambilan Data .....	33
3.2.	Tahapan Penelitian.....	34
3.2.1.	Studi Literatur .....	35
3.2.2.	Perancangan Aplikasi.....	35
3.2.3.	Pengumpulan Data .....	35
3.2.4.	Perbaikan Data .....	36
3.2.5.	Pengembangan Aplikasi.....	36
3.2.6.	Pengujian dan Revisi.....	36
BAB IV PEMODELAN SISTEM.....		38
4.1.	Gambaran Model .....	38
4.2.	Algoritma Pembelajaran untuk Penerjemahan.....	44
4.3.	Algoritma Penerjemahan Menggunakan Pembelajaran Berdasarkan Statistik.....	46
4.4.	Algoritma Pembelajaran untuk <i>Speech recognition</i> .....	47
BAB V HASIL DAN PEMBAHASAN .....		50
5.1.	Tampilan Antarmuka .....	50
5.2.	Hasil Uji Coba .....	60
5.2.1	Hasil Uji Coba Pengukuran Tingkat Akurasi Fitur <i>Speech recognition</i> .....	60
5.2.2	Hasil Uji Coba Pengukuran Tingkat Akurasi Fitur Pengoreksian Kesalahan Ejaan .....	61

5.2.3 Hasil Uji Coba Pengukuran Kecepatan Proses Penerjemahan dan <i>Speech recognition</i> .....	61
5.2.4 Hasil Penilaian Kuesioner .....	66
BAB VI KESIMPULAN DAN SARAN.....	69
6.1. Kesimpulan .....	69
6.2. Saran .....	69
DAFTAR PUSTAKA.....	71
LAMPIRAN .....	77

## Daftar Tabel

Tabel 2.1. Perbandingan Aplikasi yang Dikembangkan pada Penelitian ini dengan Aplikasi pada Penelitian Sebelumnya.....	8
Tabel 5.1. Hasil Uji Coba Pengukuran Kecepatan Proses Penerjemahan.....	62
Tabel 5.2. Hasil Uji Coba Pengukuran Kecepatan Proses <i>Speech Recognition</i> .....	64
Tabel 5.3. Hasil Penilaian Kelima Dokter Terhadap Aspek-Aspek pada Program dari Hasil Kuesioner.....	66
Tabel 5.4. Saran-Saran dari Kelima Dokter yang Telah Menguji coba Aplikasi.....	68

## Daftar Gambar

Gambar 2.1. Contoh <i>Parse Tree</i> .	19
Gambar 2.2. Tugas utama pengolahan bahasa alami (Regina dan Michael, 2005).	20
Gambar 2.3. Urutan proses konversi dari teks ke ucapan (Arman, 2003).	27
Gambar 2.4. Hidden Markov Model tiga <i>state</i> .	32
Gambar 3.1. Bagan alur penelitian.	34
Gambar 3.2. Rancangan tampilan antarmuka utama aplikasi.	35
Gambar 3.3. Rancangan kuesioner.	37
Gambar 4.1. Model penerjemahan masukan keluhan pasien.	38
Gambar 4.2. Algoritma pembelajaran dengan data statistik.	44
Gambar 4.3. Algoritma penerjemahan menggunakan hasil pembelajaran berdasarkan statistik.	47
Gambar 4.4. Algoritma pembelajaran untuk <i>speech recognition</i> .	48
Gambar 5.1. Tampilan antarmuka jendela utama aplikasi.	50
Gambar 5.2. Tampilan antarmuka jendela utama aplikasi setelah <i>user</i> men- <i>submit</i> teks masukan.	52
Gambar 5.3. Tampilan antarmuka untuk memasukkan <i>password</i> .	53
Gambar 5.4. Tampilan antarmuka untuk mengubah <i>password</i> .	53
Gambar 5.5. Tampilan antarmuka untuk pembelajaran berdasarkan contoh.	54
Gambar 5.6. Tampilan antarmuka untuk pencarian data hasil pembelajaran berdasarkan contoh.	55
Gambar 5.7. Tampilan antarmuka untuk pembelajaran berdasarkan kata atau frasa kunci.	56
Gambar 5.8. Tampilan antarmuka untuk pencarian data hasil pembelajaran berdasarkan kata atau frasa kunci.	56
Gambar 5.9. Tampilan antarmuka untuk pembelajaran berdasarkan statistik.	57



Gambar 5.10. Tampilan antarmuka untuk <i>training</i> fitur <i>speech recognition</i> . .....	58
Gambar 5.11. Tampilan antarmuka bantuan diagnosis. ....	59
Gambar 5.12. Tampilan antarmuka untuk mengedit <i>database</i> diagnosis.....	60

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Anamnesis merupakan pemeriksaan tahap awal terhadap pasien yang dilakukan lewat suatu percakapan oleh dokter baik secara langsung atau melalui orang terdekat pasien untuk penegakan diagnostik. Tujuan anamnesis adalah untuk memperoleh informasi mengenai pasien agar dokter mampu menyimpulkan dugaan organ atau sistem organ yang terganggu beserta rumusan masalah kliniknya, serta untuk menjalin hubungan yang baik antara dokter dengan pasiennya (Markum, 2007). Ada dua jenis anamnesis, yaitu autoanamnesis dan alloanamnesis atau heteroanamnesis. Autoanamnesis adalah anamnesis yang dilakukan langsung terhadap pasien, sedangkan alloanamnesis atau heteroanamnesis adalah anamnesis yang dilakukan terhadap orang lain yang mengenal pasien. Alloanamnesis atau heteroanamnesis biasa dilakukan jika pasien dianggap kurang mampu melakukan wawancara seperti pasien yang pingsan, bisu, sakit jiwa, cacat mental, maupun pasien anak-anak. Terkadang autoanamnesis bisa dilakukan bersama-sama dengan alloanamnesis juga. Data anamnesis yang berisi identitas pasien, riwayat penyakit pasien, keluhan utama pasien, riwayat penyakit keluarga pasien, serta informasi pasien ini nantinya disimpan di dalam berkas rekam medis (Kementerian Kesehatan Republik Indonesia, 2008).

Rekam medis elektronik (RME) adalah sistem informasi komputer untuk layanan kesehatan yang berisi data medis dan data sosial pasien (Andriani dkk., 2017). Dibandingkan rekam medis berbasis kertas, rekam medis elektronik lebih mudah untuk diolah oleh komputer karena datanya tersimpan dalam format teks *digital* yang dapat dibaca langsung oleh komputer. Rekam medis elektronik juga tidak memiliki kelemahan-kelemahan yang terdapat pada rekam medis kertas. Kelemahan-kelemahan yang dimaksud antara-lain seperti mudah rusak atau hilang, membutuhkan ruangan besar untuk penyimpanannya, dan tulisannya sulit dibaca oleh sebagian orang (Ajami & Bagheri-Tadi, 2013). Data teks keluhan pada rekam medis elektronik umumnya ditulis menggunakan format penulisan bebas atau format penulisan tidak terstruktur yang mempersulit proses penggalian teks pada rekam medis elektronik secara otomatis menggunakan komputer (Pereira dkk., 2015), padahal banyak informasi penting mengenai pasien yang tersimpan pada rekam medis elektronik tersebut yang dapat digunakan sebagai masukan untuk Sistem

Pendukung Keputusan Klinis. Pemeriksaan data rekam medis elektronik secara manual oleh manusia juga sulit dilakukan karena jumlah data rekam medis elektronik yang sangat banyak.

Penelitian ini bertujuan untuk mengembangkan sebuah aplikasi yang mampu menerjemahkan data keluhan pasien yang tersimpan pada rekam medis elektronik ke dalam format teks yang lebih baku dan terstruktur agar teks tersebut dapat lebih mudah dibaca dan diolah oleh komputer. Aplikasi yang dikembangkan pada penelitian ini menggunakan metode-metode Pengolahan Bahasa Alami (Pengolahan Bahasa Alami) dalam melakukan hal tersebut. Pengolahan Bahasa Alami adalah salah satu cabang dari Informatika yang berkaitan dengan pemrosesan cerdas pada bahasa manusia (Ford dkk., 2016). Metode yang digunakan dalam penerjemahan teks keluhan pasien pada aplikasi ini adalah dengan menggunakan metode statistik, dengan metode pencocokan kata-kata atau frasa-frasa kunci, maupun dengan contoh penerjemahan yang diberikan oleh *user*.

Aplikasi yang dikembangkan pada penelitian ini dirancang agar mampu mengenali masukan suara dari pasien maupun petugas kesehatan menggunakan fitur *speech recognition* yang dikembangkan menggunakan Sphinx4 API. Masukan data keluhan pasien selain berupa teks juga dapat dalam bentuk suara yang dimasukkan langsung menggunakan mikrofon maupun dengan mengimpor rekaman suara yang tersimpan di dalam media penyimpanan. *Speech recognition* pada Sphinx4 API menggunakan pendekatan probabilistik. Pada *speech recognition* menggunakan pendekatan pendekatan probabilistik, setiap *input* suara yang dimasukkan bersesuaian terhadap kata atau kumpulan kata tertentu pada kamus dengan sebuah probabilitas tertentu, hasil transkripsi dari *input* suara tersebut adalah kata atau kumpulan kata yang memiliki probabilitas tertinggi (Gruhn dkk., 2011).

## **1.2. Rumusan Masalah**

Berdasarkan latar belakang dari penelitian ini, maka rumusan masalah penelitian ini adalah bagaimana mengembangkan aplikasi yang mampu untuk:

1. Menerjemahkan keluhan pasien dalam format yang kurang baku dan kurang terstruktur menjadi lebih baku dan terstruktur?
2. Mengenali *input* suara pasien atau petugas kesehatan sebagai masukannya?
3. Dilatih agar dapat menerjemahkan keluhan pasien dan mengenali suara pasien atau petugas kesehatan secara lebih akurat?

### **1.3. Batasan Masalah**

Batasan masalah dalam pengembangan aplikasi pada penelitian ini adalah:

1. Aplikasi ini hanya mampu mengenali 600 macam keluhan pasien yang paling umum saja. Hal ini disebabkan karena keterbatasan waktu dan biaya yang digunakan untuk penelitian. Namun *user* dapat melatih aplikasi agar mampu mengenali keluhan-keluhan pasien lainnya.
2. Fungsi *speech recognition* menggunakan Sphinx4 pada aplikasi ini hanya dilatih menggunakan 992 kata atau frasa dan dengan 2 jenis suara saja karena keterbatasan waktu dan biaya. Namun *user* dapat melatih aplikasi agar mampu mengenali suara dengan lebih baik.

### **1.4. Tujuan Penelitian**

Tujuan dari penelitian ini adalah mengembangkan sebuah aplikasi yang mampu untuk:

1. Menerjemahkan keluhan pasien dalam format yang kurang baku dan terstruktur menjadi bentuk yang lebih baku dan terstruktur.
2. Mengenali *input* suara pasien atau petugas kesehatan sebagai masukannya.
3. Dilatih untuk dapat menerjemahkan keluhan pasien dan mengenali suara pasien atau petugas kesehatan dengan lebih akurat

### **1.5. Manfaat Penelitian**

Beberapa manfaat yang dapat diperoleh dari penelitian ini adalah sebagai berikut:

1. Memudahkan pengolahan data pada Sistem Pendukung Keputusan Klinis yang menggunakan data-data keluhan pasien sebagai masukannya.
2. Memudahkan petugas kesehatan dalam menuliskan data keluhan pasien pada rekam medis elektronik sehingga petugas kesehatan tidak perlu menyesuaikan penulisan data keluhan dengan format yang baku.
3. Memudahkan pasien atau petugas kesehatan yang memiliki keterbatasan fisik untuk memasukkan data keluhan pasien cukup dengan menggunakan suara mereka.

## 1.6. Metodologi Penelitian

Dalam penelitian ini, langkah pertama yang dilakukan adalah melakukan studi literatur. Pada langkah studi literatur ini dipelajari ilmu-ilmu kedokteran dasar dan teori-teori mengenai Pengolahan Bahasa Alami yang nantinya digunakan dalam pengembangan aplikasi. Ilmu-ilmu kedokteran dasar dan teori-teori mengenai Pengolahan Bahasa Alami dipelajari baik dari buku-buku maupun dari internet.

Setelah melakukan studi literatur, dilakukan perancangan aplikasi. Dibuat rancangan tampilan antarmuka dan rancangan fitur-fitur apa saja yang terdapat pada aplikasi yang dibuat. Bahasa Pemrograman Java diputuskan digunakan dalam pengembangan aplikasi karena aplikasi yang dibuat menggunakan Bahasa Pemrograman Java bersifat portabel dan karena Sphinx4 API yang digunakan untuk pengembangan fitur *speech recognition* pada aplikasi ini juga dikembangkan dengan Bahasa Pemrograman Java. Untuk pembuatan desain antarmuka, disewa seorang desainer grafis dari Oceano Design Yogyakarta untuk menggambarkan desain latar belakang pada jendela utama aplikasi. Pada langkah ini juga dicatat daftar data-data yang dibutuhkan untuk penelitian dan bagaimana cara memperolehnya.

Setelah mencatat daftar data-data yang dibutuhkan untuk penelitian, kemudian dilakukan langkah pengumpulan data untuk pengembangan aplikasi. Data-data yang dibutuhkan antara lain data daftar kata-kata yang umum digunakan dalam teks Bahasa Indonesia dan daftar kata-kata yang umum digunakan oleh pasien dalam menulis keluhanannya beserta daftar sinonimnya yang nantinya digunakan untuk pengembangan fitur pengoreksi ejaan dan penerjemahan, daftar keluhan yang umum dialami oleh pasien beserta terjemahan istilah medisnya, lokasinya, dan sistem organnya yang nantinya digunakan untuk proses penerjemahan keluhan pasien, serta kamus fonetis, model bahasa, dan model akustik yang digunakan untuk pengembangan fitur *speech recognition*. Daftar kata-kata yang umum digunakan dalam teks Bahasa Indonesia dan daftar kata-kata yang umum digunakan oleh pasien dalam menulis keluhanannya didapatkan dari penggalan teks pada situs Wikipedia Indonesia dan beberapa situs-situs konsultasi kedokteran seperti [klikdokter.com](http://klikdokter.com), [alodokter.com](http://alodokter.com), [dokter.id](http://dokter.id), dll. serta untuk daftar sinonimnya didapatkan dari kamus-kamus Tesaurus baik kamus cetak maupun *online*. Tidak semua kata dapat diperoleh daftar sinonimnya karena keterbatasan waktu dan biaya dalam penelitian. Daftar keluhan yang umum dialami oleh pasien beserta terjemahan istilah medisnya, lokasinya, dan sistem organnya, didapatkan dari situs-situs konsultasi kedokteran, buku-buku

kedokteran dasar, dan dari bertanya kepada beberapa dokter beserta dokter muda dan petugas kesehatan di Rumah Sakit Sardjito Yogyakarta. Data model akustik didapatkan dari merekam suara penulis dan dengan menggunakan fitur *speech synthesis* pada *Google Translate*. Aplikasi hanya dilatih dengan 992 kata atau kumpulan kata yang umum karena keterbatasan waktu dan biaya untuk pembuatan aplikasi, sehingga data kamus fonetis pun hanya berisi 992 kata atau kumpulan kata saja. Data model bahasa dibuat dari daftar kata-kata pada kamus fonetis beserta kombinasi bigram, trigram, maupun 4-gram yang umum dari kata-kata tersebut yang didapatkan dari penggalan teks pada situs *health.detik.com* serta tambahan dari data-data hasil eksperimen yang dibiarkan tersimpan pada model bahasa.

Setelah data-data tersebut dikumpulkan kemudian dilanjutkan ke langkah perbaikan data. Langkah ini dilakukan untuk memperbaiki data-data yang sudah dikumpulkan seperti perbaikan kesalahan ejaan, pengubahan singkatan-singkatan yang tidak baku ke bentuk kepanjangannya, pengabaian kata-kata dalam bahasa asing, pemilihan rekaman suara yang bagus, dan perbaikan rekaman suara dengan memotong bagian awal dan akhir dari rekaman suara tersebut yang hanya berisi *noise* atau *silence* untuk pembuatan model akustik.

Setelah data-data berhasil dikumpulkan dan diperbaiki, dilanjutkan ke langkah pengembangan aplikasi. Aplikasi dikembangkan menggunakan Bahasa Pemrograman Java. Untuk fitur *speech recognition* digunakan Sphinx4 API yang bersifat *open source*. Untuk fitur *speech synthesis* digunakan aplikasi *open source* eSpeak. Aplikasi *open source* FFmpeg digunakan untuk mengonversi masukan suara langsung dari *user* ke format yang dapat diproses oleh Sphinx4 untuk proses *speech recognition*. Setelah aplikasi berhasil dikembangkan, dilakukan pengujian terhadap aplikasi tersebut oleh penulis dan dilakukan perbaikan pada *source code* atau pada data-datanya, seperti perbaikan pada isi kamus-kamus yang digunakan oleh aplikasi untuk proses penerjemahan dan *speech recognition*, dan perbaikan *bug* serta algoritma yang kurang efisien pada *source code*. Setelah itu lima orang dokter diminta untuk mengujicoba aplikasi tersebut dan memberikan penilaian mereka pada kuesioner terhadap aspek kemudahan penggunaan aplikasi tersebut, tampilan antarmuka, tingkat akurasi dan kecepatan pemrosesan fitur penerjemahan, tingkat akurasi dan kecepatan pemrosesan fitur *speech recognition*, serta kemampuan aplikasi ini untuk membantu petugas kesehatan. Kelima dokter tersebut juga diminta untuk memberikan saran mereka bagi pengembangan aplikasi tersebut pada kuesioner.

## **1.7. Sistematika Penulisan**

Berikut adalah sistematika penulisan dan pengelompokan tentang uraian isi dari laporan:

### **BAB I PENDAHULUAN**

Bab ini membahas tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan laporan.

### **BAB II TINJAUAN PUSTAKA**

Bab ini membahas tentang penelitian terdahulu yang sudah pernah dilakukan, pengertian anamnesis, pengertian bahasa alami, dan pengertian pengolahan bahasa alami.

### **BAB III METODOLOGI PENELITIAN**

Bab ini membahas mengenai data-data yang dibutuhkan dalam penelitian ini serta tahapan-tahapan penelitian yang dilakukan untuk pengembangan aplikasi ini.

### **BAB IV PEMODELAN SISTEM**

Bab ini menjelaskan mengenai rancangan model aplikasi yang dikembangkan yang antara lain berupa gambaran model aplikasi secara keseluruhan, model algoritma pembelajaran untuk penerjemahan, model algoritma penerjemahan menggunakan pembelajaran berdasarkan statistik, dan model algoritma pembelajaran untuk *speech recognition*.

### **BAB V HASIL DAN PEMBAHASAN**

Bab ini berisi penjelasan antarmuka aplikasi dan hasil pengujian dari aplikasi yang telah berhasil dikembangkan beserta evaluasi dan pembahasan dari hasil pengujian tersebut.

### **BAB VI KESIMPULAN DAN SARAN**

Berisi kesimpulan yang diperoleh dari hasil penelitian beserta saran-saran yang bermanfaat untuk penelitian yang akan datang.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Penelitian Terdahulu

Sebelum penelitian ini, sudah ada beberapa penelitian serupa mengenai penerapan pengolahan bahasa alami untuk pengolahan data-data teks pada rekam medis. Ratnasari dkk. (2014) pernah melakukan penelitian mengenai pengembangan sebuah model pengolahan bahasa alami untuk perumusan keluhan pasien yang mampu menghasilkan teks dengan makna yang sama dengan teks narasi keluhan pasien, tetapi dalam bahasa yang lebih baku atau dalam bahasa medis yang tepat. Kemudian ada aplikasi pengolahan bahasa alami bernama CLiX ENRICH yang dikembangkan oleh perusahaan Clinithink (*CLiX ENRICH White Paper*, 2014) yang mampu mengekstrak informasi pada teks-teks medis yang ditulis dalam format tidak terstruktur dan kemudian mengodekan ke format SNOMED yang sesuai, aplikasi ini juga memiliki fitur *speech recognition* yang diberi nama Recognosco. The University of Texas Health Science Center at Houston pernah mengembangkan *Clinical Language Annotation, Modeling, and Processing Toolkit* (CLAMP) yaitu sebuah *software* pengolahan bahasa alami klinis yang tersusun dari komponen-komponen pengolahan bahasa alami yang dapat digunakan secara terpisah oleh *user* (CLAMP Team, n. d.). Averbis Health Discovery yang dikembangkan oleh Averbis di Jerman mampu memberikan prediksi diagnosis yang tepat dari masukan teks tidak terstruktur yang menjelaskan kondisi pasien serta mampu memprediksi diagnosis yang tepat untuk penyakit langka dengan menggunakan *machine learning* ("Averbis Health Discovery - Analysis of patient data", n.d.). Canary adalah sebuah aplikasi pengolahan bahasa alami klinis *open source* yang dikembangkan oleh Dr. Alexander Turchin's *research group* di Brigham and Women's Hospital yang didesain agar mudah digunakan oleh *user* yang tidak memiliki latar belakang ilmu komputer atau pemrograman, dan awalnya didesain khusus untuk penggunaan klinis namun dapat dikembangkan untuk penggunaan di bidang lainnya juga (Malmasi dkk., 2017)

Perbandingan kelima aplikasi pada penelitian sebelumnya dengan aplikasi yang dikembangkan pada penelitian ini dapat dilihat pada tabel 2.1. di bawah ini:



Tabel 2.1. Perbandingan Aplikasi yang Dikembangkan pada Penelitian ini dengan Aplikasi pada Penelitian Sebelumnya

	<b>Aplikasi yang Dikembangkan</b>	<b>Aplikasi oleh Ratnasari dkk.</b>	<b>CLiX ENRICH</b>	<b>CLAMP</b>	<b>Averbis</b>	<b>Canary</b>
<b>Fitur <i>Speech recognition</i></b>	Ada, menggunakan Sphinx4	Tidak Ada	Ada, menggunakan Recognosco	Tidak Ada	Ada	Tidak Ada
<b>Mempelajari Istilah Baru</b>	Bisa	Tidak Bisa	Tidak Bisa	Bisa	Bisa	Bisa
<b>Offline/Online</b>	<i>Offline</i>	<i>Online</i>	<i>Offline</i> dan dapat berbasis <i>Cloud</i>	<i>Offline</i>	<i>Offline</i>	<i>Offline</i>
<b>Lisensi</b>	<i>Closed Source</i>	<i>Closed Source</i>	<i>Closed Source</i>	Gratis untuk akademik. <i>Source Code</i> dapat diminta oleh orang yang memiliki izin.	<i>Closed Source</i>	Gratis dan <i>Open Source</i>
<b>Bahasa Alami Utama</b>	Bahasa Indonesia	Bahasa Indonesia	Bahasa Inggris	Bahasa Inggris	Bahasa Inggris dan Bahasa Jerman	Bahasa Inggris. Namun sudah diujicoba dengan bahasa lainnya.

Berdasarkan perbandingan pada tabel 2.1. di atas dapat disimpulkan bahwa aplikasi yang dikembangkan pada penelitian ini memiliki kelebihan fitur *speech recognition* yang tidak dimiliki oleh beberapa aplikasi yang pernah dikembangkan sebelumnya, yaitu aplikasi yang dikembangkan oleh Ratnasari dkk., CLAMP, dan Canary. Aplikasi yang dikembangkan pada penelitian ini juga mampu mempelajari istilah baru yang tidak dimiliki oleh aplikasi yang dikembangkan oleh Ratnasari dkk. dan CLiX ENRICH, serta komputer *user* tidak perlu terhubung ke internet karena aplikasi ini digunakan secara *offline*. *Source code* aplikasi yang dikembangkan pada penelitian ini bersifat *closed source*, namun *source code* nya suatu saat kemungkinan dapat diberikan kepada seseorang yang telah mendapatkan izin. Beberapa aplikasi yang dikembangkan pada penelitian sebelumnya umumnya menggunakan bahasa alami utama Bahasa Inggris sedangkan bahasa alami

utama pada aplikasi yang dikembangkan pada penelitian ini adalah Bahasa Indonesia, sama seperti aplikasi yang dikembangkan oleh Ratnasari dkk.

## **2.2. Anamnesis**

Anamnesis adalah sebuah kegiatan wawancara antara dokter atau petugas kesehatan lainnya terhadap pasien atau orang-orang yang memiliki hubungan dekat dengan pasien untuk memperoleh keterangan-keterangan mengenai keluhan dan penyakit yang diderita oleh pasien (Wahid, 2012). Tujuan anamnesis adalah untuk mendapatkan informasi menyeluruh dari pasien yang bersangkutan seperti data medis organobiologis, psikososial, serta lingkungan pasien. Dengan menggunakan informasi tersebut, pada akhir anamnesis diharapkan dokter mampu menyimpulkan dugaan organ atau sistem organ yang terganggu, termasuk rumusan masalah kliniknya. Tujuan anamnesis lain yang juga penting adalah untuk menjalin hubungan antara dokter dengan pasien yang profesional dan optimal (Markum, 2007).

Anamnesis terbagi menjadi dua macam, yaitu autoanamnesis dan alloanamnesis atau heteroanamnesis (Markum, 2007). Autoanamnesis adalah anamnesis yang dilakukan secara langsung terhadap pasien itu sendiri karena pasien dianggap mampu untuk melakukan wawancara. Jika pasien dianggap tidak mampu melakukan wawancara maka orang lain yang mengenal pasien dapat menggantikan pasien dalam melakukan wawancara yang disebut sebagai alloanamnesis atau heteroanamnesis. Pada umumnya alloanamnesis atau heteroanamnesis dilakukan terhadap pasien yang masih anak-anak, pasien difabel, pasien yang sedang dalam kondisi tidak sadar, atau pasien yang sedang mengalami gangguan kejiwaan. Dalam praktek sehari-hari tidak jarang juga alloanamnesis dilakukan bersama-sama dengan autoanamnesis.

Sebuah anamnesis yang baik sebaiknya berisi beberapa kelompok data penting sebagai berikut agar dalam melakukan anamnesis tidak ada pertanyaan yang terlewatkan sehingga semua keluhan pasien dapat ditampung (Markum, 2007):

1. Identitas pasien
2. Riwayat penyakit sekarang (didahului keluhan utama)
3. Riwayat penyakit dahulu
4. Riwayat kesehatan keluarga
5. Riwayat pribadi, psikologis, sosial-ekonomi-dan-budaya

Penjelasan dari poin-poin di atas dituliskan di bawah ini.

a. Identitas Pasien

Identitas pasien umumnya meliputi nama lengkap, alamat lengkap, usia, pekerjaan, jenis kelamin, status perkawinan, suku, agama, serta nomor telepon pasien yang dapat dihubungi. Data identitas pasien ini dibutuhkan oleh petugas kesehatan karena beberapa penyakit tertentu terkadang berkaitan dengan usia, jenis kelamin, profesi, status perkawinan, atau suku bangsa tertentu. Data identitas pasien juga berguna untuk memberikan kesan mengenai keadaan sosial ekonomi, budaya, dan lingkungan pasien, serta untuk dapat melakukan komunikasi secara berkesinambungan terhadap pasien. Dengan data identitas pasien, petugas kesehatan dapat merencanakan pengelolaan pasien yang lebih cepat, optimal, dan sesuai dengan kondisi pasien secara menyeluruh.

b. Keluhan Utama

Keluhan utama adalah keluhan yang menyebabkan pasien meminta pertolongan kepada petugas kesehatan. Keluhan utama ini umumnya tertulis secara ringkas beserta lama kemunculannya. Misalnya: “sakit tenggorokan sejak tiga minggu yang lalu.”

c. Riwayat Penyakit Sekarang

Riwayat Penyakit Sekarang (RPS) adalah riwayat mengenai penyakit pasien sekarang, yang dimulai sejak akhir masa sehat. Pada RPS dicatat perkembangan dan perjalanan penyakit yang ditulis secara kronologis sesuai urutan waktunya, termasuk riwayat pengobatan atau perawatan untuk penyakit yang sedang dialami tersebut. Pada RPS juga tertulis deskripsi atau analisis terhadap keluhan utama dan keluhan penting lainnya beserta lokasi dan penjalaran, kualitas, kuantitas (tipe onset, intensitas/keparahan, disabilitas), kronologis, kondisi atau keadaan saat munculnya gejala (onset, durasi, periodisitas, frekuensi), faktor yang mempengaruhi (faktor pencetus, faktor yang memperberat, faktor yang memperingan, kaitannya dengan aktivitas sehari-hari), dan gejala penyerta (Fakultas Kedokteran UI, 2009):

d. Riwayat Penyakit Dahulu

Pada Riwayat Penyakit Dahulu (RPD) dicatat daftar penyakit yang dahulu pernah dialami pasien beserta waktu kemunculannya, termasuk apakah pasien pernah mengalami operasi, kecelakaan, ataupun keadaan alergi. Riwayat penyakit dahulu mencakup anamnesis penyakit sistem pernapasan, sistem kardiovaskuler, sistem muskuloskeletal, saluran cerna, kulit, infeksi, endokrinologi, hematologi, saluran kemih, ginjal, dll. RPD merupakan data penting karena memberikan informasi kepada petugas kesehatan mengenai kemungkinan adanya gejala sisa dan kemungkinan adanya kaitan penyakit dahulu dengan penyakit sekarang, kemungkinan adanya gejala sisa dari penyakit yang dahulu diderita pasien, dan kemungkinan adanya pengaruh terhadap pengelolaan pasien selanjutnya.

e. Riwayat Penyakit Keluarga

Pada Riwayat Penyakit Keluarga, kondisi kesehatan dan umur masing-masing anggota keluarga dituliskan bila masih hidup. Bila anggota keluarga sudah meninggal, dituliskan umur saat meninggal dan penyebab meninggalnya. Anggota keluarga ini meliputi ayah, ibu, saudara laki-laki, saudara perempuan, kakek, nenek, dan anak-anak pasien. Penyakit-penyakit yang kemungkinan disebabkan oleh faktor hereditas atau oleh penularan anggota keluarga dicari dan dicatat pada Riwayat Penyakit Keluarga.

f. Riwayat Pribadi, Psikologis, Sosial Ekonomi dan Budaya

Riwayat Pribadi, Psikologis, Sosial Ekonomi dan Budaya ini berisi data mengenai keterangan kelahiran pasien, peristiwa penting pada masa kecil pasien, sikap pasien terhadap keluarga dekat, keterangan pendidikan pasien, keterangan pekerjaan pasien, aktivitas sehari-hari pasien, lingkungan tempat tinggal pasien, gaya hidup pasien, kondisi perkawinan pasien, masalah-masalah yang dihadapi pasien, dll. Data riwayat pribadi, psikososial, ekonomi, budaya dan keluarga ini merupakan informasi penting yang dapat digunakan sebagai bahan pertimbangan dalam pengelolaan yang optimal dan menyeluruh terhadap pasien untuk saat ini maupun selanjutnya.

Data-data anamnesis nantinya akan tersimpan di dalam rekam medis (Kemenkes RI, 2008). Berdasarkan Peraturan Menteri Kesehatan Republik Indonesia no. 269/MENKES/PER/ III/2008 tentang Rekam Medis, rekam medis adalah berkas yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, tindakan dan pelayanan lain yang telah diberikan kepada pasien. Menurut Peraturan

Menteri Kesehatan Republik Indonesia no. 269/MENKES/PER/ III/2008 tentang Rekam Medis, rekam medis dapat dipakai untuk hal-hal sebagai berikut:

- a. Pemeliharaan kesehatan dan pengobatan pasien;
- b. Alat bukti dalam proses penegakan hukum, disiplin kedokteran dan kedokteran gigi dan penegakan etika kedokteran dan etika kedokteran gigi;
- c. Keperluan pendidikan dan penelitian;
- d. Dasar pembayaran biaya pelayanan kesehatan
- e. Data statistik kesehatan

Rekam medis berbasis elektronik atau disebut RME menyimpan informasi-informasi pasien pada komputer bukan pada kertas. Rekam medis berbasis kertas memiliki beberapa kelemahan seperti mudah rusak, mudah hilang, membutuhkan tempat penyimpanan yang besar, dan tulisannya sulit dibaca oleh sebagian orang (Ajami dan Bagheri-Tadi, 2013). Karena pada rekam medis elektronik data medis dicatat dan disimpan menggunakan komputer, maka kelemahan-kelemahan pada rekam medis berbasis kertas tersebut dapat diminimalisasi. Rekam medis elektronik mampu mengurangi biaya penyimpanan dan penyalinan rekaman medis, meningkatkan dokumentasi, memberikan efisiensi dalam penanganan rekaman medis, serta meningkatkan kecepatan untuk mengakses rekaman medis (Bilimoria, 2007)

### **2.3. Bahasa Alami**

Bahasa adalah sebuah cara berkomunikasi secara lisan maupun tulisan yang terdiri dari kata-kata yang terstruktur, kumpulan-kumpulan simbol (seperti huruf, angka, dan karakter khusus), serta kumpulan-kumpulan aturan yang mengatur penyusunan dan penggunaan dari kata-kata dan simbol-simbol tersebut secara konvensional (Iroju dan Olakeke, 2015). Bahasa alami adalah bahasa yang manusia pelajari dari lingkungan mereka dan digunakan untuk berkomunikasi dengan sesamanya. Bagaimana pun bentuk komunikasinya, bahasa-bahasa alami digunakan untuk mengekspresikan pengetahuan dan emosi seseorang, serta untuk menyampaikan respons ke orang-orang lain atau lingkungan sekitarnya. Bahasa alami biasanya dipelajari pada awal masa kanak-kanak dari orang-orang sekitarnya (Reshamwala dkk., 2013). Contoh bahasa alami adalah bahasa Indonesia, Bahasa Jawa, dan Bahasa Inggris. Bahasa alami berbeda dengan bahasa-bahasa buatan dan

bahasa formal seperti bahasa-bahasa yang digunakan untuk menuliskan program komputer atau bahasa-bahasa yang digunakan untuk mempelajari ilmu logika (Lyons, 1991).

Bahasa alami berkembang secara alamiah untuk memenuhi kebutuhan komunikasi antar manusia dan tidak secara sengaja dirancang untuk kemudahan pemrosesan oleh komputer. Oleh karena itu, pengolahan bahasa alami oleh komputer jauh lebih sulit dilakukan dibandingkan dengan pengolahan bahasa buatan oleh komputer. Beberapa karakteristik umum bahasa alami yang menyebabkannya sulit diproses oleh komputer antara lain (Utami dan Hartati, 2007):

1. Sering terjadi ambiguitas dalam bahasa alami.
2. Jumlah kosa kata dalam bahasa alami sangat besar dan berkembang dari waktu ke waktu.

Ambiguitas dalam bahasa alami adalah adanya lebih dari satu kemungkinan arti pada teks maupun ucapan bahasa alami. Jenis-jenis ambiguitas dalam bahasa alami antara lain (Reshamwala dkk., 2013).

1. Ambiguitas Sintaksis

Ambiguitas sintaksis terjadi ketika adanya lebih dari satu cara untuk melakukan *parsing* pada sebuah kalimat. Contohnya adalah pada kalimat "Polisi menangkap penjahat dengan topi merah". Frasa "topi merah" bisa memiliki kemungkinan arti sebagai bagian dari frasa nomina yang menjelaskan si penjahat, namun bisa juga memiliki kemungkinan arti sebagai frasa preposisi yang menjelaskan cara polisi menangkap si penjahat tersebut, yaitu dengan menggunakan topi merah.

2. Ambiguitas Semantik

Ambiguitas Semantik terjadi jika sebuah teks mengandung kata atau frasa yang dapat memiliki dua arti. Contohnya pada kalimat "Dia sakit hati.". Kata "hati" pada kalimat tersebut bisa memiliki arti salah satu organ tubuh manusia atau juga bisa memiliki arti perasaan manusia.

3. Ambiguitas Referensial

Ambiguitas referensial terjadi ketika suatu kalimat mengacu pada suatu kata tanpa secara eksplisit menyebut kata tersebut dan menggunakan kata ganti. Contohnya pada kalimat "Andi bertengkar dengan Budi hingga dia terluka". Kata "dia" pada kalimat tersebut memiliki kemungkinan mengacu kepada "Andi" atau "Budi".

#### 4. Ambiguitas Lokal

Ambiguitas lokal terjadi ketika suatu bagian dari sebuah kalimat kurang jelas artinya secara lokal namun keseluruhan kalimat membuatnya menjadi lebih jelas artinya. Contohnya adalah pada bagian kalimat ini "Dia datang bulan...". Penggalan kalimat tersebut memiliki arti kurang jelas, namun setelah dilengkapi menjadi kalimat "Dia datang bulan Januari" atau menjadi kalimat "Dia datang bulan tidak teratur" penggalan kalimat tersebut menjadi lebih jelas artinya. Pada contoh kalimat pertama "datang bulan" artinya adalah "datang pada saat bulan...", sedangkan pada contoh kalimat kedua "datang bulan" artinya adalah "menstruasi".

#### 2.3.1. Tata Bahasa

Tata bahasa terbagi menjadi dua macam pengertian menurut Keraf (1999). Pertama, cabang ilmu bahasa yang mempelajari dan mendeskripsikan kaidah-kaidah yang menjadi dasar bentuk sebuah bahasa; dan kedua, semacam buku yang memuat himpunan kaidah dan patokan umum mengenai struktur suatu bahasa. Tata Bahasa mencakup beberapa aspek antara lain fonologi, morfologi, sintaksis, dan semantik (Kridalaksana, 2001). Berikut ini dijelaskan hal-hal yang berkaitan dengan aspek-aspek tersebut.

##### a. Fonologi

Fonologi adalah bidang linguistik yang mempelajari, menganalisis, dan membicarakan runtunan bunyi-bunyi bahasa (Achmad dan Abdullah, 2012). Fonologi dapat dibagi menjadi fonetik dan fonemik. Fonetik adalah cabang studi fonologi yang mempelajari bunyi bahasa tanpa memedulikan apakah bunyi-bunyi tersebut berfungsi sebagai pembeda makna atau tidak. Objek kajian fonetik adalah fon, yaitu bunyi yang tidak memperhatikan fungsinya sebagai pembeda makna. Fonemik adalah cabang studi fonologi yang mempelajari bunyi bahasa dengan memperhatikan fungsi bunyi tersebut sebagai pembeda makna. Objek kajian fonemik adalah fonem, yaitu bunyi bahasa yang membedakan makna suatu kata, jika suatu bunyi tidak membedakan makna maka bunyi tersebut bukanlah fonem. Sebagai contoh huruf 'k' pada kata "botak" pada Bahasa Indonesia dapat diucapkan dengan bunyi *glottal stop* maupun dengan bunyi *voiceless velar stop*, namun kedua bunyi tersebut tidak membedakan makna suatu kata, sehingga kedua bunyi tersebut tidak dianggap sebagai fonem berbeda. Sedangkan huruf 'e' pada kata "apel" memiliki arti berbeda jika huruf 'e' tersebut diucapkan sebagai e pepet dan jika

huruf ‘e’ tersebut diucapkan sebagai e taling, jika huruf ‘e’ pada kata “apel” diucapkan sebagai e pepet maka makna kata “apel” tersebut adalah sejenis buah-buahan, sedangkan jika huruf ‘e’ pada kata “apel” diucapkan sebagai e taling maka makna kata “apel” tersebut adalah upacara. Oleh karena itu e pepet dan e taling dianggap sebagai fonem berbeda karena dapat membedakan makna. Setiap bahasa memiliki jumlah fonem yang berbeda-beda, menurut Achmad dan Abdullah (2012) jumlah fonem dalam Bahasa Indonesia ada 24 buah, yang terdiri dari 6 buah fonem vokal (yakni a, i, u, e, o, dan ə) dan 18 buah fonem konsonan (yakni, p, t, c, k, b, d, j, g, m, n, ŋ, ɲ, s, h, r, l, w, dan y). Bunyi-bunyi yang merupakan realisasi dari sebuah fonem, seperti bunyi [k] dan [ʔ] untuk fonem Bahasa Indonesia /k/ pada kata “bakti” dan kata “bapak” disebut sebagai alofon-alofon dari fonem /k/. Fonem dapat dibagi menjadi fonem segmental dan fonem suprasegmental. Fonem segmental terdiri atas fonem vokal dan fonem konsonan, sedangkan fonem suprasegmental terdiri atas tekanan, durasi, dan nada.

## **b. Morfologi**

Morfologi adalah cabang ilmu linguistik yang menyelidiki struktur kata, bagian-bagiannya, serta cara pembentukannya (Achmad dan Abdullah, 2012). Morfem adalah satuan bahasa terkecil dalam tataran morfologis yang tidak dapat dibagi lagi menjadi bagian bermakna yang lebih kecil. Bentuk-bentuk realisasi yang berbeda dari sebuah morfem yang sama disebut alomorf. Misalnya morfem me- memiliki bentuk realisasi mem- pada kata "membawa", men- pada kata "mencegah", meny- pada kata "menyuap" dan meng- pada kata "menggunting". Morfem dapat dibedakan menjadi dua macam yaitu morfem bebas dan morfem terikat. Morfem bebas mampu berdiri sendiri sebagai kata, sedangkan morfem terikat tidak mampu untuk berdiri sendiri dan harus selalu disatukan dengan morfem lainnya melalui proses morfologis atau pembentukan kata. Semua imbuhan dalam Bahasa Indonesia adalah morfem terikat. Kata-kata seperti "renta" yang hanya muncul dalam "tua renta" dan "kerontang" yang hanya muncul dalam "kering kerontang" juga termasuk morfem terikat. Morfem dapat berbentuk morfem utuh maupun morfem terbagi. Morfem terbagi contohnya adalah konfiks seperti ke-...-an pada kata "kesatuan" dan konfiks per-...-an pada kata perbaikan. Infiks dapat mengubah morfem utuh menjadi morfem terbagi, misalnya morfem “getar” jika diberi infiks “-em-“ maka menjadi kata “gemetar” yang membagi morfem “getar” tersebut. Pada beberapa bahasa terdapat morfem suprasegmental, yaitu morfem yang dibentuk oleh unsur-unsur suprasegmental seperti nada, tekanan, dll. Pada Bahasa Mandarin misalnya, kata "ma" dapat memiliki arti



"kuda", "ibu", atau "memarahi" tergantung nada dalam pengucapannya, sehingga kata "ma" dalam Bahasa Mandarin merupakan morfem suprasegmental karena mengantung unsur suprasegmental yaitu nada. Bahasa Indonesia sendiri tidak memiliki morfem suprasegmental.

Menurut Kamus Besar Bahasa Indonesia (2008), kata adalah satuan (unsur) bahasa yang terkecil yang dapat diujarkan sebagai bentuk yang bebas; satuan (unsur) bahasa yg berupa morfem bebas. Kata adalah satuan bahasa terbesar dalam tataran morfologis. Proses pembentukan kata-kata dari satuan lain yang merupakan dasarnya sering disebut juga proses morfologi. Ada tiga macam proses morfologi dalam Bahasa Indonesia yaitu afiksasi, reduplikasi, dan komposisi (Ramlan, 1985). Afiksasi adalah proses penambahan imbuhan pada kata dasar yang dapat dibedakan menjadi prefiks, infiks, sufiks, dan konfiks atau simulfiks. Reduplikasi adalah pengulangan bentuk dasar atau satuan gramatik menjadi sebuah kata ulang. Komposisi adalah penggabungan kata-kata yang membentuk kata majemuk.

Kelas kata ialah pengelompokan kata berdasarkan perilaku atau sifat kata tersebut dalam kalimat. Secara umum, kelas kata terbagi menjadi berikut ini:

1. Nomina

Nomina atau kata benda adalah kata yang mengacu kepada suatu benda yang bersifat konkret maupun abstrak. Contoh nomina adalah rumah, mobil, akal, dan hati.

2. Verba

Verba atau kata kerja adalah kata-kata yang menyatakan perbuatan atau tindakan. Contoh verba adalah berlari, memakan, dan tertidur.

3. Adjektiva

Adjektiva atau kata sifat merupakan kata yang menyatakan sifat atau keadaan dari suatu nomina (kata benda) atau suatu pronomina (kata ganti). Contoh adjektiva adalah merah, bulat, kasar, dan besar.

4. Adverbia

Adverbia atau kata keterangan adalah kata yang memberi keterangan pada verba, adjektiva, nomina predikatif, atau kalimat. Contoh adverbia adalah sangat, terlalu, agak, dan cukup.

5. Pronomina

Pronomina atau kata ganti adalah kata yang digunakan untuk menggantikan benda atau sesuatu yang dibendakan. Contohnya adalah aku, kita, ini, dan sesuatu.

6. Numeralia

Numeralia atau kata bilangan adalah kata yang menyatakan jumlah, kumpulan, dan urutan atau tingkatan suatu benda atau sesuatu yang dibedakan. Contohnya adalah satu, seratus, kedua, dan sebagian.

7. Kata Tugas

Kata tugas hanya mempunyai arti gramatikal dan tidak memiliki arti leksikal. Kata tugas umumnya merupakan kelas kata yang keanggotaannya tertutup yaitu hampir tidak pernah ada penambahan kata baru pada kelas kata ini, dan kebanyakan tidak mengalami proses morfologis. Kata tugas terbagi menjadi artikel (kata sandang), preposisi (kata depan), konjungsi (kata hubung), partikel, dan interjeksi (kata seru). Artikel atau kata sandang adalah kata yang mendampingi kata benda atau yang membatasi makna jumlah orang atau benda seperti sang dan si. Preposisi atau kata depan adalah kata yang selalu berada di depan kata benda, kata sifat, atau kata kerja untuk membentuk frasa preposisional seperti ke, dari, dan sekitar. Konjungsi atau kata hubung adalah kata tugas yang menghubungkan antar klausa, antar kalimat, atau antar paragraf seperti karena, tetapi, dan sehingga. Partikel adalah kata yang bertugas memulai, mempertahankan, atau mengukuhkan sebuah kalimat dalam komunikasi seperti lah, dong, dan deh. Dan terakhir yaitu interjeksi atau kata seru ialah kata yang mengungkapkan perasaan seperti wah, hai, dan yes.

**c. Sintaksis**

Sintaksis mempelajari kata, frasa, klausa, dan kalimat, serta hubungannya dan cara penyusunannya hingga menjadi satu ujaran yang tepat (Achmad dan Abdullah, 2012). Seperangkat aturan yang mengatur deretan kata-kata yang membentuk kalimat disebut sebagai alat sintaksis. Alat sintaksis yang mengatur unsur-unsur bahasa sehingga terbentuk satuan bahasa yang disebut kalimat antara lain urutan kata, kata tugas, bentuk kata, dan intonasi.

Bidang kajian sintaksis adalah frasa, klausa, dan kalimat. Frasa adalah satuan konstruksi yang terdiri dari dua atau lebih kata yang membentuk satu kesatuan (Keraf, 1984). Sebanyak apapun kata penyusun frasa, selama tidak melampaui batas fungsi atau jabatannya baik sebagai subjek, predikat, objek, pelengkap, maupun keterangan, maka masih bisa disebut sebagai frasa (Ramlan, 2001). Menurut Keraf (1984), klausa adalah sebuah konstruksi yang di dalamnya terdapat beberapa kata yang mengandung unsur predikatif. Kalimat adalah satuan bahasa terkecil yang merupakan kesatuan pikiran

(Widjono, 2007). Manaf (2009) menjelaskan bahwa yang membedakan klausa dan kalimat adalah klausa tidak diakhiri intonasi final, sedangkan kalimat diakhiri dengan intonasi final yang dapat berupa intonasi berita, intonasi tanya, intonasi perintah, atau intonasi kagum.

Fungsi sintaksis adalah wadah yang dapat diisi oleh bentuk bahasa tertentu (Manaf, 2009). Subjek dan predikat adalah fungsi sintaksis yang wajib ada pada setiap kalimat, sedangkan fungsi sintaksis objek, pelengkap, dan keterangan tidak wajib ada pada setiap kalimat. Subjek merupakan unsur utama suatu kalimat yang dijelaskan oleh fungsi sintaksis predikat. Predikat merupakan unsur kalimat yang menjelaskan subjek. Objek adalah unsur kalimat yang melengkapi verba transitif pada suatu kalimat aktif. Pelengkap berfungsi melengkapi predikat dan tidak dapat diubah menjadi subjek. Keterangan adalah unsur tambahan dalam kalimat yang berfungsi memperjelas atau memberi keterangan pada seluruh kalimat.

#### **d. Semantik**

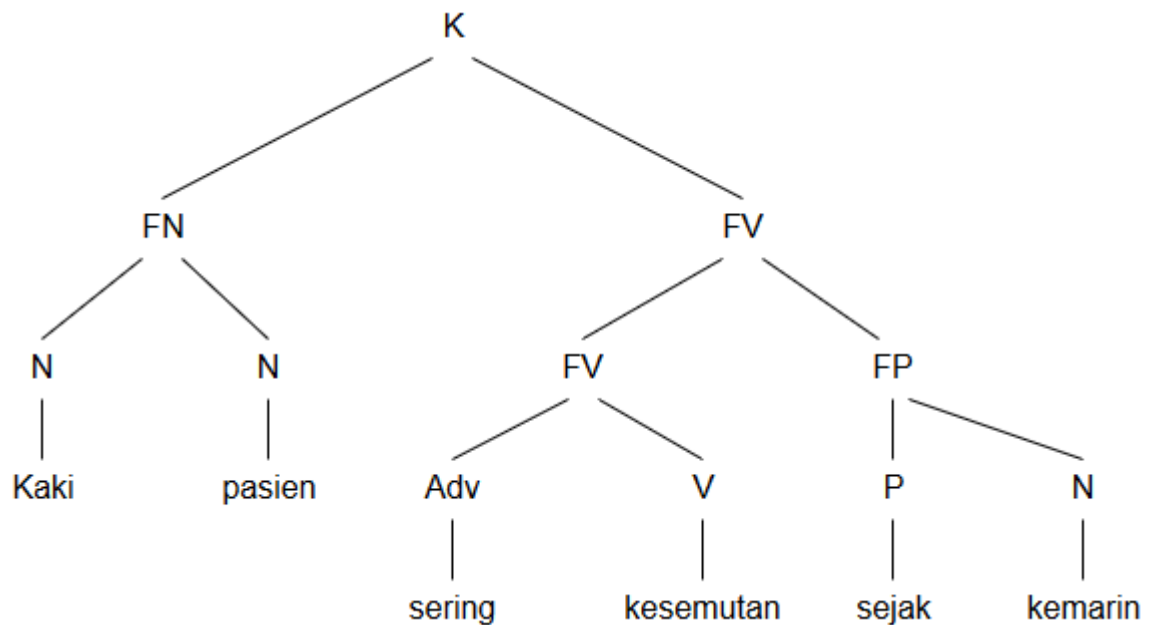
Semantik adalah cabang ilmu linguistik yang mempelajari arti atau makna suatu kata (Achmad dan Abdullah, 2012). Arti kata dapat dibagi menjadi tiga yaitu arti leksikal, arti gramatikal, dan arti idiomatik. Arti leksikal adalah arti kata menurut kamus, yang umumnya dianggap sejajar dengan arti denotatif. Arti gramatikal muncul setelah proses gramatika, misalnya morfem ber- jika berdiri sendiri tidak memiliki makna, morfem tersebut baru memiliki makna setelah bergabung dengan morfem lainnya. Arti idiomatik muncul karena beberapa kata bergabung membentuk sebuah makna baru yang tidak dapat ditelusuri dari kata-kata pembentuknya, contohnya adalah "buah tangan" yang memiliki arti "oleh-oleh" dan "besar kepala" yang memiliki arti "sombong".

#### **2.3.2. Parsing**

*Parsing* adalah proses pengonversian sebuah kalimat menjadi sebuah pohon (dinamakan *parse tree*) yang merepresentasikan struktur sintaksis dari kalimat tersebut (Reshamwala dkk., 2013). *Parse tree* dapat dihasilkan dari kalimat-kalimat pada bahasa alami maupun dari pemrosesan bahasa komputer seperti bahasa pemrograman. Kalimat "Mobil hitam berhenti di jalan" terdiri dari frasa nominal "Mobil hitam" dan frasa verbal "berhenti di jalan". Pada *top-down parsing*, pohon kalimat tersebut memiliki akar kalimat tersebut dan kemudian bercabang menjadi frasa nominal dan frasa verbal, dan kemudian akhirnya bercabang kembali menjadi nomina-nomina, adjektiva, verba, dan preposisi.

*Parsing* menentukan apakah sebuah kalimat adalah valid menurut aturan-aturan tata bahasa sebuah bahasa. Ada dua macam teknik *parsing*, yaitu *top-down* dan *bottom-up*.

*Top down parser* memulai pemeriksaan dari tingkatan tertinggi pada *parse tree* dan bekerja ke bawah mencari *node* terminal sesuai dengan kelas kata dari kata-kata pada kalimat masukan. Gambar 2.1. di bawah ini adalah contoh *parse tree* yang dibentuk oleh *top down parser*. Awalnya kalimat "Kaki pasien sering kesemutan sejak kemarin" bercabang menjadi frasa nominal "Kaki pasien" dan frasa verbal "sering kesemutan sejak kemarin". Kemudian frasa nominal "Kaki pasien" bercabang kembali menjadi nomina "Kaki" dan nomina "pasien", dan frasa verbal " sering kesemutan sejak kemarin" bercabang kembali menjadi frasa verbal "sering kesemutan" dan frasa preposisional "sejak kemarin". Dan terakhir frasa verbal "sering kesemutan" bercabang menjadi adjektiva "sering" dan verba "kesemutan", dan frasa preposisional "sejak kemarin" bercabang menjadi preposisi "sejak" dan nomina "kemarin".



Gambar 2.1. Contoh *Parse Tree*.

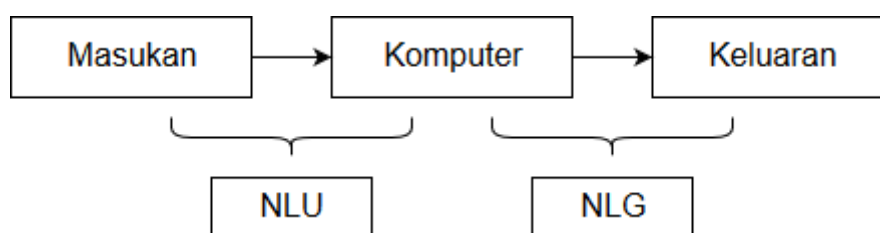
Bottom up parser merupakan kebalikan dari top down parser, yaitu memulai pemeriksaan dari *node-node* terminal dan bekerja ke atas mencari *node* akar dari *parse tree*. Pada contoh gambar 4 di atas, bottom up parser melakukan pencarian dari nomina "Kaki", nomina "pasien", adjektiva "sering", verba "kesemutan", preposisi "sejak" dan verba

“kesemutan” menuju ke arah pembentukan frasa-frasa yang membentuk kalimat pada akar, hingga ke pembentukan kalimat pada akarnya.

#### 2.4. Pengolahan Bahasa Alami

Pengolahan Bahasa Alami atau juga dikenal sebagai Komputasi Linguistik adalah sebuah cabang Ilmu Komputer yang mengurus mengenai penggunaan teknik-teknik komputasi untuk mempelajari, memahami, dan memproduksi bahasa alami manusia (Hirschberg dan Manning, 2015). Pengolahan Bahasa Alami memiliki beberapa tujuan, yaitu membantu komunikasi antar manusia seperti pada *Machine Translation*, membantu komunikasi antara manusia dengan mesin misalnya pada *speech recognition* dan *speech synthesis*, atau memudahkan manusia dan mesin dengan belajar dan menganalisis dari konten-konten bahasa alami manusia yang saat ini tersedia dalam jumlah besar secara *online*. Pemrosesan Bahasa Alami merupakan bidang ilmu multidisipliner yang merupakan gabungan dari bidang ilmu Ilmu Komputer, Linguistik, Sains Kognitif, Psikologi, Filsafat, dan Logika Matematika (Iroju dan Olakeke, 2015).

Pemrosesan Bahasa Alami terbagi menjadi dua tugas utama (Regina dan Michael, 2005), yaitu *Natural Language Understanding* (NLU) dan *Natural Language Generation* (NLG) yang diilustrasikan pada gambar 2.2. di bawah. Sesuai namanya, NLU adalah sebuah proses pemahaman bahasa alami oleh komputer. NLU mengonversi sebuah masukan bahasa alami manusia ke bentuk yang dapat dipahami oleh komputer (Saetre, 2004). NLG adalah proses pengkonstruksian sebuah bahasa alami oleh komputer untuk tujuan komunikasi tertentu (Dale, 2006).



Gambar 2.2. Tugas utama pengolahan bahasa alami (Regina dan Michael, 2005)

Beberapa tahun belakangan ini Pengolahan Bahasa Alami sudah berkembang menjadi area riset yang menarik dan beberapa teknologi terapannya telah digabungkan pada produk-produk konsumen seperti Siri pada produk-produk Apple dan *Skype*

*Translator*. Empat faktor kunci menjadi penyebab perkembangan pesat Pengolahan Bahasa Alami beberapa tahun belakangan ini (Hirschberg dan Manning, 2015), yaitu peningkatan kemampuan komputer yang pesat, ketersediaan data-data linguistik yang melimpah yang salah satunya disebabkan oleh berkembangnya internet, kesuksesan pengembangan metode-metode *machine learning*, dan pemahaman yang lebih mendalam dari struktur bahasa manusia. Salah satu keterbatasan Pengolahan Bahasa Alami saat ini adalah sebagian besar sumber daya dan sistem Pengolahan Bahasa Alami hanya tersedia untuk *high-resource languages* (HRL), seperti Bahasa Inggris, Bahasa Prancis, Bahasa Spanyol, Bahasa Jerman, dan Bahasa Mandarin. Banyak *low-resource languages* (LRL) seperti Bahasa Indonesia, Bahasa Bengali, Bahasa Cebuano, dan Bahasa Swahili tidak memiliki sistem atau sumber daya yang cukup meskipun diucapkan dan ditulis oleh jutaan orang (Hirschberg dan Manning, 2015). Di masa depan ilmuwan berharap dapat mengembangkan sumber daya dan sistem untuk ratusan ribu atau jutaan bahasa, tidak hanya beberapa saja.

#### **2.4.1. Tingkatan Analisis Pengolahan Bahasa Alami**

Pengolahan Bahasa Alami biasanya dikerjakan pada tingkatan-tingkatan yang dideskripsikan di bawah ini (Kumar, 2012):

##### **1. Analisis Fonologi**

Tingkatan ini menangani interpretasi suara-suara pembicaraan di dalam dan antar kata-kata. Ada tiga aturan yang digunakan dalam analisis fonologi, yaitu aturan fonetik, aturan fonemik, dan aturan prosodi. Aturan fonetik menangani tentang suara-suara di dalam kata. Aturan fonemik menangani tentang variasi-variasi dari pengucapan ketika kata-kata diucapkan bersama-sama. Sedangkan aturan prosodi menangani fluktuasi pada tekanan dan intonasi dalam sebuah kalimat. Dalam sebuah sistem pengolahan bahasa alami yang menerima masukan suara, gelombang suara masukan dianalisis dan dikodekan ke bentuk sinyal digital untuk penginterpretasian menggunakan berbagai aturan tersebut.

##### **2. Analisis Morfologi**

Kata-kata dalam sebuah teks dapat tersusun dari satu atau lebih morfem. Morfem adalah unit terkecil dalam sebuah teks yang memiliki arti. Sebagai contoh, kata “menerjemahkan” dapat dianalisis secara morfologis menjadi tiga morfem terpisah, yaitu awalan “me-“, kata dasar “terjemah”, dan akhiran “-kan”. Aplikasi pengolahan bahasa

alami hanya perlu mengenali arti dari seluruh morfem saja dan aturan pembentukan kata-kata dari morfem-morfem tersebut untuk dapat mengenali arti dari kata-kata yang tersusun dari lebih dari satu morfem, dan tidak perlu mengenali seluruh kata yang ada pada kamus.

### 3. Analisis Leksikal

Pada tingkatan analisis ini, sistem pengolahan bahasa alami menginterpretasikan arti dari kata-kata individu. Sistem pengolahan bahasa alami memecah masukan menjadi unit-unit kecil bernama *token* pada tingkatan analisis ini. *Token* ini dapat berupa kata, angka, atau tanda baca. Proses pemecahan masukan menjadi *token-token* ini disebut juga *tokenization*. Proses *tokenization* pada Bahasa Indonesia atau Bahasa Inggris cukup sederhana karena setiap kata umumnya dipisahkan oleh spasi atau tanda baca saja. Pada beberapa bahasa, seperti Bahasa Mandarin, kata-kata tidak dipisahkan oleh spasi sehingga proses *tokenization* menjadi lebih sulit.

### 4. Analisis Sintaksis

Tingkatan ini berfokus pada penganalisisan kata-kata dalam sebuah kalimat untuk menemukan struktur tata bahasa dari kalimat tersebut. Tingkatan ini membutuhkan sebuah aturan tata bahasa dan sebuah *parser*. Keluaran dari tingkat pemrosesan ini adalah sebuah representasi dari kalimat tersebut yang mengungkapkan hubungan ketergantungan struktural antara kata-kata. Sintaksis menyampaikan makna pada sebagian besar bahasa karena urutan dan ketergantungan kata-kata berkontribusi pada makna suatu kalimat. Misalnya kalimat “polisi menembak penjahat” dan kalimat “penjahat menembak polisi” hanya berbeda dalam hal sintaksis, namun memiliki arti yang cukup berbeda.

### 5. Analisis Semantik

Tingkatan analisis semantik berusaha mencari arti sebuah kata, frasa, atau kalimat yang paling tepat. Jika suatu kata, frasa, atau kalimat memiliki kemungkinan makna lebih dari satu, analisis semantik akan mencari makna yang paling sesuai dari kemungkinan-kemungkinan makna tersebut. Misal kata “rapat” dapat memiliki makna “pertemuan” atau memiliki makna “tidak renggang”, analisis semantik akan menentukan kemungkinan makna dari kata tersebut dengan melihat konteks penggunaan kata tersebut.

## 6. Analisis Wacana

Tingkatan analisis wacana menganalisis unit yang lebih besar dari kalimat dan hubungannya dengan konteks yang sedang dibahas. Tingkatan ini berusaha menemukan keteraturan linguistik dalam sebuah wacana menggunakan kriteria-kriteria tata bahasa, fonologi, dan semantik. Misalnya kata “dia” pada kumpulan kalimat “Danu menabrak mobil Dina. Akibat kecelakaan tersebut dia tewas” dapat berarti si “Danu” atau si “Dina”. Analisis wacana berusaha mencari tahu siapa yang dimaksud dengan “dia” pada kalimat tersebut dengan menganalisis kalimat-kalimat lainnya baik kalimat sesudah maupun sebelumnya.

## 7. Analisis Pragmatik

Tingkatan analisis ini menganalisis bagaimana mencari tahu makna tersembunyi dalam suatu teks meski tidak tertulis secara eksplisit. Tingkatan analisis ini membutuhkan pengetahuan umum yang luas mencakup pemahaman dari maksud, rencana, dan tujuan dari suatu teks. Misalnya kata “akar” dapat memiliki kemungkinan arti bagian dari tumbuhan atau memiliki arti salah satu operator matematika, pengetahuan pragmatis bahwa teks tersebut membahas adalah mengenai tumbuhan atau mengenai matematika dapat digunakan untuk mengetahui arti dari kata “akar” tersebut.

### 2.4.2. Pendekatan Pengolahan Bahasa Alami

Berdasarkan pendekatannya, Pengolahan Bahasa Alami dibagi menjadi *rule-based NLP*, *statistical NLP*, dan *hybrid NLP* (Wolniewicz, 2015). Pada *Rule-Based NLP* sekelompok pakar-pakar menulis aturan-aturan deterministik untuk mengimplementasikan pemetaan-pemetaan dalam komponen-komponen Pengolahan Bahasa Alami tersebut. Sebagian besar sistem Pengolahan Bahasa Alami menggunakan pendekatan *rule-based* untuk *tokenization* karena bekerja sangat baik untuk langkah tersebut dalam proses Pengolahan Bahasa Alami. Aturan-aturan adalah sebuah cara bagus untuk mengenkapsulasi pengetahuan-pengetahuan pakar tertentu. Namun *parsing* sulit untuk dikodekan, sehingga sebagian besar *parser* modern adalah berbasis statistik.

*Statistical NLP* artinya sistem mempelajari pemetaan terhadap komponen-komponen Pengolahan Bahasa Alami sebagai hubungan-hubungan statistik dengan memproses banyak contoh. Contoh dari *statistical NLP* adalah pemfilter *spam* pada



aplikasi *email*. Sebuah pemfilter *spam* melihat pada statistik keseluruhan sebuah kelompok besar dari masukan konsumen, sehingga keterlibatan pakar sangat minimal. Tingkat akurasi dari model statistik meningkat seiring dengan jumlah data yang tersedia untuk pembelajaran dan model statistik ini dapat terus belajar dengan ongkos yang lebih murah dibandingkan dengan sistem *rule-based*. Metode statistik membutuhkan data-data dalam jumlah sangat banyak untuk melatihnnya agar menjadi lebih handal, sehingga metode ini tidak cocok digunakan jika jumlah data sebagai contohnya sangat terbatas.

Saat ini sebagian besar sistem Pengolahan Bahasa Alami menggunakan kombinasi kedua model, atau yang biasa disebut dengan pendekatan *hybrid*. *Hybrid NLP* mencoba menyatukan keunggulan dari kedua metode. Contohnya *tokenizer* lebih cocok menggunakan metode *rule-based*, dan jika tersedia data pembelajaran dalam jumlah yang sangat banyak, metode *statistical NLP* cenderung lebih mudah dan lebih murah untuk digunakan dibandingkan metode *rule-based*.

Pengembangan aplikasi pada penelitian ini menggunakan kedua pendekatan Pengolahan Bahasa Alami, yaitu *statistical NLP* pada proses penerjemahan menggunakan data pembelajaran statistik serta pada proses *speech recognition*, dan *rule-based NLP* pada proses penerjemahan menggunakan data pembelajaran berdasarkan contoh, proses penerjemahan menggunakan data pembelajaran berdasarkan *keyword*, proses penerjemahan menggunakan data pembelajaran berdasarkan *keyword*, proses penerjemahan menggunakan data *default*, serta pada proses *speech recognition*. Pengembangan aplikasi pada penelitian ini menggunakan kedua pendekatan Pengolahan Bahasa Alami karena kedua pendekatan Pengolahan Bahasa Alami tersebut memiliki keunggulan masing-masing. *Rule-based NLP* dapat dengan mudah diprogram untuk melakukan proses Pengolahan Bahasa Alami tanpa perlu memasukkan data *training* dalam jumlah yang sangat banyak, namun sulit dilatih untuk mempelajari masukan baru seperti pada *statistical NLP*. Meskipun *statistical NLP* mudah dilatih untuk mempelajari masukan baru, *statistical NLP* membutuhkan data *training* yang sangat banyak untuk dapat melakukan proses Pengolahan Bahasa Alami dengan hasil yang cukup memuaskan.

### **2.4.3. Aplikasi Pengolahan Bahasa Alami**

Ada banyak aplikasi dari Pengolahan Bahasa Alami, di bawah ini adalah daftar enam aplikasi Pengolahan Bahasa Alami yang umum dan menjadi topik penelitian yang populer:

1. *Machine Translation.*

*Machine Translation* mengkonversi teks dalam satu bahasa menjadi bahasa lainnya. Contoh aplikasi ini adalah layanan penerjemah *online Google Translate*. Pada awal perkembangannya, *Machine Translation* dikembangkan menggunakan metode *rule-based* yang kurang menghasilkan kesuksesan. Bidang ini bertransformasi pada awal tahun 1990-an ketika peneliti-peneliti dari IBM memperoleh teks paralel Bahasa Inggris dan Bahasa Prancis dalam jumlah besar dan menggunakannya untuk mengembangkan sistem *Machine Translation* Bahasa Prancis-Bahasa Inggris menggunakan metode statistik (Brown dkk., 1993). Pada *Machine Translation* dengan metode statistik, sebuah dokumen ditranslasikan berdasarkan distribusi probabilitas  $p(i/e)$  bahwa sebuah *string*  $i$  di bahasa tujuan (misal Bahasa Indonesia) adalah hasil translasi dari *string*  $i$  dalam bahasa sumber (misalnya Bahasa Inggris). Salah satu pendekatan untuk pemodelan distribusi probabilitas  $p(i/e)$  adalah dengan penerapan Teorema Bayes, yaitu bahwa nilai  $p(i/e)$  proporsional dengan nilai  $p(e/i) * p(i)$ , dimana  $p(e/i)$  adalah kebalikan dari  $p(i/e)$ , yaitu probabilitas *string*  $e$  adalah hasil translasi dari *string*  $i$ , dan  $p(i)$  adalah probabilitas kemunculan *string*  $i$  pada teks yang tertulis dalam bahasa tujuan. Pencarian hasil translasi terbaik dilakukan dengan mencari nilai  $i$  yang memaksimalkan nilai  $p(e/i) * p(i)$  (Brown dkk., 1993). Beberapa tahun belakangan ini berkembang pendekatan *Machine Translation* yang cukup menjanjikan dengan penggunaan *deep learning* (Hirschberg dan Manning, 2015).

2. *Speech recognition.*

*Speech recognition* adalah sistem Pengolahan Bahasa Alami yang mengkonversi suara manusia menjadi bentuk teks. Pendekatan *speech recognition* yang paling umum dan yang terbaik saat ini adalah menggunakan pendekatan probabilistik. *Speech recognition* berbeda dengan *voice recognition*. Pada *voice recognition*, mesin berusaha menebak identitas sang pemilik suara, sedangkan pada *speech recognition* mesin berusaha memahami maksud dari ucapan seseorang.

3. *Speech synthesis*

*Speech synthesis* adalah kebalikan dari *speech recognition*. Pada *speech synthesis*, mesin menghasilkan suara dengan makna yang sama dengan masukan teks dari *user*. Dua teknik yang paling sering digunakan pada *speech synthesizer* adalah *formant synthesis* dan *concatenation synthesis*. Pada *concatenation synthesis*, mesin membutuhkan *database* contoh suara manusia untuk menghasilkan *output* suara, sedangkan pada *formant synthesis*

tidak dibutuhkan contoh suara manusia, sehingga teknik *concatenation synthesis* menghasilkan *output* suara yang lebih alamiah.

4. *Question answering.*

Aplikasi Pengolahan Bahasa Alami ini mengekstrak informasi relevan dari kumpulan teks untuk menjawab pertanyaan yang diajukan dalam bahasa alami. Contoh penerapan yang paling terkenal adalah IBM Watson yang mampu mengalahkan juara manusia pada acara kuis Jeopardy!

5. *Classification.*

Aplikasi Pengolahan Bahasa Alami ini mengurutkan dan mengorganisasikan informasi ke kelompok kategori-kategori yang sesuai (Wolniewicz, 2015). Contoh aplikasinya adalah filter *spam* pada aplikasi *email* dan Google News yang mengelompokkan teks ke dalam kategori-kategori yang sesuai seperti *headline* olahraga, skor sepakbola, ulasan film, berita mengenai perang di Suriah, dan lain-lain.

6. *Text Summarization.*

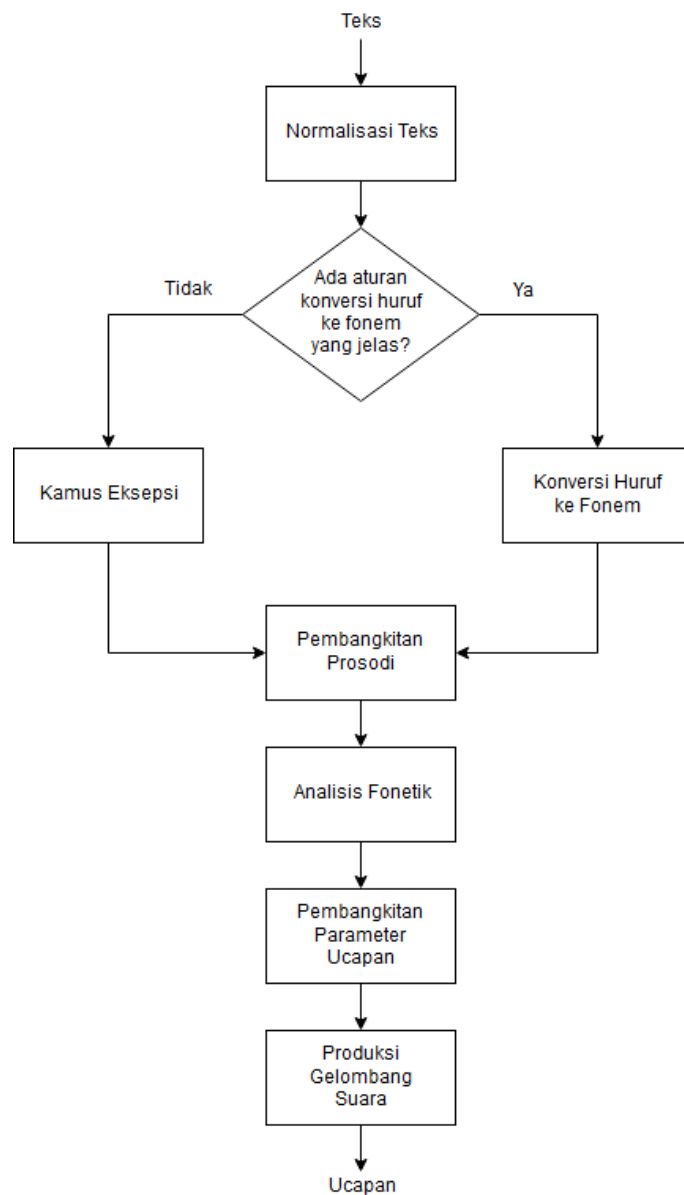
Sistem Pengolahan Bahasa Alami ini berusaha membuat ringkasan dari sebuah teks. Ringkasan dapat didefinisikan sebagai sebuah teks yang diproduksi dari satu atau lebih teks yang menyampaikan informasi-informasi penting dalam teks sumber dan umumnya tidak lebih panjang dari setengah teks sumbernya (Radev dkk., 2002). *Text summarization* biasanya dilakukan pada tingkat analisis wacana.

Pada subbab berikut ini dijelaskan lebih detail mengenai *speech synthesis* dan *speech recognition*, karena kedua aplikasi Pengolahan Bahasa Alami tersebut yang digunakan pada sistem yang dikembangkan pada penelitian ini.

#### **2.4.4. *Speech Synthesis***

Sistem *speech synthesis* terdiri dari dua bagian, yaitu bagian konverter teks ke fonem yang mengubah teks masukan menjadi rangkaian kode-kode bunyi yang direpresentasikan dengan kode fonem, *pitch*, dan durasinya serta bagian konverter fonem ke ucapan yang menerima masukan berupa kode-kode bunyi yang dihasilkan oleh bagian sebelumnya untuk menghasilkan ucapan yang sesuai dengan teks masukan (Arman, 2003).

Ada dua teknik yang banyak digunakan untuk mengimplementasikan konverter fonem ke ucapan, yaitu *formant synthesizer* dan *diphone concatenation*. *Formant synthesizer* melakukan komputasi untuk menghasilkan sinyal ucapan berdasarkan suatu model matematis, sedangkan *diphone concatenation* menghasilkan sinyal ucapan dengan cara menggabung-gabungkan segmen-segmen bunyi berupa *diphone* (gabungan dua buah fonem) yang telah direkam sebelumnya dan tersimpan di dalam *diphone database*. *Diphone concatenation* menghasilkan bunyi yang terdengar lebih alamiah dibandingkan *formant synthesizer*.



Gambar 2.3. Urutan proses konversi dari teks ke ucapan (Arman, 2003)

Tahapan-tahapan utama konversi dari teks menjadi ucapan dapat dilihat pada gambar 2.3. di atas. Tahap normalisasi teks mengubah semua teks masukan menjadi teks yang secara lengkap menunjukkan cara pengucapannya. Tahap berikutnya melakukan konversi dari teks yang dihasilkan pada tahap sebelumnya menjadi kode-kode fonem. Konversi teks menjadi kode-kode fonem dapat dilakukan dengan konversi sederhana dan berlaku umum untuk berbagai kondisi, atau dengan aturan konversi yang bersifat kondisional tergantung dari huruf-huruf atau fonem-fonem tetangganya, bahkan terdapat bentuk-bentuk translasi yang tidak dapat ditemukan keteraturannya. Konversi sederhana dapat diimplementasikan dengan tabel konversi yang berisi pasangan antara urutan huruf dan urutan fonem. Aturan konversi yang bersifat kondisional biasanya diimplementasikan dengan tabel konversi yang diterapkan jika kondisi rangkaian huruf tetangga kiri dan kanannya terpenuhi. Bahasa Indonesia termasuk contoh bahasa yang sebagian besar kata-katanya dapat dikonversikan menjadi fonem dengan aturan yang sederhana. Namun ada kondisi-kondisi yang tidak dapat ditemukan keteraturannya, seperti simbol huruf e yang dapat diucapkan sebagai e pepet atau e taling. Pada diagram gambar 2.3. di atas, konversi yang memiliki aturan yang jelas ditangani oleh bagian Konversi Huruf ke Fonem, sedangkan konversi yang tidak memiliki keteraturan ditangani oleh bagian Kamus Eksepsi. Setelah rangkaian fonem yang merepresentasikan bunyi kalimat telah dihasilkan, bagian Pembangkitan Prosodi melengkapi setiap unit fonem yang dihasilkan dengan data *pitch* serta durasi pengucapannya. Tahap berikutnya adalah tahap Analisis Fonetik yang melakukan perbaikan di tingkat bunyi, misalnya seperti memberi sisipan fonem /y/ dalam pengucapan kata ilmiah di antara fonem /i/ dan /a/.

Pada penelitian ini digunakan sebuah aplikasi *open source* yang menerapkan *speech synthesis* yang bernama eSpeak. Secara *default* eSpeak menggunakan teknik *formant synthesis* dalam menghasilkan suara dan untuk saat ini mampu menghasilkan suara untuk 100 bahasa dan dialek berbeda. Suara yang dihasilkan cukup jernih dan cukup cepat dalam menghasilkan suara, namun tidak menghasilkan suara se-alamiah suara yang dihasilkan aplikasi *speech syntesis* lainnya yang menghasilkan suara dengan menggunakan teknik *diphone concatenation* ("eSpeak text to speech", n.d.).

#### **2.4.5. *Speech recognition***

*Speech recognition* adalah sistem yang mengkonversi suara dari sebuah rekaman sinyal audio menjadi teks. Salah satu metode *Speech recognition* yang paling umum dan

yang terbaik saat ini adalah menggunakan pendekatan probabilistik. Pada pendekatan ini, setiap *input* suara yang dimasukkan dicari hasil transkripsinya dengan mencari kata-kata atau kumpulan-kumpulan pada kamus yang memiliki probabilitas tertinggi merupakan hasil transkripsi dari *input* suara tersebut (Gruhn dkk., 2011). Masalah-masalah yang bisa muncul pada *speech recognition* adalah gangguan kebisingan latar belakang, perbedaan logat *user*, jumlah kosakata sebuah bahasa yang sangat besar, pendeteksian suara-suara yang bukan merupakan kata seperti suara batuk, suara bersin, dan lain lain.

Tingkat akurasi *speech recognition* dapat bervariasi karena beberapa alasan sebagai berikut (Tebelskis, 1995):

1. Ukuran kosakata

Sistem lebih mudah mengenali kata menggunakan kamus dengan jumlah kosakata yang sedikit. Tingkat akurasi berkurang seiring dengan penambahan kosakata pada kamus. Hal ini disebabkan karena dengan bertambahnya kosakata semakin membuat sistem kebingungan memilih kata yang sesuai terutama kata-kata yang memiliki ucapan yang sangat mirip, misalnya kata "tubuh" dengan kata "tuduh", sehingga sistem kadang memberikan *output* kata yang tidak sesuai namun mirip pengucapannya.

2. *Speaker dependence* atau *speaker independence*

Sistem *speaker dependent* berarti sistem *speech recognition* tersebut didesain hanya mampu mengerti ucapan dari orang-orang tertentu saja, biasanya orang-orang yang suaranya menjadi sumber latihan sistem tersebut. Sedangkan sistem *speaker independent* didesain agar mampu mengerti ucapan siapapun dengan tingkat akurasi yang cukup memuaskan. Sistem *speaker dependent* biasanya memiliki tingkat akurasi yang lebih baik dibandingkan sistem *speaker independent* meski jumlah *user* yang dapat menggunakannya hanya terbatas.

3. *Isolated, discontinuous*, atau *continuous speech*

Beberapa sistem *speech recognition* didesain untuk hanya menerima masukan *isolated speech* maksudnya hanya mencoba mengenali satu kata dalam satu waktu saja. Sistem *speech recognition* lainnya didesain untuk hanya menerima masukan *discontinuous speech*, yaitu mampu mengenali beberapa kata dalam satu waktu, namun antar kata yang berbeda harus dipisahkan dengan keheningan selama

beberapa waktu tertentu. Sistem *speech recognition* yang didesain untuk menerima masukan *continuous speech* mampu mengenali rangkaian beberapa kata dalam satu waktu yang diucapkan secara kontinyu tanpa terputus-putus dan sistem sendiri lah yang mencoba menebak batas antar kata. Sistem *speech recognition* yang didesain untuk hanya menerima masukan *isolated speech* dan *discontinuous speech* memberikan tingkat akurasi yang lebih tinggi namun *user* tidak dapat berbicara secara alamiah, sedangkan *user* dapat berbicara secara alamiah tanpa terputus-putus pada *speech recognition* yang didesain untuk menerima masukan *continuous speech*.

4. Batasan bahasa

Maksud batasan bahasa disini adalah sistem hanya didesain untuk mengenali kalimat-kalimat tertentu saja yang sudah dilatihkan kepadanya. Misalnya sistem *speech recognition* untuk menyalakan lampu hanya menerima ucapan "lampu menyala" dan menolak ucapan "dimana dompet saya". Batasan lainnya bisa berupa semantik, misal menolak kalimat "motor saya tertawa" atau bisa juga sintaksis, misalnya menolak kalimat "saya tertawa motor". Batasan direpresentasikan oleh *grammar* yang memfilter kalimat-kalimat yang tidak masuk akal. *Grammar* biasanya dinilai berdasarkan *perplexity*, yaitu angka yang menunjukkan faktor percabangan rata-rata dari *grammar*. Tingkat *perplexity* suatu grammr menentukan tingkat kesulitan suatu sistem dalam mengenali suara.

5. Ucapan spontan atau hasil membaca

Sistem lebih mudah memahami ucapan yang dihasilkan dari membaca ketimbang ucapan yang dihasilkan secara spontan karena ucapan-ucapan seperti "uh", "um", suara batuk, suara tertawa, dll. dapat muncul.

6. Kualitas perekaman

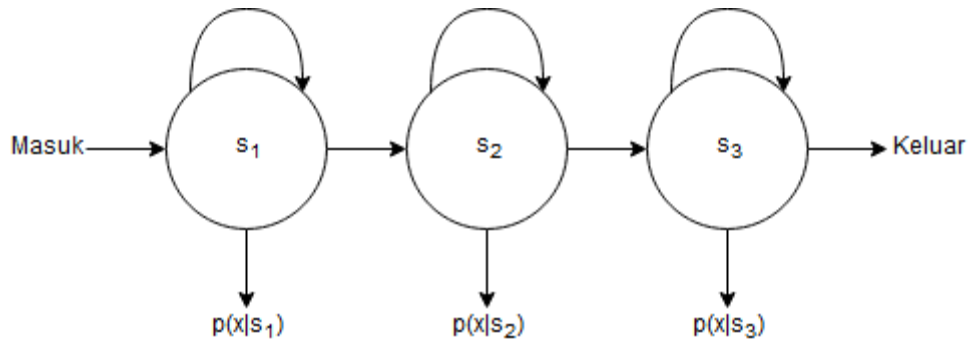
Performa sistem *speech recognition* dapat dipengaruhi oleh kebisingan latar belakang, gaung, gema, jenis mikrofon, dan lain-lain.

Pada penelitian ini digunakan Sphinx4 sebagai fitur *speech recognition* pada aplikasi yang dikembangkan. Sphinx4 bersifat *speaker independent* dan dibangun menggunakan *Gaussian Mixture Models* bersama-sama dengan *Hidden Markov Models*

(GMM-HMM) ("*Basic concepts of speech recognition*", n.d.). Sphinx4 dapat dilatih untuk mengenali berbagai macam bahasa. Tiga model digunakan dalam Sphinx4 untuk melakukan *speech recognition*, yaitu model akustik, kamus fonetik, dan model bahasa ("*Basic concepts of speech recognition*", n.d.). Model akustik berisi sifat-sifat akustik untuk tiap-tiap *senone*. Kamus fonetik berisi pemetaan dari kata-kata ke fon-fon penyusunnya. Model bahasa digunakan untuk membatasi pencarian kata-kata, yaitu dengan menebak kata-kata apa yang dapat muncul setelah kata-kata yang sudah dikenali sebelumnya, sehingga kata-kata yang tidak mungkin muncul diabaikan dalam pencarian. Model bahasa dapat menggunakan pendekatan *grammar* maupun statistik ("*Building a language model*", n.d.). *Grammar* untuk model bahasa dapat dibuat menggunakan Java Speech Grammar Format (JSGF). *Grammar* memungkinkan pengembang untuk menentukan masukan yang sesuai dengan sangat tepat, misalnya kata A hanya bisa ditulis setelah kata B, kata C hanya dapat diulang sebanyak dua kali, dan sebagainya. Namun, dalam percakapan alami sehari-hari seseorang sering tidak berkata sesuai dengan aturan-aturan tersebut, sehingga saat ini pengembang aplikasi lebih suka menggunakan model bahasa dengan pendekatan statistik ketimbang dengan pendekatan *grammar* ("*Building a language model*", n.d.). Model bahasa dengan pendekatan statistik mengandung probabilitas dari kata-kata dan kombinasi-kombinasi kata. Probabilitas-probabilitas tersebut dihitung dari data-data contoh berupa kumpulan *corpus* pada suatu bahasa. Model bahasa menggunakan statistik direkomendasikan untuk masukan yang lebih bebas. Model bahasa memiliki masalah pada pengenalan nama. Untuk mengatasi hal ini, sebuah model bahasa dapat berisi potongan lebih kecil seperti suku kata atau bahkan fon, namun tingkat akurasi *speech recognition* nya menjadi lebih rendah dibandingkan dengan model bahasa yang berbasis kata.

HMM adalah proses stokastik ganda dengan sebuah proses stokastik yang mendasarinya tidak dapat diobservasi langsung dan hanya dapat diobservasi melalui himpunan proses stokastik lainnya yang memproduksi urutan dari simbol-simbol yang diobservasi (Singh dkk., 2012). Proses stokastik atau proses acak adalah sebuah objek matematika yang biasanya didefinisikan sebagai sebuah kumpulan dari variabel-variabel acak. Gambar 2.4. dibawah mengilustrasikan sebuah HMM tiga *state* dimana tiap-tiap *state*  $s_i$  memiliki sebuah kepadatan probabilitas  $p(x|s_i)$  yang menyatakan kepadatan probabilitas untuk observasi akustik  $x$  untuk *state*  $s_i$ . Tiga *state*  $s_1$ ,  $s_2$ , dan  $s_3$  membentuk kata S.





Gambar 2.4. Hidden Markov Model tiga *state*.

*Gaussian Mixture Model* (GMM) adalah fungsi kepadatan probabilitas parametrik yang direpresentasikan sebagai penjumlahan berbobot dari kepadatan komponen *Gaussian* (Vyas, 2013). GMM umumnya digunakan sebagai sebuah model parametrik dari distribusi probabilitas pada pengukuran kontinyu atau fitur-fitur dalam sebuah sistem biometrik, seperti fitur spektral dalam sistem *speech recognition*. Parameter-parameter GMM diperkirakan dari data pelatihan menggunakan algoritma Ekspektasi-Maksimalisasi (EM) iteratif atau estimasi *Maximum A Posteriori* (MAP) dari model sebelumnya yang telah dilatih. GMM yang lengkap diparameterisasi oleh vektor rata-rata, matriks kovariansi dan bobot campuran dari semua kepadatan komponen.

## BAB III

# METODOLOGI PENELITIAN

### 3.1. Pengambilan Data

Untuk mengembangkan aplikasi pada penelitian ini beberapa data yang dibutuhkan adalah sebagai berikut:

1. Data keluhan pasien yang umum beserta istilah medis, lokasi, serta sistem organnya.
2. Data kata-kata yang umum pada teks Bahasa Indonesia dan kata-kata yang umum pada penulisan keluhan pasien beserta sinonimnya.
3. Data model akustik, kamus fonetis, dan model bahasa untuk *speech recognition*.

Data keluhan pasien yang umum beserta istilah medis, lokasi, serta sistem organnya digunakan untuk proses penerjemahan pada aplikasi. Data keluhan pasien yang umum diperoleh dari beberapa situs konsultasi kedokteran *online* seperti *alodokter.com*, *dokter.id*, *klikdokter.com*, dan *tanyadok.com*. Untuk istilah medis, lokasi, dan sistem organ dari keluhan tersebut diperoleh dari membaca-baca buku kedokteran dasar, internet, dan bertanya kepada beberapa dokter dan petugas kesehatan di Rumah Sakit Sardjito Yogyakarta.

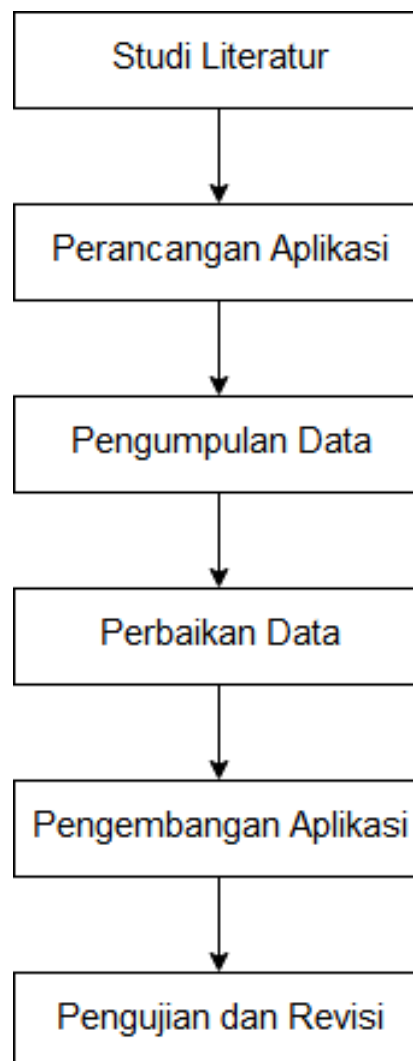
Data kata-kata yang umum pada teks Bahasa Indonesia dan kata-kata yang umum pada penulisan keluhan pasien beserta sinonimnya. Data kata-kata tersebut diperoleh dari penggalan teks pada situs Wikipedia dan beberapa situs konsultasi kedokteran *online* seperti *alodokter.com*, *dokter.id*, *klikdokter.com*, dan *tanyadok.com*. Untuk sinonimnya diperoleh dari kamus tesaurus Bahasa Indonesia *online* maupun cetak.

Data model akustik, kamus fonetis, dan model bahasa digunakan sebagai basis pengetahuan untuk proses *speech recognition*. Data model akustik diperoleh dari merekam contoh suara oleh penulis dan contoh suara yang dihasilkan oleh fitur *speech synthesis* pada Google Translate. Hanya 992 kata dan kumpulan kata umum yang dapat diambil contoh suaranya karena keterbatasan waktu dan biaya untuk penelitian, sehingga kamus fonetis hanya berisi 992 kata dan kumpulan kata saja. Data model bahasa berisi 992 kata dan kumpulan kata pada kamus fonetis tersebut termasuk kombinasi 2-gram, 3-gram,

maupun 4-gram yang umum dari kata-kata dan kumpulan-kumpulan kata tersebut yang diperoleh dari penggalian teks pada *website* health.detik.com beserta data-data untuk eksperimen yang sengaja tetap disimpan pada model bahasa.

### 3.2. Tahapan Penelitian

Gambar 3.1. di bawah menunjukkan alur tahapan yang dilakukan dalam penelitian ini. Berdasarkan gambar 3.1. di bawah, alur tahapan dalam penelitian ini adalah studi literatur, perancangan aplikasi, pengumpulan data, pengolahan data, pengembangan aplikasi, dan diakhiri dengan pengujian dan revisi pada aplikasi. Tiap-tiap tahapan tersebut dijelaskan pada subbab-subbab berikutnya.



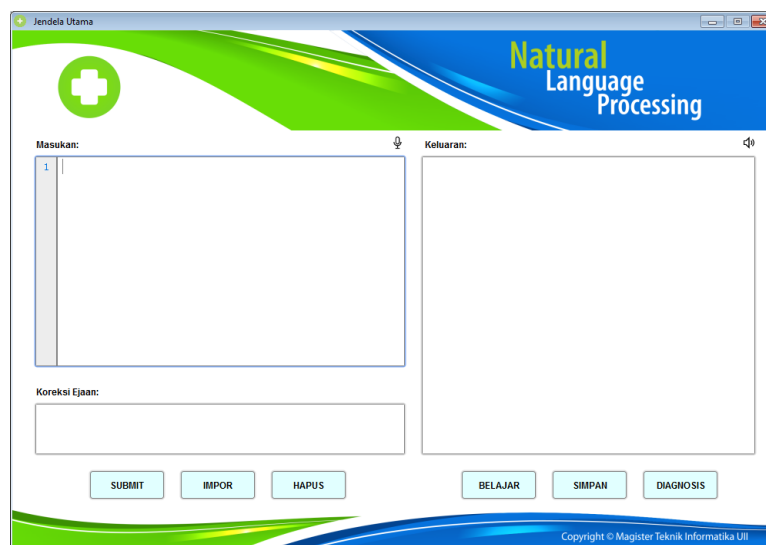
Gambar 3.1. Bagan alur penelitian.

### 3.2.1. Studi Literatur

Pada tahap studi literatur ini dipelajari beberapa literatur baik dari media *online* maupun media cetak mengenai teori-teori dan aplikasi-aplikasi pemrosesan bahasa alami yang sudah pernah dibuat terutama mengenai *speech recognition*, *speech synthesis*, dan *statistical machine translation*. Pada tahap ini dipelajari juga ilmu kedokteran dasar dari beberapa media *online* dan media cetak agar lebih memahami penyakit-penyakit yang umum diderita di masyarakat

### 3.2.2. Perancangan Aplikasi

Setelah dilakukan studi literatur, kemudian dilakukan perancangan aplikasi yang ingin dikembangkan. Pada tahapan perancangan aplikasi, dibuat rancangan antarmuka, rancangan algoritma, dan rencana data-data yang dibutuhkan dalam pengembangan aplikasi. Rancangan tampilan antarmuka utama aplikasi yang dikembangkan dapat dilihat pada gambar 3.2. di bawah ini.



Gambar 3.2. Rancangan tampilan antarmuka utama aplikasi.

### 3.2.3. Pengumpulan Data

Setelah rancangan aplikasi selesai, kemudian dikumpulkan data-data yang dibutuhkan untuk pengembangan aplikasi. Data-data tersebut antara lain adalah data keluhan pasien yang umum yang didapatkan dari situs-situs konsultasi kedokteran *online*,

data kata-kata Bahasa Indonesia yang paling sering digunakan, data model bahasa yang diperoleh dari penggalian teks pada Wikipedia Indonesia dan beberapa situs-situs kedokteran, dan data model akustik yang didapatkan dengan merekam suara penulis sendiri dan dengan menggunakan aplikasi *speech synthesizer* yang ada pada Google Translate.

#### **3.2.4. Perbaikan Data**

Data yang sudah dikumpulkan kemudian diperbaiki terlebih dahulu sebelum digunakan oleh aplikasi. Data-data keluhan pasien yang sama persis dengan data-data keluhan pasien lainnya dihapus. Pada daftar kata-kata yang umum pada teks Bahasa Indonesia dan daftar kata-kata yang umum pada penulisan keluhan pasien, kata-kata yang bukan kata Bahasa Indonesia, seperti kata Bahasa Inggris, diabaikan. Data-data suara untuk model akustik dipilih yang paling baik dan dipotong bagian awal dan akhirnya yang hanya berisi *silence* atau *noise*.

#### **3.2.5. Pengembangan Aplikasi**

Pada tahap ini aplikasi dikembangkan menggunakan Bahasa Java. Bahasa Java dipilih karena Sphinx4 yang digunakan sebagai fitur *speech recognition* pada aplikasi ini dikembangkan menggunakan Bahasa Java. Bahasa Java juga menghasilkan *file executable* yang memiliki performa cukup baik dan aplikasi yang dihasilkannya bersifat *portable*.

#### **3.2.6. Pengujian dan Revisi**

Setelah aplikasi berhasil dikembangkan, aplikasi kemudian diujicoba oleh penulis. Dilakukan pengujian akurasi dan performa penerjemahan pada aplikasi ini dengan memasukkan *input* keluhan sebanyak 30 kalimat secara acak serta pengujian akurasi dan performa fitur *speech recognition* pada aplikasi dengan memasukkan *input* suara sebanyak 30 kali secara acak dengan lingkungan suara sekitar yang berbeda-beda. Jika didapatkan hasil yang kurang memuaskan maka dilakukan revisi baik pada kode program maupun data-data pada program. Setelah proses pengujian oleh penulis dan revisi selesai dilakukan, lima orang dokter masing-masing diminta untuk memasukkan keluhan secara acak untuk 10 penyakit pada aplikasi dan hasil terjemahan oleh aplikasi dicek oleh dokter yang bersangkutan apakah sesuai atau tidak, dari hasil tersebut dihitung tingkat akurasi dan

performa penerjemahan oleh aplikasi. Kelima orang dokter tersebut masing-masing juga diminta untuk mengetes fitur *speech recognition* dengan menggunakan masukan suara mereka sendiri sebanyak 5 kali untuk mengecek tingkat akurasi dan performa fitur *speech recognition* pada aplikasi ini. Setelah itu kelima dokter yang bersangkutan diminta untuk mengisi kuesioner untuk mengetahui penilaian dan saran mereka mengenai aplikasi yang dikembangkan ini. Tampilan rancangan kuesioner yang diberikan kepada dokter-dokter yang mengujicoba aplikasi dapat dilihat pada gambar 3.3. di bawah ini.

### Kuesioner Penilaian Program

Nama Dokter :

Tanggal :

	Sangat Tidak Setuju	Tidak Setuju	Ragu-Ragu	Setuju	Sangat Setuju
Program ini mudah dipelajari dan digunakan					
Langkah-langkah pengoperasian program ini mudah diingat					
Program ini memiliki tampilan antarmuka yang cukup bagus					
Tingkat akurasi pengenalan suara pada program ini sudah cukup baik					
Tingkat akurasi penerjemahan program ini sudah cukup baik					
Kecepatan program ini dalam mengenali suara sudah cukup baik					
Kecepatan program ini dalam menerjemahkan masukan sudah cukup baik					
Program ini dapat membantu pekerjaan petugas kesehatan					
Apakah ada saran bagi pengembangan program?					

Tanda Tangan Dokter

( )

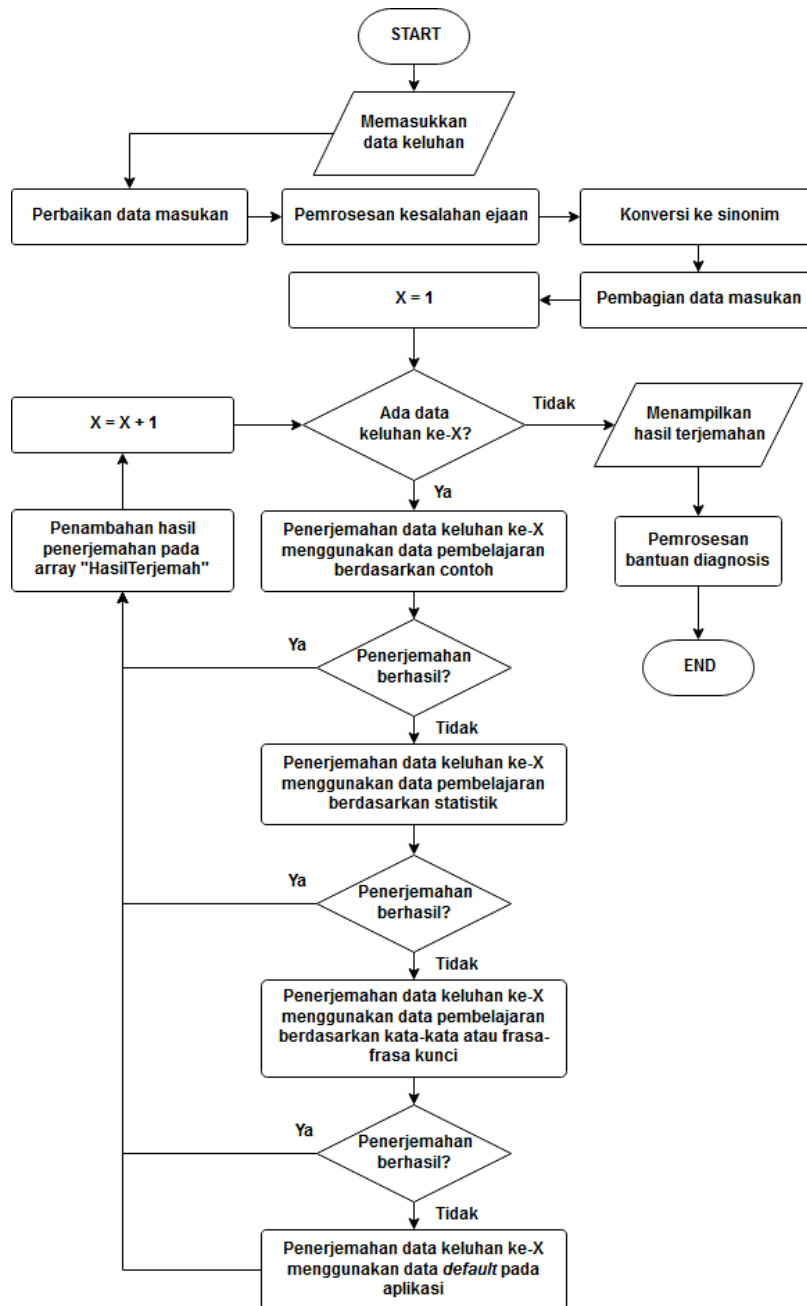
Gambar 3.3. Rancangan kuesioner.

# BAB IV

## PEMODELAN SISTEM

### 4.1. Gambaran Model

Gambaran model keseluruhan proses penerjemahan data masukan keluhan pasien pada aplikasi ini dapat dilihat pada Gambar 4.1. di bawah ini:



Gambar 4.1. Model penerjemahan masukan keluhan pasien.

Berikut ini adalah penjelasan dari Gambar 4.1. di atas:

**a. Masukan Data Keluhan Pasien**

Aplikasi mampu menerima masukan dalam bentuk teks atau suara. Untuk masukan dalam bentuk teks dapat diketikkan secara langsung pada antarmuka aplikasi atau dengan mengimpor *file* teks (.txt) atau *file* excel (.xlsx) yang tersimpan pada media penyimpanan yang berisi data teks keluhan pasien dengan format yang sudah ditentukan. Untuk masukan dalam bentuk suara dapat dimasukkan langsung menggunakan mikrofon atau dengan mengimpor *file* suara dalam format WAV atau format MP3 yang nantinya dikonversi dahulu menggunakan aplikasi *open source* FFmpeg ke format WAV secara otomatis oleh aplikasi.

Fitur *speech recognition* dengan menggunakan Sphinx4 API digunakan untuk mengubah masukan dalam bentuk suara ke bentuk teks. Pada fitur *speech recognition* ini dilakukan analisis tingkat fonologi untuk mengenali fon-fon yang diucapkan oleh *user*. Kata-kata yang diucapkan oleh *user* dibagi menjadi beberapa *frame* kemudian tiap-tiap *frame* tersebut dicocokkan dengan model-model trifon yang ada pada model akustik. Model trifon yang paling sesuai dengan kata-kata yang diucapkan oleh *user* dipilih sebagai hasil transkripsi. Model bahasa pada fitur *speech recognition* kemudian melakukan analisis sintaksis untuk memilih rangkaian kata yang benar atau memiliki probabilitas paling tinggi pada proses transkripsi suara tersebut. Misalnya bigram “tubuh sakit” memiliki probabilitas lebih tinggi terdapat dalam teks Bahasa Indonesia ketimbang bigram “tujuh sakit”, sehingga jika *user* mengucapkan bigram “tubuh sakit” maka aplikasi kemungkinan besar tidak akan keliru mentranskripsikannya menjadi “tujuh sakit”, begitu pula trigram “buang air besar” memiliki probabilitas lebih tinggi terdapat dalam teks Bahasa Indonesia ketimbang trigram “buang akhir besar”, sehingga jika *user* mengucapkan trigram “buang air besar” maka aplikasi kemungkinan besar tidak akan keliru mentranskripsikannya menjadi “buang akhir besar”

**b. Perbaikan Data Masukan**

Data teks keluhan pasien yang sudah dimasukkan kemudian diperbaiki dahulu isinya untuk mempermudah pemrosesan di tahap berikutnya. Perbaikan pada data teks keluhan pasien antara lain perubahan seluruh huruf pada teks ke bentuk huruf kecil (*case folding*), penghapusan kata-kata partikel yang tidak diperlukan dalam proses penerjemahan (misal kata loh, ah, wah, duh, dll.), perbaikan penulisan singkatan dan kata ulang agar



sesuai dengan tata bahasa baku Bahasa Indonesia, perbaikan penulisan tanda baca, penghapusan baris kosong, dan penghapusan spasi-spasi berlebihan termasuk spasi-spasi di awal dan di akhir kalimat (*trimming*). Pada tahap ini dilakukan analisis morfologi untuk mengubah penulisan singkatan dan kata ulang yang tidak baku menjadi lebih baku, misal penulisan kata ulang “paru2” diubah menjadi kata ulang “paru-paru” dan penulisan kata “jari jari” diubah menjadi kata “jari-jari”.

#### **c. Pengoreksian Kesalahan Ejaan/Penulisan**

Setelah isi teks masukan diperbaiki kemudian dilakukan pencarian kesalahan ejaan pada kata-kata yang terdapat di dalam teks masukan tersebut. Hanya kata-kata dengan jumlah karakter lima atau lebih yang akan dicari kesalahan ejaannya. Kesalahan ejaan yang dicari oleh aplikasi antara lain berupa kelebihan satu huruf seperti kata “makan” salah tulis menjadi “makann”, kekurangan satu huruf seperti kata “sakit” salah tulis menjadi “akit”, atau kesalahan menulis sebuah huruf menjadi huruf lainnya seperti kata “pusing” salah tulis menjadi “puting”. Pada proses pengoreksian kesalahan ejaan ini, setiap kata yang terdapat pada teks masukan dicek dahulu apakah ada pada daftar kata-kata yang dikenali oleh aplikasi, jika tidak ada maka aplikasi menganggap bahwa kata tersebut salah tulis dan akan menampilkan rekomendasi kata yang sesuai kepada *user*. Rekomendasi kata didapatkan dengan mencari kata-kata pada *database* aplikasi yang paling mirip dengan kata yang dianggap salah oleh aplikasi tersebut, yaitu mencari kata yang hanya berbeda satu huruf saja dengan kata yang dianggap salah oleh aplikasi tersebut. Jika tidak ditemukan kata yang mirip pada *database* maka aplikasi tidak akan menampilkan rekomendasi kata kepada *user*. Jika ada lebih dari satu kata yang mirip maka dipilih kata yang probabilitas kemunculannya dalam teks Bahasa Indonesia serta dalam teks keluhan pasien yang paling tinggi. Setelah aplikasi menampilkan rekomendasi kata yang sesuai kepada *user*, *user* dapat mengklik rekomendasi kata tersebut untuk mengubah seluruh kata yang dianggap salah tersebut menjadi rekomendasi kata oleh aplikasi.

#### **d. Konversi ke Sinonim**

Proses konversi kata ke sinonimnya dilakukan untuk mencegah dua kata yang memiliki makna yang sama dianggap sebagai kata yang berbeda oleh aplikasi. Misalnya keluhan “sakit telinga” memiliki makna yang sama dengan keluhan “sakit kuping”, sehingga semua kata “kuping” pada teks masukan diubah terlebih dahulu menjadi kata “telinga” oleh aplikasi sebelum dilakukan pemrosesan pada tahap selanjutnya. Proses

konversi kata ke sinonimnya ini juga dilakukan untuk mengubah kata-kata tidak baku menjadi kata-kata yang lebih baku, mengubah kata-kata dalam bahasa daerah menjadi kata-kata dalam Bahasa Indonesia, dan mengubah penulisan bilangan dalam bentuk teks seperti kata “seratus” menjadi penulisan bilangan dalam bentuk angka seperti “100”.

**e. Pembagian Data Masukan**

Sebelum dilakukan proses penerjemahan, data masukan tersebut dibagi-bagi terlebih dahulu menjadi beberapa data masukan untuk pasien berbeda dan dibagi kembali menjadi beberapa data keluhan untuk masing-masing pasien tersebut. Aplikasi mampu menerima data teks masukan untuk beberapa pasien berbeda secara sekaligus. Teks masukan untuk tiap-tiap pasien berbeda dipisahkan oleh baris baru. Keluhan-keluhan berbeda pada setiap pasien dipisahkan oleh tanda titik (.). Tanda koma (,), kata “dan”, dan kata “serta” terkadang juga dapat memisahkan keluhan yang berbeda, dan proses pengecekan apakah tanda koma (,), kata “dan”, dan kata “serta” ini memisahkan keluhan yang berbeda dilakukan pada tahap penerjemahan.

**f. Penerjemahan Data Keluhan**

Proses penerjemahan dilakukan dalam empat tahapan. Berdasarkan urutannya, tahap pertama adalah proses penerjemahan menggunakan data pembelajaran berdasarkan contoh, tahap kedua adalah proses penerjemahan menggunakan data pembelajaran berdasarkan statistik, tahap ketiga adalah proses penerjemahan menggunakan data pembelajaran berdasarkan kata-kata atau frasa-frasa kunci, dan tahap terakhir adalah proses penerjemahan menggunakan data *default* pada aplikasi. Jika *user* tidak melakukan proses pembelajaran pada aplikasi, maka aplikasi akan melakukan proses penerjemahan menggunakan data *default*. Pada proses ini aplikasi akan mencari kata-kata kunci, frasa-frasa kunci, dan lokasi kemunculannya untuk mencari hasil terjemahannya yang sesuai berdasarkan data pada *database*. Pada proses penerjemahan ini dilakukan analisis sintaksis untuk melakukan *parsing* terhadap masukan keluhan supaya diketahui urutan kata-kata yang benar dalam proses penerjemahan. Misalnya kalimat “Menstruasi tidak teratur.” menunjukkan suatu keluhan, namun kalimat “Teratur tidak menstruasi.” tidak menunjukkan suatu keluhan karena urutan kata-katanya tidak sesuai dengan aturan tata bahasa. Analisis sintaksis juga digunakan untuk menerjemahkan keluhan yang mengandung tanda koma (,), kata "dan", atau kata “serta”. Misalnya pada kalimat “kaki sakit, dan kesemutan”, aplikasi memisahkan kalimat tersebut menjadi dua keluhan yaitu

“kaki sakit” dan “kaki kesemutan”, pada keluhan kedua diketahui bahwa organ yang mengalami kesemutan adalah kaki setelah melihat kata-kata sebelum tanda koma. Untuk pencarian waktu awal kemunculan keluhan, frekuensi keluhan, dan sifat keluhan, dilakukan proses pencarian menggunakan ekspresi reguler.

*User* dapat melakukan proses pembelajaran terhadap aplikasi agar tingkat akurasi penerjemahan oleh aplikasi semakin baik. Ada tiga metode pembelajaran yang dapat dilakukan terhadap aplikasi, yaitu pembelajaran untuk penerjemahan berdasarkan contoh, pembelajaran untuk penerjemahan berdasarkan kata-kata atau frasa-frasa kunci, dan pembelajaran untuk penerjemahan berdasarkan statistik. Pada pembelajaran untuk penerjemahan berdasarkan contoh, *user* memasukkan contoh masukan keluhan beserta hasil terjemahannya ke dalam *database* aplikasi, nantinya jika *user* memasukkan keluhan yang sama dengan salah satu contoh masukan yang ada pada *database* maka aplikasi akan menampilkan terjemahan yang sesuai berdasarkan contoh masukan tersebut. Misal jika *user* memberikan contoh masukan “dahi panas” dengan hasil terjemahan X untuk proses pembelajaran berdasarkan contoh, maka jika nanti *user* memasukkan *input* “dahi panas” atau “jidat panas”, aplikasi akan menampilkan hasil terjemahan X.

Pada pembelajaran untuk penerjemahan berdasarkan kata-kata atau frasa-frasa kunci, *user* memasukkan dua atau tiga kata-kata atau frasa-frasa kunci yang harus ada pada sebuah teks masukan keluhan beserta hasil terjemahan dari teks masukan keluhan tersebut ke dalam *database* aplikasi, nantinya jika *user* memasukkan keluhan yang berisi kata-kata atau frasa-frasa kunci yang sama dengan yang ada pada sebuah entri di dalam *database* aplikasi maka aplikasi akan menampilkan terjemahan yang sesuai berdasarkan kata-kata atau frasa-frasa kunci tersebut. Misal jika *user* memberikan kata kunci “berak” dan kata kunci “darah” dengan hasil terjemahan X untuk proses pembelajaran berdasarkan kata-kata atau frasa-frasa kunci, maka jika nanti *user* memasukkan *input* “berak darah” atau “berak mengeluarkan darah”, aplikasi akan menampilkan hasil terjemahan X karena kedua *input* tersebut mengandung kata kunci “berak” dan kata kunci “darah”.

Pada pembelajaran untuk penerjemahan berdasarkan statistik, *user* memasukkan contoh masukan keluhan beserta hasil terjemahannya ke dalam *database* aplikasi, kemudian aplikasi akan mengecek apakah contoh masukan keluhan tersebut mirip dengan salah satu entri yang sudah dimasukkan pada *database* sebelumnya atau tidak, jika tidak ada yang mirip maka aplikasi akan memasukkan contoh masukan keluhan tersebut beserta hasil terjemahannya sebagai entri baru, namun jika sudah ada entri yang mirip maka entri tersebut akan diperbaharui data statistiknya yaitu data probabilitas entri tersebut

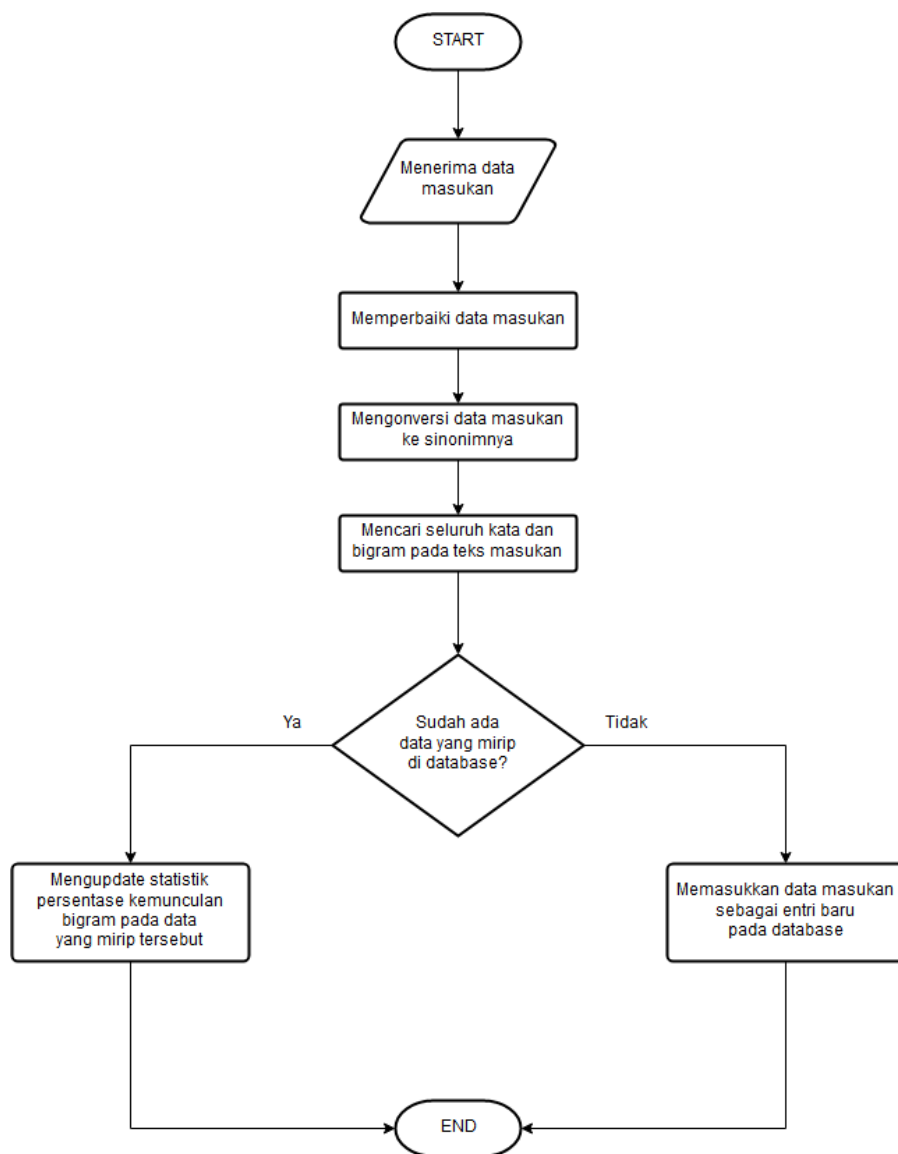
mengandung bigram tertentu. Pengecekan kemiripan adalah berdasarkan jumlah kata, kata-katanya, dan bigram-bigramnya. Dua kata yang memiliki makna yang sama dianggap sebagai kata yang sama. Untuk bisa disebut mirip, contoh masukan keluhan harus berisi jumlah kata dan kata-kata yang sama dengan salah satu entri yang ada pada *database*, serta memiliki jumlah bigram yang sama minimal sebanyak jumlah kata dibagi dua. Nantinya jika *user* memasukkan keluhan maka aplikasi akan mencari apakah keluhan tersebut mirip dengan salah satu entri yang ada pada *database* aplikasi atau tidak. Masukan keluhan dianggap mirip dengan salah satu entri pada *database* jika memiliki jumlah kata yang sama, memiliki kata-kata yang sama, dan memiliki bigram yang sama minimal sebanyak jumlah kata dibagi dua. Jika ada dua atau lebih entri yang mirip, maka dipilih entri yang paling mirip bigram-bigram nya sebagai entri yang paling mirip dengan masukan keluhan, jika masih ada dua atau lebih entri yang mirip maka akan dipilih entri yang pertama ditemukan. Jika ditemukan entri yang mirip maka aplikasi akan menampilkan terjemahan yang sesuai berdasarkan entri tersebut. Dengan menggunakan metode pembelajaran ini, jika *user* memberikan contoh masukan “sakit telinga kemarin” dengan hasil terjemahan X untuk proses pembelajaran berdasarkan statistik, maka jika nanti *user* memasukkan *input* “sakit kuping kemarin” atau “kemarin sakit telinga”, aplikasi akan menampilkan hasil terjemahan X karena kedua *input* tersebut memiliki jumlah kata yang sama, kata-kata yang sama, serta minimal memiliki jumlah bigram sama sebanyak jumlah kata dibagi dua, dengan entri yang sudah dimasukkan ke dalam *database*.

**g. Menampilkan Hasil Penerjemahan**

Setelah proses penerjemahan selesai, hasil terjemahan akan ditampilkan kepada *user* yang isinya antara lain nama umum keluhan, lokasi keluhan, sistem organ keluhan, istilah medis keluhan, awal kemunculan keluhan, frekuensi keluhan, dan sifat keluhan. Sebelumnya teks hasil terjemahan akan diperbaiki dulu agar sesuai dengan tampilan keluaran yang diinginkan. Hasil terjemahan ini juga dapat didiktekan oleh aplikasi menjadi suara yang dapat didengar oleh *user* dengan menggunakan fitur *speech synthesis* pada aplikasi. Teks hasil terjemahan ini dapat disimpan ke dalam file berformat teks atau excel, dan suara yang dihasilkan oleh fitur *speech synthesis* dapat disimpan ke dalam *file* audio berformat WAV. Hasil terjemahan ini nantinya dapat digunakan juga untuk masukan fitur bantuan diagnosis pada aplikasi yang akan menampilkan kemungkinan penyakit yang diderita oleh pasien beserta rekomendasi pengobatannya jika ada. Fitur bantuan diagnosis ini adalah fitur tambahan yang datanya nantinya dapat diisi sendiri oleh dokter.

## 4.2 Algoritma Pembelajaran untuk Penerjemahan

Aplikasi yang dikembangkan dapat mempelajari keluhan baru dengan data pembelajaran yang diberikan oleh *user*. Untuk pembelajaran menggunakan contoh dan *keyword*, algoritmanya pembelajarannya cukup sederhana, yaitu masukan oleh *user* diperbaiki dulu isinya oleh aplikasi, diubah ke sinonimnya, dan kemudian langsung dimasukkan ke dalam *database*. Untuk pembelajaran menggunakan data statistik, algoritma pembelajarannya dapat dilihat pada *flowchart* pada gambar 4.2. di bawah ini:



Gambar 4.2. Algoritma pembelajaran dengan data statistik.

Untuk pembelajaran menggunakan data statistik, pertama-tama *user* memasukkan data masukan lalu kemudian aplikasi memperbaiki data masukan tersebut seperti mengubah semua katanya menjadi huruf kecil semua, menghilangkan spasi ganda, memperbaiki penulisan tanda baca yang kurang tepat, menghapus spasi di awal dan di akhir teks masukan (*trimming*), dan memperbaiki singkatan-singkatan yang tidak baku. Setelah data masukan diperbaiki, isi teks pada data masukan tersebut kemudian dikonversi ke sinonimnya untuk mencegah kata yang sama dianggap berbeda oleh aplikasi. Kemudian aplikasi mencari seluruh kata dan bigram yang ada di dalam teks contoh keluhan pada data masukan.

Aplikasi kemudian mengecek apakah sudah ada data yang mirip di dalam *database*. Pengecekan kemiripan ini dilakukan dengan melakukan pengecekan sebagai berikut:

1. Apakah teks contoh keluhan pada data sebelumnya memiliki jumlah kata yang sama dengan jumlah kata pada teks contoh keluhan pada data yang dimasukkan?
2. Apakah teks contoh keluhan pada data sebelumnya memiliki seluruh kata-kata yang terdapat pada teks contoh keluhan pada data yang dimasukkan tanpa mempedulikan urutannya?
3. Apakah teks contoh keluhan pada data sebelumnya memiliki bigram yang mirip sejumlah setengah dari banyak kata yang terdapat pada teks contoh keluhan pada data yang dimasukkan?
4. Apakah data lokasi, sistem organ, dan istilah medis pada data sebelumnya sama dengan data yang ada pada data masukan.

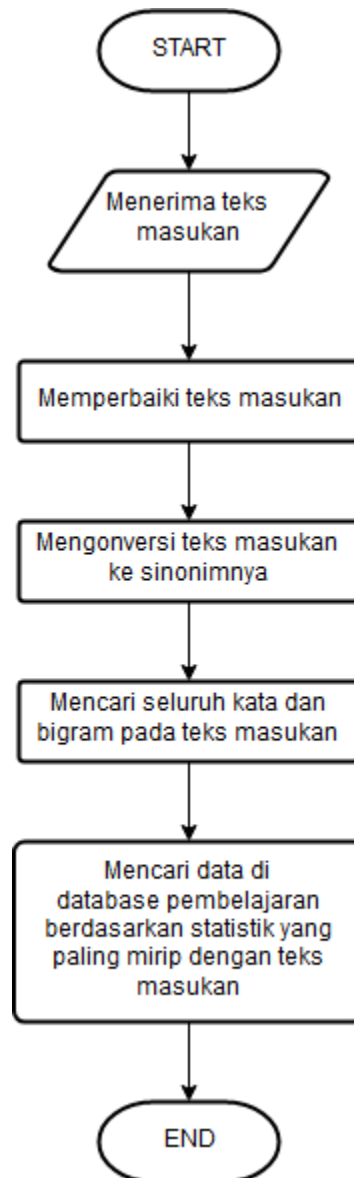
Jika jawaban dari hasil pengecekan keempat pertanyaan tersebut adalah “ya”, maka data masukan digunakan untuk mengubah data statistik probabilitas kemunculan bigram pada data yang mirip yang sudah ada di *database*. Namun jika ada satu saja jawaban tidak dari hasil pengecekan keempat pertanyaan di atas, maka data masukan dimasukkan sebagai entri baru pada *database*.

### 4.3. Algoritma Penerjemahan Menggunakan Pembelajaran Berdasarkan Statistik

Untuk penerjemahan menggunakan hasil pembelajaran berdasarkan statistik, pertama-tama *user* memasukkan *input* ke dalam aplikasi, kemudian aplikasi memperbaiki *input* tersebut yaitu dengan mengubah semua katanya menjadi huruf kecil semua, menghilangkan spasi ganda, memperbaiki penulisan tanda baca yang kurang tepat, menghapus spasi di awal dan di akhir teks masukan (*trimming*), dan memperbaiki singkatan-singkatan yang tidak baku. Teks masukan kemudian diubah ke sinonimnya untuk mencegah aplikasi menganggap kata yang sama artinya sebagai kata berbeda. Kemudian aplikasi mencari data di dalam *database* yang paling mirip dengan teks masukan dengan melakukan pengecekan sebagai berikut:

1. Apakah teks contoh keluhan pada data sebelumnya memiliki jumlah kata yang sama dengan jumlah kata pada teks contoh keluhan pada data yang dimasukkan?
2. Apakah teks contoh keluhan pada data sebelumnya memiliki seluruh kata-kata yang terdapat pada teks contoh keluhan pada data yang dimasukkan tanpa mempedulikan urutannya?
3. Apakah teks contoh keluhan pada data sebelumnya memiliki bigram yang mirip sejumlah setengah dari banyak kata yang terdapat pada teks contoh keluhan pada data yang dimasukkan?

Data yang sesuai adalah data yang memberikan jawaban “ya” pada ketiga pertanyaan di atas. Jika hanya ada satu data yang sesuai pada data di *database*, maka data tersebut yang digunakan untuk penerjemahan. Jika ada lebih dari satu data yang sesuai, pilih data yang memiliki probabilitas bigram paling mendekati data masukan. Jika tidak ada data yang sesuai maka proses penerjemahan gagal dilakukan. Algoritma penerjemahan menggunakan data pembelajaran statistik dapat dilihat *flowchart* nya pada gambar 4.3. di bawah ini:

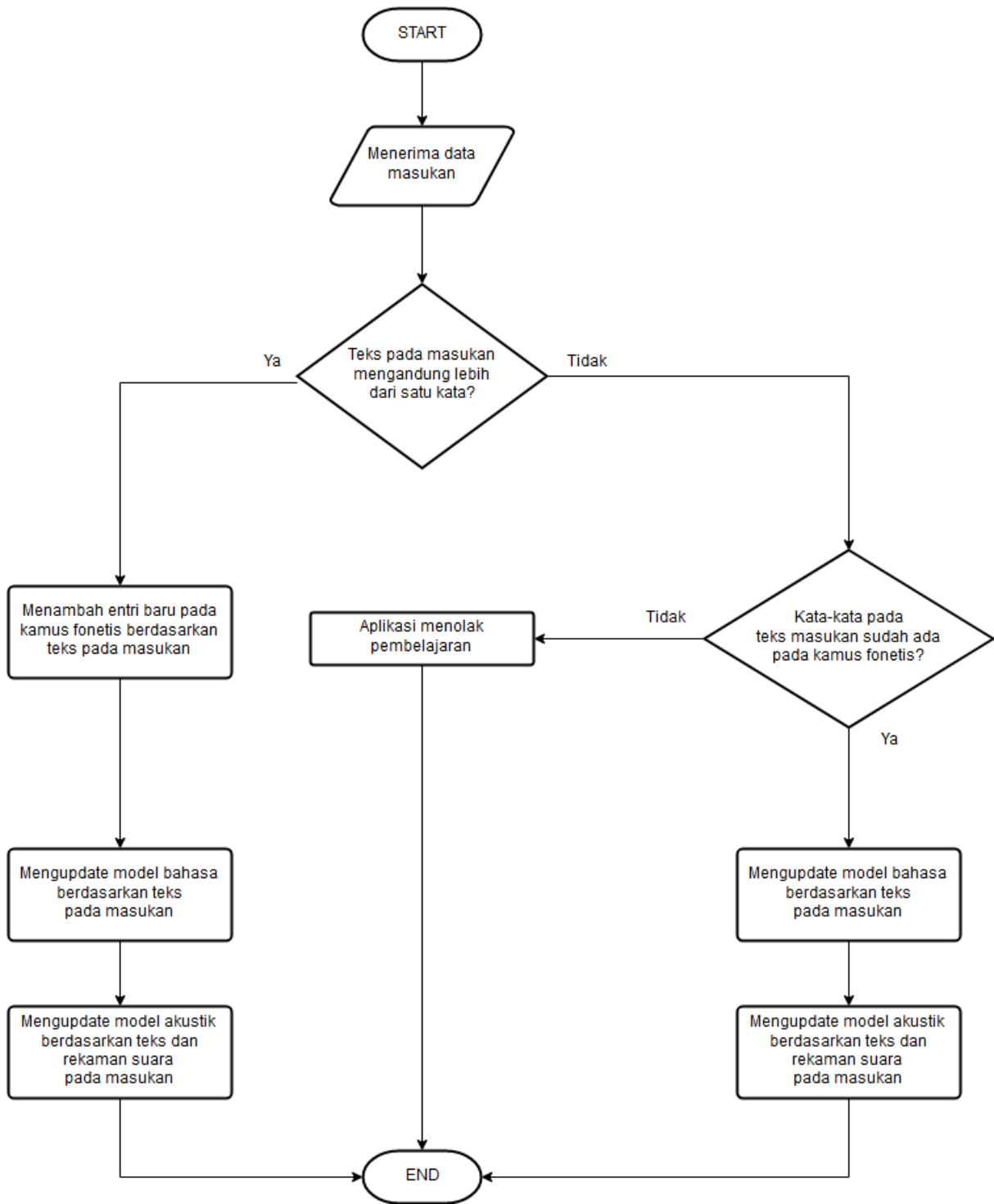


Gambar 4.3. Algoritma penerjemahan menggunakan hasil pembelajaran berdasarkan statistik.

#### 4.4 Algoritma Pembelajaran untuk *Speech recognition*

Pembelajaran aplikasi untuk *speech recognition* dilakukan dengan memasukkan *input* contoh suara dan teks transkripsi yang sesuai dari contoh suara tersebut. *Flowchart* algoritma pembelajaran untuk *speech recognition* dapat dilihat pada gambar 4.4. di bawah ini.





Gambar 4.4. Algoritma pembelajaran untuk *speech recognition*.

Penjelasan dari *flowchart* di atas adalah sebagai berikut. Setelah *user* memasukkan *input* teks dan suara, aplikasi memeriksa apakah *input* teks tersebut hanya berisi satu kata atau lebih. Jika hanya berisi satu kata, maka data teks masukan digunakan untuk meng-

*update* isi kamus fonetis dan model bahasa, sedangkan data masukan suara dan teks digunakan untuk meng-*update* model akustik. Jika berisi lebih dari satu kata dicek lagi apakah kata-kata tersebut sudah ada di dalam kamus fonetis atau belum, jika belum maka aplikasi menolak proses pembelajaran, jika sudah maka data masukan teks digunakan untuk meng-*update* isi model bahasa dan data masukan teks dan suara digunakan untuk meng-*update* model akustik.

# BAB V

## HASIL DAN PEMBAHASAN

### 5.1. Tampilan Antarmuka

Desain antarmuka aplikasi dirancang sederhana agar *user* mudah dalam mempelajari dan menggunakannya. Gambar 5.1. di bawah ini adalah tampilan antarmuka jendela utama pada aplikasi yang sudah dikembangkan:

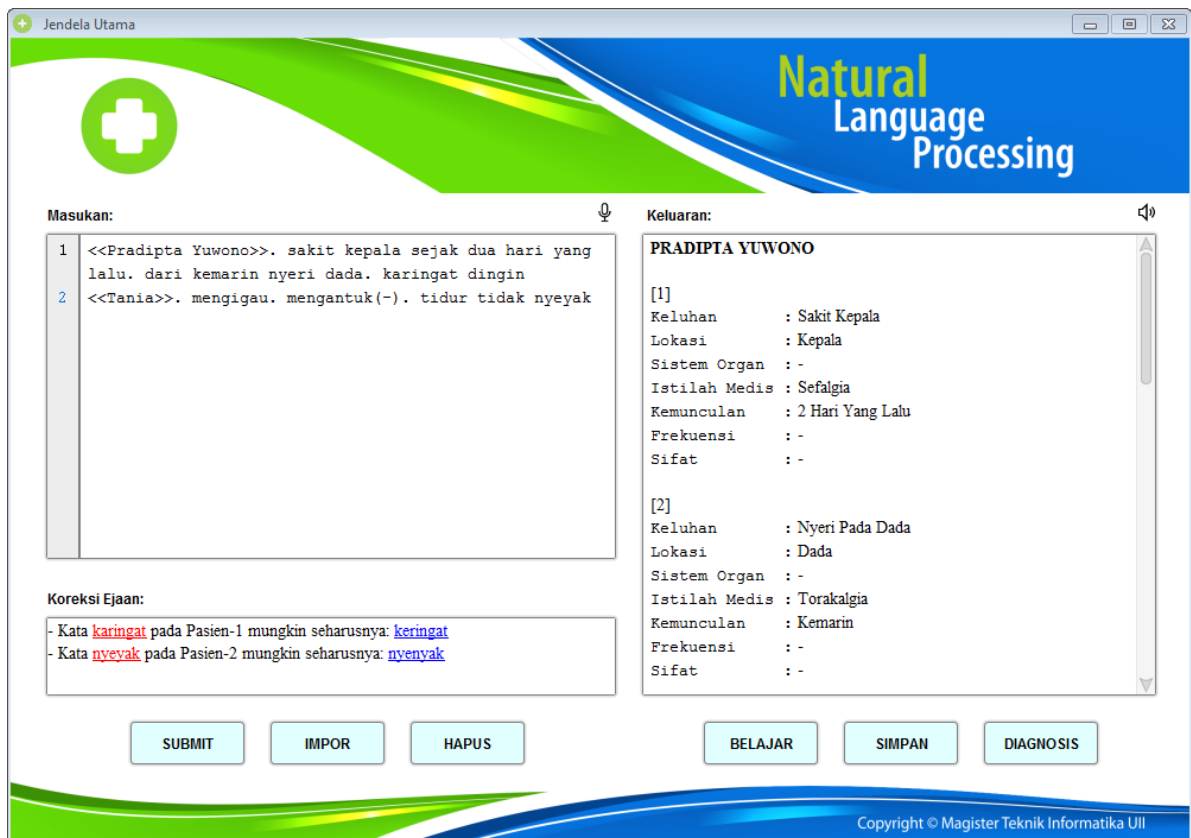


Gambar 5.1. Tampilan antarmuka jendela utama aplikasi.

Pada sebelah kiri atas jendela utama aplikasi terdapat *Text Area* Masukan tempat *user* mengetikkan keluhan secara langsung pada aplikasi. Di sebelah kanan atas *Text Area* Masukan terdapat tombol bergambar mikrofon yang dapat digunakan oleh *user* untuk memasukkan *input* suara ke dalam aplikasi yang nanti diubah menjadi teks pada *Text Area* Masukan menggunakan fitur *speech recognition*. Di sebelah bawah *Text Area* Masukan ada *Text Area* Koreksi Ejaan yang menampilkan hasil koreksi ejaan jika ada penulisan kata

yang dianggap salah oleh aplikasi pada *Text Area* Masukan. Di sebelah kanan atas jendela utama aplikasi terdapat *Text Area* Keluaran tempat hasil terjemahan dituliskan oleh aplikasi untuk ditampilkan kepada *user*. Di sebelah kanan atas *Text Area* Keluaran terdapat tombol bergambar *speaker* yang dapat digunakan oleh *user* untuk mendiktekan hasil terjemahan menggunakan fitur *speech synthesis*. Fitur ini bermanfaat bagi *user* yang memiliki keterbatasan penglihatan. Di paling bawah jendela utama aplikasi terdapat enam tombol-tombol yaitu tombol “Submit”, tombol “Impor”, tombol “Hapus”, tombol “Belajar”, tombol “Simpan”, dan tombol “Diagnosis”. Tombol “Submit” digunakan untuk melakukan proses penerjemahan berdasarkan masukan teks pada *Text Area* Masukan dan menampilkan hasil terjemahannya pada *Text Area* Keluaran, dan jika aplikasi menganggap ada kesalahan penulisan kata-kata pada teks masukan maka aplikasi akan menampilkan hasil koreksinya pada *Text Area* Koreksi Ejaan. Tombol “Impor” digunakan untuk mengimpor isi *file* teks atau *file* excel untuk dituliskan pada *Text Area* Masukan dan untuk mengimpor *file* audio berformat WAV atau MP3 untuk ditranskripsikan menjadi teks pada *Text Area* Masukan menggunakan fitur *speech recognition*. Tombol “Hapus” digunakan untuk menghapus seluruh isi *Text Area* Masukan, *Text Area* Keluaran, dan *Text Area* Koreksi Ejaan. Tombol “Belajar” digunakan untuk melakukan pembelajaran untuk penerjemahan berdasarkan contoh, penerjemahan berdasarkan kata-kata atau frasa-frasa kunci, dan penerjemahan berdasarkan statistik, serta untuk melakukan pelatihan pada fitur *speech recognition* pada aplikasi. Tombol “Simpan” digunakan untuk menyimpan hasil terjemahan pada *Text Area* Keluaran ke dalam *file* berformat teks atau excel, serta untuk menyimpan suara pendiktean dari hasil terjemahan pada *Text Area* Keluaran dalam format WAV. Tombol “Diagnosis” digunakan untuk membuka jendela bantuan diagnosis yang akan menampilkan kemungkinan penyakit-penyakit yang dialami pasien dan rekomendasi pengobatan dari penyakit tersebut menurut data yang ada pada *database*, sesuai keluhan-keluhan yang dialami oleh pasien.

Contoh tampilan antarmuka jendela utama aplikasi setelah *user* memasukkan teks pada *Text Area* Masukan dan menekan tombol "Submit", ditunjukkan pada Gambar 5.2. berikut ini:

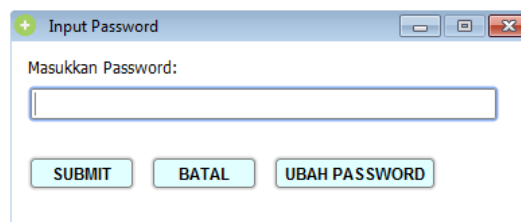


Gambar 5.2. Tampilan antarmuka jendela utama aplikasi setelah *user* men-*submit* teks masukan.

Pada *Text Area* Masukan, *user* dapat mengetikkan keluhan-keluhan untuk satu pasien atau lebih. Keluhan-keluhan untuk pasien-pasien yang berbeda dipisahkan oleh baris baru. Tiap keluhan-keluhan berbeda yang dialami oleh pasien dipisahkan oleh tanda titik (.), tanda koma (,), kata “dan”, atau kata “serta”. *User* dapat mengetikkan nama pasien sebelum daftar keluhan yang dialami pasien dengan ditulis di antara tanda kurung lancip ganda, nantinya nama pasien tersebut ditampilkan pada hasil terjemahan. Isi hasil terjemahan pada *Text Area* Keluaran akan diawali nama pasien jika *user* memasukkan nama pasien pada teks masukan atau tulisan Pasien-X dengan X adalah urutan pasien pada teks masukan jika *user* tidak memasukkan nama pasien pada teks masukan, dan di bawahnya berisi hasil penerjemahan tiap-tiap keluhan yang diawali dengan nomor urutan keluhan dan dilanjutkan nama keluhan, lokasi keluhan, sistem organ keluhan, istilah medis keluhan, awal kemunculan keluhan, frekuensi keluhan, dan sifat keluhan. Jika aplikasi tidak mengenali suatu keluhan, maka akan ditulis “Tidak Ada Di Dalam *Database*” pada hasil terjemahan. *Text Area* Koreksi Ejaan menampilkan rekomendasi penulisan kata yang benar

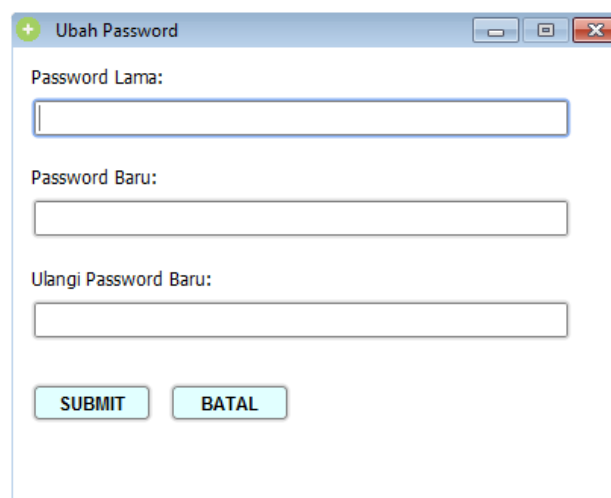
jika aplikasi menganggap *user* salah menuliskan kata tertentu pada *Text Area* Masukan, dimana jika rekomendasi penulisan kata tersebut diklik maka seluruh kata yang dianggap salah penulisannya pada *Text Area* Masukan akan secara otomatis diperbaiki menjadi sesuai dengan rekomendasi penulisan kata oleh aplikasi.

Ketika *user* mengklik tombol “Belajar” atau tombol “Diagnosis”, maka *user* pertama-tama diminta untuk memasukkan *password* dengan tampilan seperti gambar 5.3. di bawah ini. Masukkan *password* pada *Text Area* kemudian klik tombol “Submit” untuk mengecek apakah *password* yang kita tuliskan sudah benar atau belum.



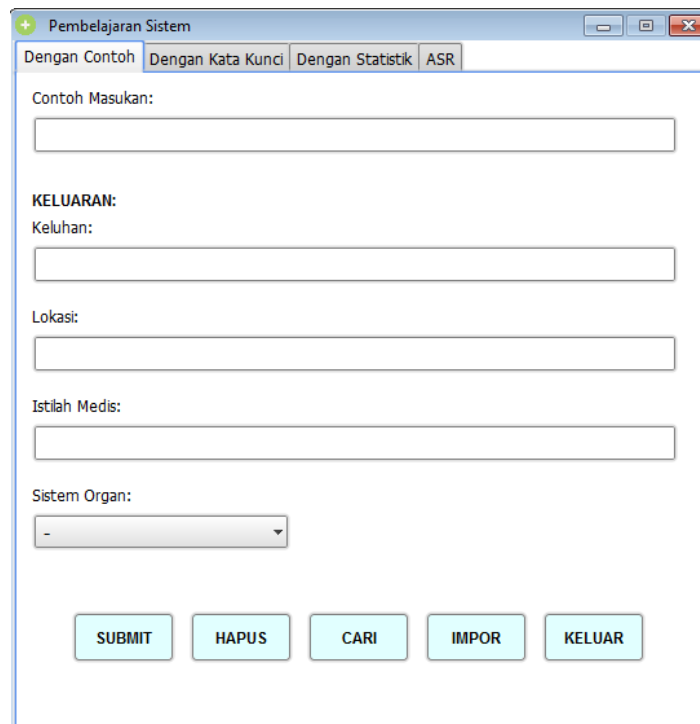
Gambar 5.3. Tampilan antarmuka untuk memasukkan *password*.

*User* dapat mengubah *password* dengan mengklik tombol “Ubah *Password*”. Tampilan jendela perubahan *password* dapat dilihat pada gambar 5.4. di bawah ini. Untuk mengubah *password*, *user* harus memasukkan *password* lama, memasukkan *password* baru, dan mengulangi memasukkan *password* baru untuk konfirmasi bahwa *password* baru yang dimasukkan tidak salah tulis, kemudian menekan tombol “Submit”.



Gambar 5.4. Tampilan antarmuka untuk mengubah *password*.

Gambar 5.5. di bawah ini menunjukkan jendela untuk mengajari sistem penerjemahan berdasarkan contoh, yang muncul setelah *user* mengklik tombol “Belajar” pada jendela utama dan memasukkan *password* yang benar. *User* tuliskan contoh masukan keluhan, nama keluhan, lokasi keluhan, dan istilah medis keluhan pada *Text Area* yang sesuai, serta pilih sistem organ keluhan yang sesuai pada *combo box* yang ada di bawah, kemudian mengklik tombol “Submit” untuk melakukan proses pembelajaran. Untuk menghapus data di dalam *database*, ketikkan contoh masukan keluhan yang ingin dihapus dari *database* kemudian klik tombol “Hapus”. Selain dengan mengetikkan langsung pada antarmuka, *user* juga dapat mengimpor data pembelajaran berformat teks atau excel dengan mengklik tombol “Impor” dan memilih *file* yang sesuai.



Gambar 5.5. Tampilan antarmuka untuk pembelajaran berdasarkan contoh.

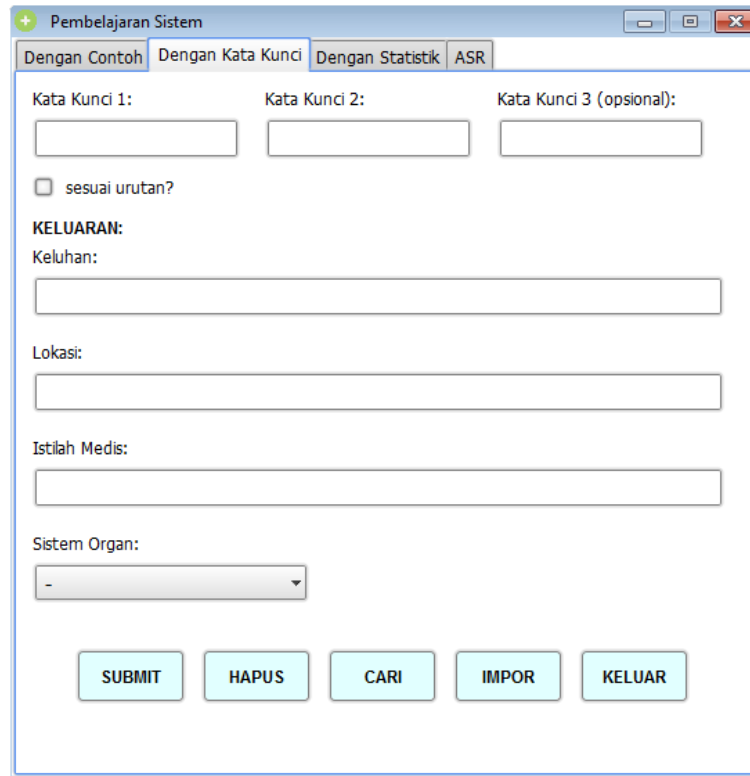
Gambar 5.6. di bawah ini menampilkan jendela yang muncul ketika *user* mengklik tombol “Cari” pada jendela pembelajaran aplikasi berdasarkan contoh. *User* masukkan contoh keluhan yang tersimpan dalam *database* pada *Text Area*, kemudian klik tombol “cari”, untuk menampilkan contoh masukan keluhan, nama keluhan, lokasi keluhan, dan istilah medis keluhan pada *Text Area* yang sesuai, serta sistem organ keluhan yang sesuai pada *combo box* di jendela pembelajaran aplikasi berdasarkan contoh.



Gambar 5.6. Tampilan antarmuka untuk pencarian data hasil pembelajaran berdasarkan contoh.

*User* dapat mengganti mode pembelajaran aplikasi dengan mengklik *tab* yang ada di atas. Gambar 5.7. di bawah ini menunjukkan jendela untuk mengajari sistem penerjemahan berdasarkan kata/frasa kunci. *User* tuliskan 2 atau 3 kata atau frasa kunci dimana kata atau frasa kunci ketiga bersifat opsional, nama keluhan, lokasi keluhan, dan istilah medis keluhan pada *Text Area* yang sesuai, pilih sistem organ keluhan yang sesuai pada *combo box* yang ada di bawah, jika *user* ingin kata atau frasa kunci tersebut sesuai urutan pada teks maka *user* bisa mengklik *checkbox* yang bertuliskan label “sesuai urutan?” di kanannya, kemudian *user* mengklik tombol “Submit” untuk melakukan proses pembelajaran. Untuk menghapus data di dalam *database*, ketikkan contoh masukan keluhan yang ingin dihapus dari *database* kemudian klik tombol “Hapus”. Selain dengan mengetikkan langsung pada antarmuka, *user* juga dapat mengimpor data pembelajaran berformat teks atau excel dengan mengklik tombol “Impor” dan memilih *file* yang sesuai.





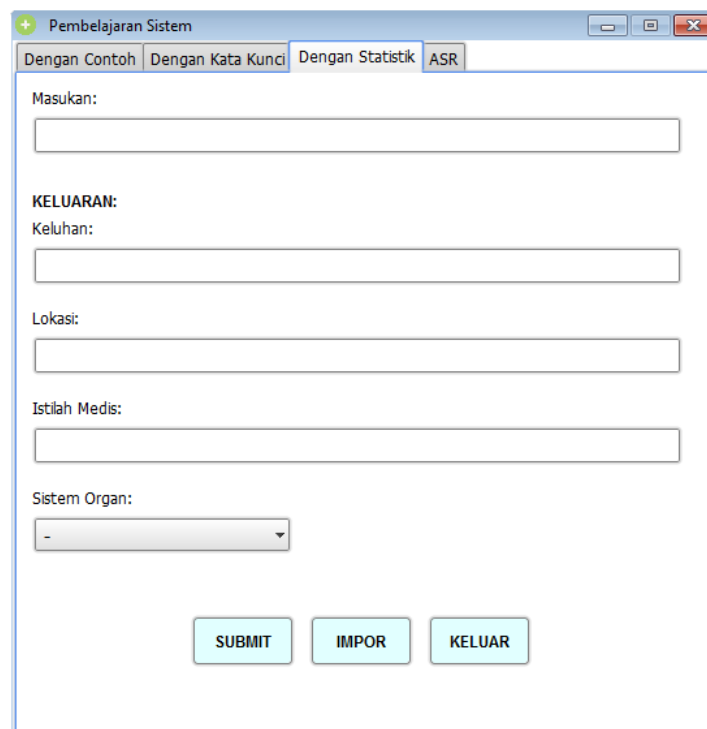
Gambar 5.7. Tampilan antarmuka untuk pembelajaran berdasarkan kata atau frasa kunci.

Gambar 5.8. di bawah ini menampilkan jendela yang muncul ketika *user* mengklik tombol “Cari” pada jendela pembelajaran aplikasi berdasarkan kata/frasa kunci. *User* masukkan kata atau frasa kunci yang tersimpan dalam *database* pada *Text Area*, klik tombol “cari”, dan nantinya aplikasi menampilkan kata atau frasa kunci, nama keluhan, lokasi keluhan, dan istilah medis keluhan pada *Text Area* yang sesuai, sistem organ keluhan yang sesuai pada *combo box*, serta pilihan *checkbox* yang sesuai di jendela pembelajaran aplikasi berdasarkan kata atau frasa kunci.



Gambar 5.8. Tampilan antarmuka untuk pencarian data hasil pembelajaran berdasarkan kata atau frasa kunci.

Gambar 5.9. di bawah ini menunjukkan jendela untuk mengajari sistem penerjemahan berdasarkan data statistik. Sama seperti pembelajaran berdasarkan contoh, *user* tuliskan contoh masukan keluhan, nama keluhan, lokasi keluhan, dan istilah medis keluhan pada *Text Area* yang sesuai, serta pilih sistem organ keluhan yang sesuai pada *combo box* yang ada di bawah, kemudian mengklik tombol “Submit” untuk melakukan proses pembelajaran. Selain dengan mengetikkan langsung pada antarmuka, *user* juga dapat mengimpor data pembelajaran berformat teks atau excel dengan mengklik tombol “Impor” dan memilih *file* yang sesuai.

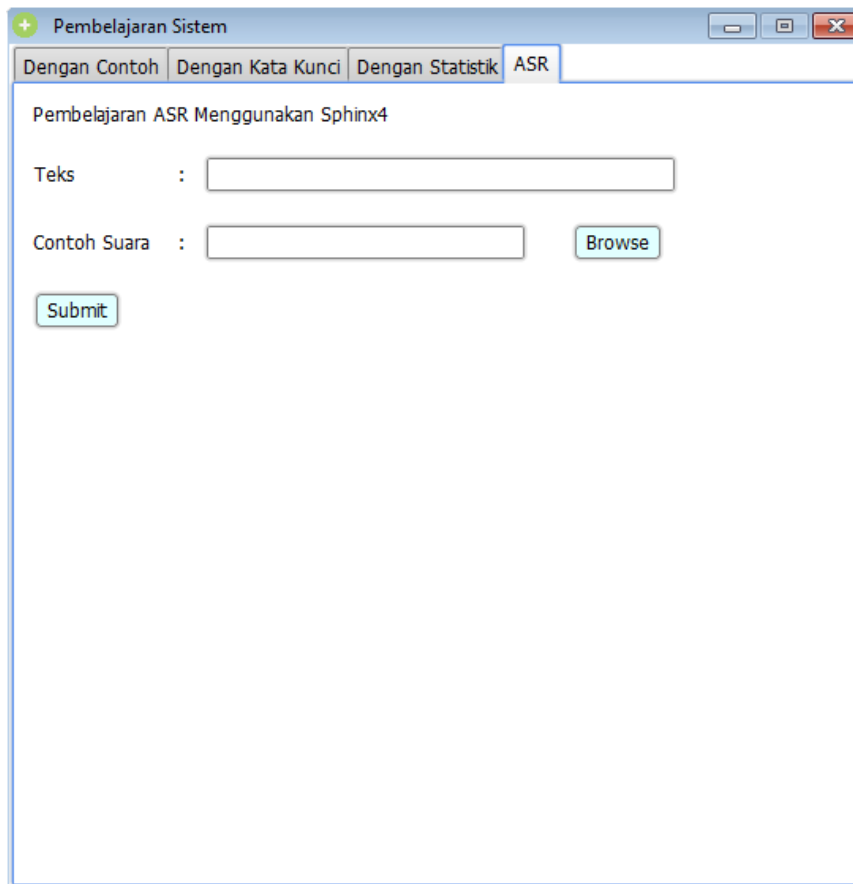


The screenshot shows a window titled "Pembelajaran Sistem" with four tabs: "Dengan Contoh", "Dengan Kata Kunci", "Dengan Statistik" (which is selected), and "ASR". The main content area contains the following elements:

- A text input field labeled "Masukan:".
- A section header "KELUARAN:" followed by a text input field labeled "Keluhan:".
- A text input field labeled "Lokasi:".
- A text input field labeled "Istilah Medis:".
- A dropdown menu labeled "Sistem Organ:" with a downward arrow.
- Three buttons at the bottom: "SUBMIT", "IMPOR", and "KELUAR".

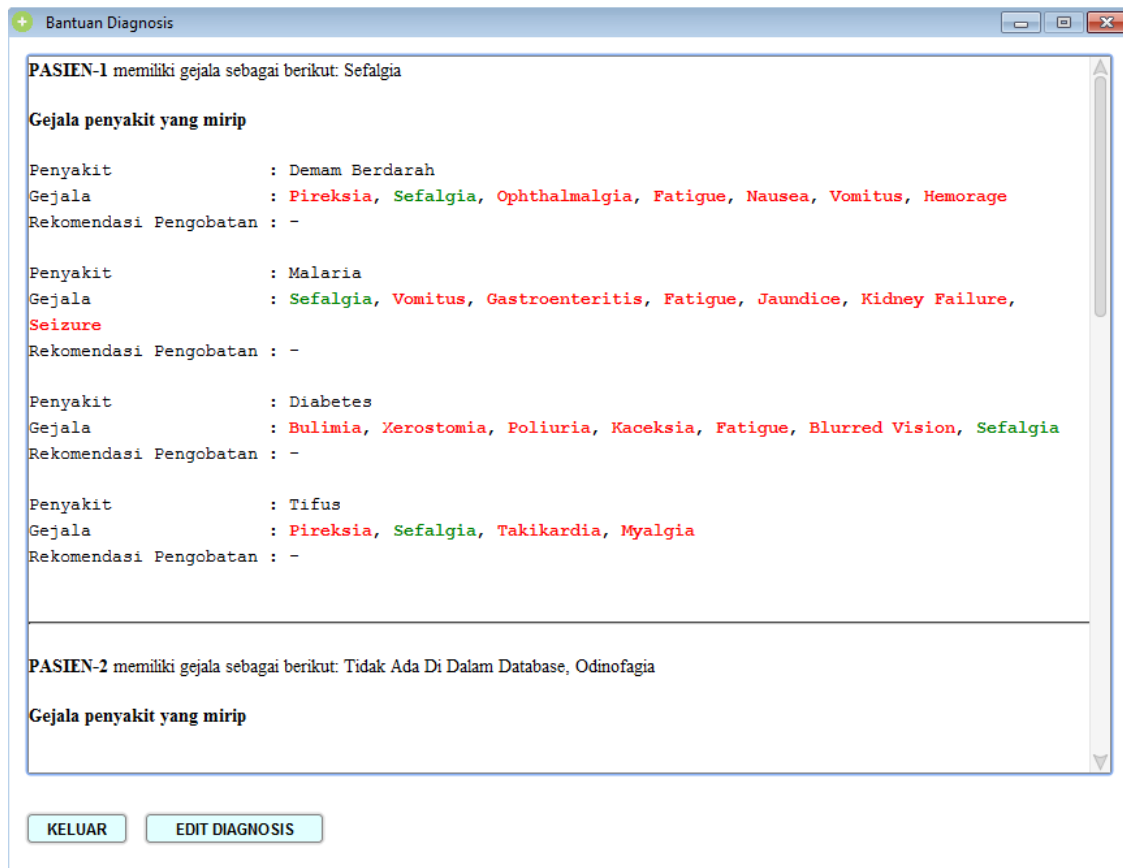
Gambar 5.9. Tampilan antarmuka untuk pembelajaran berdasarkan statistik.

Gambar 5.10. di bawah ini menunjukkan jendela untuk mengajari fitur *speech recognition* pada aplikasi. *User* masukkan contoh rekaman suara dengan memilih *file* nya setelah mengklik tombol “Browse”, kemudian *user* ketikkan transkripsi yang sesuai dari contoh suara tersebut pada *Text Area* di atas tombol “Browse”, dan terakhir *user* mengklik tombol “Submit” untuk melatih fitur *speech recognition* pada aplikasi.



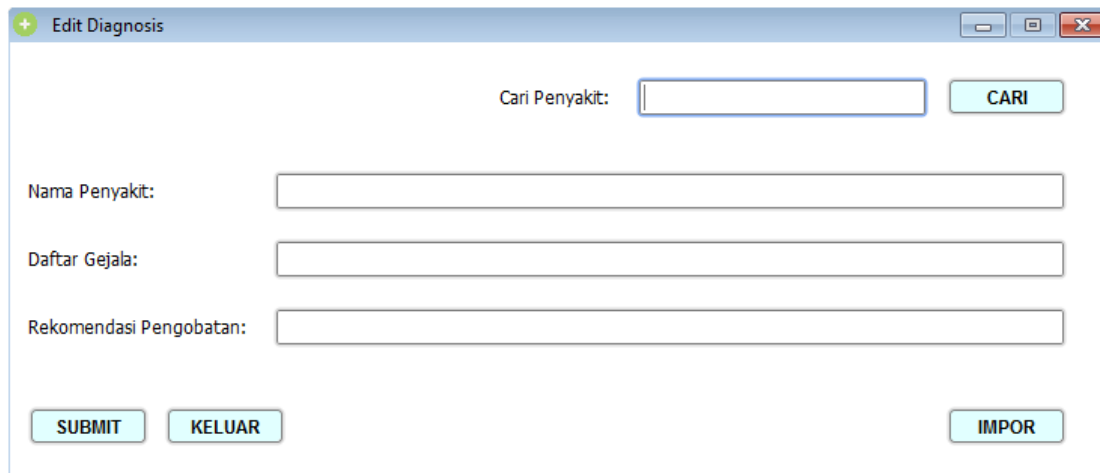
Gambar 5.10. Tampilan antarmuka untuk *training* fitur *speech recognition*.

Gambar 5.11. di bawah ini menunjukkan tampilan jendela bantuan diagnosis yang muncul setelah *user* mengklik tombol “Diagnosis” pada jendela utama dan memasukkan *password* yang sesuai. Gambar 5.11. di bawah menampilkan contoh jika *user* memasukkan keluhan sakit kepala yang dalam istilah medisnya disebut sefalgia pada jendela utama. dan mengklik tombol “Submit” sebelum masuk ke jendela bantuan diagnosis. *User* dapat mengedit isi *database* untuk bantuan diagnosis dengan mengklik tombol “EDIT DIAGNOSIS”.



Gambar 5.11. Tampilan antarmuka bantuan diagnosis.

Gambar 5.12 di bawah ini menampilkan jendela untuk mengedit *database* diagnosis. *User* memasukkan nama penyakit, daftar gejala, dan rekomendasi pengobatan pada *Text Area* yang sesuai, kemudian klik tombol “Submit” untuk menambahkan data diagnosis baru atau mengubah data diagnosis yang sudah ada pada *database*. Pada daftar gejala, tiap gejala yang berbeda dipisahkan oleh tanda titik koma. *User* juga dapat mengimpor data diagnosis yang ingin dimasukkan dari file berformat teks atau excel dengan mengklik tombol “Impor” dan memilih file yang sesuai. Untuk mencari nama penyakit yang sudah tersimpan dalam *database*, *user* bisa memasukkan nama penyakit pada *Text Area* di samping kiri tombol “Cari” kemudian mengklik tombol “Cari”, nantinya jika nama penyakit yang dicari ada dalam *database* maka akan ditampilkan nama penyakitnya, daftar gejalanya, dan rekomendasi pengobatannya pada *Text Area* di bawah.



Gambar 5.12. Tampilan antarmuka untuk mengedit *database* diagnosis.

## 5.2. Hasil Uji Coba

### 5.2.1 Hasil Uji Coba Pengukuran Tingkat Akurasi Fitur *Speech recognition*

Pada penelitian ini dilakukan uji coba tingkat akurasi *speech recognition* menggunakan suara penulis sebanyak 25 kali dan suara 5 orang dokter sebanyak 25 kali. Kelima dokter tersebut adalah dr. Nadia Lakshita Anindya Devi, dr. Setya Rahma Wardhany, dr. Andam Lestari, dr. Ari Eka, dan dr. Novi Kharisma, kelimanya adalah dokter yang bekerja di Klinik Apotek Unisia Polifarma Sleman. Rumus untuk menghitung tingkat akurasi *speech recognition* adalah sebagai berikut (McCowan, 2005):

$$\text{Tingkat Akurasi} = 1 - \frac{(S + D + I)}{N} \quad (5.1)$$

Dengan S adalah jumlah transkripsi kata yang salah (*substitution*), D adalah jumlah kata yang tidak muncul hasil transkripsinya (*deletion*), I adalah jumlah kata hasil transkripsi yang tidak pernah diucapkan oleh *user* (*insertion*), dan N adalah jumlah keseluruhan kata yang diucapkan oleh *user*. Dari hasil pengujian didapatkan bahwa tingkat akurasi *speech recognition* oleh suara penulis adalah sebesar 76% yang didapatkan dari menghitung 100% dikurangi dengan hasil penjumlahan antara *substitution* sebanyak 13 kata dan *deletion* sebanyak 5 kata pada hasil pengenalan suara dan hasilnya dibagi dengan jumlah keseluruhan kata yang diucapkan oleh penulis, yaitu sebanyak 75 kata. Sedangkan

hasil *speech recognition* yang diujicobakan ke 5 orang dokter menunjukkan hasil yang kurang memuaskan karena hampir seluruh suara dokter tersebut gagal atau salah dikenali oleh aplikasi. Hal ini kemungkinan disebabkan karena aplikasi hanya dilatih dengan suara penulis, dan seluruh dokter yang mengujicoba fitur *speech recognition* pada aplikasi adalah perempuan sedangkan penulis sendiri adalah laki-laki, suara perempuan memiliki frekuensi suara yang lebih tinggi dari suara laki-laki. *Speech recognition* dengan Sphinx4 juga hanya dilatih untuk mengenali 992 kata dan frasa saja karena keterbatasan waktu dan biaya penulis dalam pengembangan aplikasi ini. *User* nantinya dapat memberikan pelatihan kalimat-kalimat, kata-kata, atau suara-suara baru pada Sphinx4 jika ingin meningkatkan akurasi *speech recognition* menggunakan Sphinx4 ini. Menurut tim pengembang Sphinx4, dibutuhkan sekitar 50 jam data rekaman suara dari sekitar 200 orang berbeda agar Sphinx4 dapat mengenali suara dari sebagian besar orang yang menggunakannya ("*Training an acoustic model for CMU Sphinx*", n.d.). Karena keterbatasan waktu dan biaya, maka fitur *speech recognition* menggunakan Sphinx4 hanya dapat dilatih dengan data yang terbatas.

### **5.2.2 Hasil Uji Coba Pengukuran Tingkat Akurasi Fitur Pengoreksian Kesalahan Ejaan**

Hasil pengoreksian kesalahan penulisan pada teks masukan setelah diujicoba menggunakan 150 kata berbeda secara acak menunjukkan tingkat akurasi sebesar  $121/150 = 80,67\%$ , yang dihitung dari jumlah kesalahan hasil pengoreksian dibagi dengan jumlah kata yang diujicobakan. Kesalahan pengoreksian umumnya disebabkan karena adanya alternatif rekomendasi kata yang lebih dari satu, misalnya kata "semakan" pada teks masukan kemungkinan adalah kesalahan tulis dari kata "memakan" atau kata "semakin". Ketika *user* bermaksud ingin menulis "memakan", aplikasi justru memberikan rekomendasi kata "semakin" karena kata "semakin" memiliki probabilitas kemunculan yang lebih tinggi pada teks-teks dalam Bahasa Indonesia.

### **5.2.3 Hasil Uji Coba Pengukuran Kecepatan Proses Penerjemahan dan *Speech recognition***

Waktu yang dibutuhkan dalam proses penerjemahan dan *speech recognition* pada aplikasi diukur menggunakan *stop watch*. Komputer yang digunakan pada pengujian aplikasi adalah Laptop ASUS A455L yang memiliki spesifikasi sebagai berikut:

Prosesor : Intel Core i5-4210U (2 Core, frekuensi hingga 2,7 GHz)  
 Memori : DDR3 4 GB  
 GPU : Nvidia GeForce 820 M  
 Sistem Operasi : Microsoft Windows 7 Professional 64-Bit

Kecepatan penerjemahan pada aplikasi diujicoba dengan masukan yang terdiri dari 1 keluhan, 2 keluhan, 3 keluhan, 5 keluhan, dan 10 keluhan, masing-masing sebanyak 5 kali dengan teks keluhan yang acak. Hasil uji coba pengukuran waktu yang dibutuhkan untuk proses penerjemahan dapat dilihat pada tabel 5.1. di bawah:

Tabel 5.1. Hasil Uji Coba Pengukuran Kecepatan Proses Penerjemahan

<b>1 Keluhan</b>	
<b>Daftar Keluhan</b>	<b>Waktu Pemrosesan (detik)</b>
batuk berdahak	5,38
pusing tiap hari	5,37
sakit tenggorokan	5,4
kepala panas	5,45
tubuh pegal-pegal	5,47
<b>Rata-Rata Waktu Pemrosesan 1 Keluhan: 5,41 detik</b>	
<b>2 Keluhan</b>	
<b>Daftar Keluhan</b>	<b>Waktu Pemrosesan (detik)</b>
penglihatan buram. mata berair	5,48
tangan pegal. kaki kesemutan	5,48
muntah-muntah sejak tadi malam. diare parah	5,55
perut kembung tiap hari. berak berwarna hitam	5,53
sejak kemarin muntaber, dan demam parah	5,77
<b>Rata-Rata Waktu Pemrosesan 2 Keluhan: 5,56 detik</b>	
<b>3 Keluhan</b>	
<b>Daftar Keluhan</b>	<b>Waktu Pemrosesan (detik)</b>

kaki, tangan, dan bahu terasa pegal	5,73
sariawan, gusi berdarah, serta lidah terasa pahit.	5,6
kejang-kejang. mengigau. depresi	5,48
sulit mengingat, idiot, dan hilang ingatan	5,54
mimisan. muntah darah. batuk-batuk	5,56
<b>Rata-Rata Waktu Pemrosesan 3 Keluhan: 5,58 detik</b>	
<b>5 Keluhan</b>	
<b>Daftar Keluhan</b>	<b>Waktu Pemrosesan (detik)</b>
perut mules. diare. muntah-muntah. pusing. dan mencret tiap hari	5,46
nyeri perut, nyeri pinggang, nyeri dada, nyeri pinggang, dan nyeri boyok.	5,54
kulit bentol-bentol, berdarah, bernanah, kemerahan, dan muncul bintik putih	5,57
panu. kadas. kurap. biang keringat. kulit berjamur.	5,47
mata minus, penglihatan berbayang, pendengaran berkurang, muncul kotoran telinga, telinga berdenging	5,7
<b>Rata-Rata Waktu Pemrosesan 5 Keluhan: 5,55 detik</b>	
<b>10 Keluhan</b>	
<b>Daftar Keluhan</b>	<b>Waktu Pemrosesan (detik)</b>
nyeri kepala sejak 2 bulan lalu memberat saat istirahat nyeri tenggorokan. nyeri telan diare parah. nyeri perut demam. bintik merah di kulit nyeri boyok kram otot tiap hari batuk	5,81
nyeri tengkuk BAB berdarah Nyeri perut, mual, muntah Batuk berdahak Bersin-bersin setiap pagi, hidung tersumbat Pusing berputar Keluar cairan dari telinga	5,99
sakit pinggang. sakit perut.	6,03



sesak. sakit dada. nyeri kencing dan kencing berdarah batuk dan pilek sakit menelan pusing berputar	
nyeri uluhati. mual. muntah. dada panas. lidah pahit. demam sejak tiga hari. muncul ruam. demam turun hari ketiga. pusing berputar. telinga berdenging.	5,66
muntah. diare nyeri ulu hati BAB berdarah demam lama batuk lama nyeri sendi nyeri boyok lemas mimisan	5,68
<b>Rata-Rata Waktu Pemrosesan 10 Keluhan: 5,83 detik</b>	

Hasil uji coba tersebut menunjukkan bahwa peningkatan jumlah keluhan yang harus diterjemahkan oleh aplikasi tidak meningkatkan secara drastis waktu untuk proses penerjemahannya.

Untuk mengujicoba performa fitur *speech recognition* pada aplikasi, penulis mencoba mengucapkan 1 kata, 2 kata, 3 kata, 4 kata, dan 5 kata secara acak sebagai masukan suara pada aplikasi, masing-masing sebanyak 5 kali. Hasil uji coba pengukuran waktu yang dibutuhkan untuk proses penerjemahan dapat dilihat pada tabel 5.2. di bawah:

Tabel 5.2. Hasil Uji Coba Pengukuran Kecepatan Proses *Speech Recognition*

1 Kata	
Daftar Kata	Waktu Pemrosesan (detik)
demam	15
batuk	20,42
pusing	6,5
diare	6,54
muntah	7,39

<b>Rata-Rata Waktu Pemrosesan 1 Kata: 11,17 detik</b>	
<b>2 Kata</b>	
<b>Daftar Kata</b>	<b>Waktu Pemrosesan (detik)</b>
sakit kepala	10,76
gagal ginjal	15,25
kepala pusing	14,5
batuk batuk	15,34
hidung pilek	23,78
<b>Rata-Rata Waktu Pemrosesan 2 Kata: 15,93 detik</b>	
<b>3 Kata</b>	
<b>Daftar Kata</b>	<b>Waktu Pemrosesan (detik)</b>
pegal tadi pagi	37,91
otot sangat sakit	44,25
sakit ginjal parah	31,51
kaki tidak bertenaga	33,46
mata sulit melihat	46
<b>Rata-Rata Waktu Pemrosesan 3 Kata: 38,63 detik</b>	
<b>4 Kata</b>	
<b>Daftar Kata</b>	<b>Waktu Pemrosesan (detik)</b>
susah buang air besar	13,15
perut mual sejak kemarin	56,53
tadi malam kepala panas	35,35
sejak bulan lalu depresi	46,37
detak jantung sangat tinggi	40,68
<b>Rata-Rata Waktu Pemrosesan 4 Kata: 38,42 detik</b>	
<b>5 Kata</b>	
<b>Daftar Kata</b>	<b>Waktu Pemrosesan (detik)</b>
dada sakit sejak minggu lalu	19,34
batuk dan demam tiap hari	25,71
kepala pusing dan panas parah	16,04
radang tenggorokan dan muntah muntah	31,82
otot tangan tidak bisa digerakkan	36,35

**Rata-Rata Waktu Pemrosesan 5 Kata: 25,85 detik**

Hasil uji coba tersebut menunjukkan bahwa peningkatan jumlah kata hingga 3 kata pada proses *speech recognition* meningkatkan waktu pemrosesan secara drastis, sedangkan untuk pengenalan suara lebih dari 3 kata waktu pemrosesannya tidak berubah drastis. Waktu pemrosesan untuk pengenalan suara pada 5 kata dapat lebih cepat dibandingkan waktu pemrosesan untuk pengenalan suara pada 3 atau 4 kata kemungkinan disebabkan karena data kumpulan 5 kata tersebut sudah tersimpan pada model bahasa.

#### 5.2.4 Hasil Penilaian Kuesioner

Kuesioner diberikan kepada kelima dokter untuk mengetahui penilaian mereka terhadap beberapa aspek pada program. Penilaian dilakukan menggunakan Skala Likert, yaitu dokter diminta memilih antara pilihan Sangat Setuju, Setuju, Ragu-Ragu, Tidak Setuju, dan Sangat Tidak Setuju terhadap pernyataan yang ada pada kuesioner. Adapun skor untuk masing-masing pilihan jawaban adalah:

- 1 = Sangat Tidak Setuju
- 2 = Tidak Setuju
- 3 = Ragu-Ragu
- 4 = Setuju
- 5 = Sangat Setuju

Tabel 5.3. di bawah ini menampilkan hasil penilaian kelima dokter terhadap aspek-aspek pada program berdasarkan kuesioner yang telah diisi.

Tabel 5.3. Hasil Penilaian Kelima Dokter Terhadap Aspek-Aspek pada Program dari Hasil Kuesioner

	<b>dr. Nadia</b>	<b>dr. Setya</b>	<b>dr. Andam</b>	<b>dr. Ari</b>	<b>dr. Novi</b>
Program mudah dipelajari dan digunakan	4	4	4	4	4
Langkah pengoperasian program mudah diingat	4	4	4	4	4
Program memiliki tampilan antarmuka yang cukup bagus	4	4	2	4	4

Tingkat akurasi pengenalan suara pada program sudah cukup baik	3	2	2	2	3
Tingkat akurasi penerjemahan pada program sudah cukup baik	4	3	4	2	4
Kecepatan pengenalan suara pada program sudah cukup baik	3	2	2	2	3
Kecepatan penerjemahan pada program sudah cukup baik	4	3	2	3	4
Program dapat membantu pekerjaan petugas kesehatan	4	4	5	3	4

Berdasarkan tabel 5.3. di atas didapatkan persentase dokter yang setuju terhadap penilaian aspek-aspek pada program berdasarkan kuesioner yang telah diisi, sebagai berikut:

1. Program mudah dipelajari dan digunakan : 100%
2. Langkah pengoperasian program mudah diingat : 100%
3. Program memiliki tampilan antarmuka yang cukup bagus : 80%
4. Tingkat akurasi pengenalan suara pada program sudah cukup baik : 0%
5. Tingkat akurasi penerjemahan pada program sudah cukup baik : 60%
6. Kecepatan pengenalan suara pada program sudah cukup baik : 0%
7. Kecepatan penerjemahan pada program sudah cukup baik : 40%
8. Program dapat membantu pekerjaan petugas kesehatan : 80%

Berdasarkan perhitungan persentase di atas dapat disimpulkan bahwa secara keseluruhan rancangan antarmuka aplikasi sudah baik menurut para dokter karena mudah dipelajari, mudah digunakan, pengoperasiannya mudah diingat, dan tampilannya sudah cukup bagus. Para dokter menganggap fungsi penerjemahan pada aplikasi sudah mampu menghasilkan terjemahan dengan cukup baik dan memiliki kecepatan penerjemahan yang cukup baik, meski dr. Ari menganggap tingkat akurasi penerjemahan pada aplikasi masih kurang baik dan dr. Andam menganggap kecepatan penerjemahan pada aplikasi masih kurang baik. Untuk fungsi pengenalan suara pada aplikasi baik tingkat akurasi maupun kecepataannya masih dianggap kurang baik oleh para dokter dan hanya memiliki skor rata-rata kurang dari 3. Dari hasil penilaian pada kuesioner diketahui bahwa para dokter setuju bahwa program yang dikembangkan ini dapat membantu pekerjaan petugas kesehatan.

Saran-saran dari kelima dokter terhadap aplikasi yang telah dikembangkan pada penelitian ini dapat dilihat pada tabel 5.4. di bawah ini:

Tabel 5.4. Saran-Saran dari Kelima Dokter yang Telah Mengujicoba Aplikasi

<b>Nama Dokter</b>	<b>Saran-Saran</b>
dr. Nadia Lakshita Anindya Devi	Dikembangkan untuk dapat menyimpan rekam medis dalam kurun waktu tertentu sehingga bisa melihat riwayat penyakit pasien di waktu dahulu.
dr. Setya Rahma Wardhany	<i>Database</i> nya bisa ditambah lagi, kriteria diagnosis juga bisa lebih dimaksimalkan. Secara keseluruhan cukup bagus.
dr. Andam Lestari	Perlu dikembangkan lagi.
dr. Ari Erna	Pengenalan suara lebih ditingkatkan, istilah medis banyak yang tidak sesuai.
dr. Novi Kharisma	Perbaiki pengenalan suara.

# BAB VI

## KESIMPULAN DAN SARAN

### 6.1. Kesimpulan

Berdasarkan hasil pengujian aplikasi, kesimpulan yang didapatkan dari penelitian ini adalah:

1. Proses penerjemahan keluhan pasien pada aplikasi yang dikembangkan sudah memberikan hasil yang cukup baik. Kesulitan utama yang dihadapi dalam pengembangan fungsi ini adalah banyaknya kemungkinan penulisan teks keluhan pasien termasuk penggunaan kata-kata maupun tata bahasa yang tidak baku yang ada pada teks masukan. Fitur pembelajaran yang ada pada aplikasi ini dapat meningkatkan tingkat akurasi penerjemahan keluhan pada aplikasi.
2. Pada fitur *speech recognition* menggunakan Sphinx4 terdapat beberapa kekurangan seperti kurang akurat dan lamanya proses pengenalan suara terutama pada kalimat yang panjang, dan sedikitnya jumlah suara yang digunakan untuk pelatihan *speech recognition*. Fitur pembelajaran yang ada pada aplikasi ini dapat meningkatkan tingkat akurasi fitur *speech recognition* ini.
3. Aplikasi sudah mampu melakukan pembelajaran keluhan baru dan pelatihan pengenalan suara baru dengan cukup baik. Kekurangan pada fitur pembelajaran ini adalah aplikasi membutuhkan banyak data *training* dalam proses pelatihan pengenalan suara agar tingkat akurasi *speech recognition* dapat meningkat secara signifikan.
4. Pengoreksian ejaan otomatis pada teks masukan menunjukkan tingkat akurasi hanya 80,67% karena pencarian kata rekomendasi hanya berdasarkan probabilitas kemunculan kata dalam teks Bahasa Indonesia saja.

### 6.2. Saran

Mengingat keterbatasan waktu dan biaya dalam melakukan penelitian ini, maka penulis menyadari bahwa aplikasi yang dikembangkan ini masih belum cukup sempurna. Berikut ini daftar saran-saran yang penulis berikan untuk pengembangan aplikasi ini di masa mendatang:

1. Jumlah kata dan keluhan yang dapat dikenali oleh aplikasi sebaiknya ditambahkan hingga tingkat akurasi aplikasi dalam penerjemahan masukan menjadi cukup baik untuk digunakan oleh sebagian besar petugas kesehatan. Untuk meningkatkan jumlah kata dan keluhan yang dapat dikenali aplikasi dapat dilakukan dengan mengajari aplikasi menggunakan antarmuka yang sudah ada pada aplikasi atau dengan memodifikasi isi *database* atau *source code* aplikasi ini.
2. Aplikasi harus sering dilatih oleh berbagai macam suara manusia untuk kata yang berbeda-beda dan lingkungan yang berbeda-beda agar fitur pengenalan suara menggunakan Sphinx4 API pada aplikasi ini menjadi semakin akurat.
3. Untuk penanganan tanda koma pada masukan keluhan, aplikasi masih belum sempurna dalam menanganinya. Diharapkan dapat disempurnakan pada pengembangan aplikasi yang akan datang
4. Pada pengembangan berikutnya diharapkan aplikasi mampu membaca langsung tulisan yang ditulis di atas kertas menggunakan tangan dengan menggunakan fitur *Optical Character Recognition*.
5. Untuk pengembangan ke depannya diharapkan juga aplikasi dapat menerima masukan-masukan lain seperti riwayat penyakit dahulu, riwayat penyakit keluarga, kebiasaan pasien, kondisi sosial dan ekonomi pasien, hasil pemeriksaan laboratorium, dll.
6. Aplikasi sebaiknya dapat diintegrasikan dengan Sistem Pendukung Keputusan Klinis (SPKK) pada penelitian yang akan datang, dimana hasil keluaran dari aplikasi ini dapat menjadi masukan untuk Sistem Pendukung Keputusan Klinis sehingga dapat membantu pekerjaan petugas kesehatan.

## DAFTAR PUSTAKA

- Abdullah, Alex. H.P., Achmad. (2012). *Linguistik Umum*. Jakarta: Penerbit Erlangga.
- Ajami, S., dan Bagheri-Tadi, T. (2013). Barriers for Adopting Electronic Health Records (EHRs) by Physicians. *Acta Informatica Medica*, 21(2), 129–134.
- Alessandro, M. (2003). *Natural Language Processing and Automated Text Categorization: A Study on the Reciprocal Beneficial Interactions*. Roma: University of Rome.
- Andriani, R., Kusnanto, H., Istiono, W. (2017). Analisis Kesuksesan Implementasi Rekam Medis Elektronik di RS Universitas Gadjah Mada. *Jurnal Sistem Informasi*, 13(2), 90-96
- Arman, A. A. (2003). *Konversi dari Teks ke Ucapan*. Bandung: Departemen Teknik Elektro ITB.
- Averbis Health Discovery - Analysis of patient data*. (2018). *Averbis GmbH*. Diakses 12 Februari 2018, dari <https://averbis.com/en/health-discovery/>.
- Bacchiani, M., Beaufays, F., Schalkwyk, J., Schuster, M., dan Strobe, B. (2008). Deploying GOOG-411: Early Lessons in Data, Measurement, and Testing. *IEEE International Conference*, pp. 5260-5263.
- Basic concepts of speech*. (n.d.). Diakses 30 September 2017, dari <https://cmusphinx.github.io/wiki/tutorialconcepts>.
- Bilimoria, N. M. (2007). Electronic Health Records Implementation: What Hospitals and Physicians Need to Know to Comply with Recent Health Law Requirements. *Bloomberg Corporate Law Journal*, vol. 501, pp. 415–425, 2007.
- Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., Mercer, R. L. (1993). The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, Volume 19, Issue 2.
- Building a language model*. (n.d.). Diakses 30 September 2017, dari <https://cmusphinx.github.io/wiki/tutorialconcepts>.
- Chaer, A. (2003). *Linguistik Umum*. Jakarta: Rineka Cipta.
- Chapman W. W., Dowling J. N., Ivanov O., Gesteland P., Olszewski R., Espino J., dkk. (2004). Evaluating Natural Language Processing Applications Applied to Outbreak and Disease Surveillance. *Proceedings of 36th Symposium on The Interface: Computing Science and Statistics, Baltimore*.
- Chapman, N. P. (1987). *LR Parsing: Theory and Practice*. Cambridge: Cambridge University Press.



- CLAMP Team. *CLAMP | Natural Language Processing (NLP) Software*. (n.d.). Diakses 12 Februari 2018, dari <https://clamp.uth.edu/faq.php>.
- CLiX ENRICH White Paper. (2014). Diakses 12 Februari 2018, dari <http://clinithink.com/wp-content/uploads//2016/12/Clinithink-CLiX-ENRICH-White-Paper.pdf>.
- Copestake, A. (2003). *Natural Language Processing: Part 1 of Lecture Notes*. Cambridge: Ann Copestake Lecture Note Series.
- Dale, R. (2006). *An Introduction to Natural Language Generation*. Australia: Microsoft Institute of Advanced Software Technology.
- Dorland, W. A. (2012). *Dorlands Illustrated Medical Dictionary* (32nd ed.). Philadelphia: Elsevier Saunders.
- eSpeak text to speech*. (n.d.). Diakses 30 September 2017, dari <http://espeak.sourceforge.net>.
- Fakultas Kedokteran Universitas Indonesia. (2009). *Pedoman Rekam Medis Berorientasi Masalah*. Jakarta: Fakultas Kedokteran Universitas Indonesia.
- Ford, E., Carroll, J. A., Smith, H. E., Scott, D., dan Cassell, J. A. (2016). Extracting Information from The Text of Electronic Medical Records to Improve Case Detection: A Systematic Review. *Journal of the American Medical Informatics Association*.
- Friedman C. (2000). A Broad-Coverage Natural Language Processing System. *Proc AMIA Symp 2000*, 270-4.
- Friedman, C., dan Johnson, S. B. (2006). *Natural Language and Text Processing in Biomedicine*. New York: Springer.
- Friedman, C., Hripcsak, G., Shagina, L., Liu, H. (1999). Representing Information in Patient Reports Using Natural Language Processing and Extensible Markup Language. *J Am Med Inform Assoc* 1999; 6: 76-87.
- Gruhn, R. E., Minker, W., dan Nakamura, S. (2011). *Statistical Pronunciation Modeling for Non-Native Speech Processing*. New York: Springer Science & Business Media.
- Grune, D., Jacobs, C. J. H. (1990). *Parsing Techniques - A Practical Guide*. New Jersey: Ellis Horwood.
- Gupta, V., dan Lehal, G. (2009). A Survey of Text Mining Techniques and Applications. *Journal of Emerging Technologies in Web Intelligence*, vol. 1, pp. 60-76.

- Hinton, G., dkk. (2012). Deep Neural Networks for Acoustic Modeling in Speech Recognition. *Signal Processing Magazine*.
- Hirschberg, J., dan Manning, C. D. (2015). Advances in Natural Language Processing. *Science*, 349(6245):261–266.
- Iman, K., dan Mohammad, S. A. (2014). A Metric-Based Approach for Web-Based Question Answering. *International Journal of Information Technology and Computer Science*, vol. 9, pp, 39-45.
- Iroju, O. G., Olaleke, J. O. (2015). A Systematic Review of Natural Language Processing in Healthcare. *I.J. Information Technology and Computer Science*, 2015, 08, 44-50.
- Jurafsky, D., dan Martin, J. H. (2008). *Speech and Language Processing*. Englewood Cliffs: Prentice-Hall.
- Kementerian Kesehatan Republik Indonesia. (2008). *Peraturan Menteri Kesehatan Republik Indonesia Nomor 269/Menkes/Per/III/2008 tentang Rekam Medis*. Jakarta: Kementerian Kesehatan Republik Indonesia.
- Keraf, G. (1984). *Tata Bahasa Indonesia*. Flores: Nusa Indah.
- Keraf, G. (1999). *Tata Bahasa Rujukan Bahasa Indonesia*. Jakarta: P.T. Gramedia Widiasarana.
- Krause, W. B. (2017). *Forming Indonesian Words & Using Indonesian Affixes*. Diakses 10 Juni 2017, dari <http://indodic.com/affixeng.html>.
- Kridalaksana, H. (2001). *Kamus Linguistik*. Jakarta: P.T. Gramedia Pustaka Utama.
- Kumar, P. (2012). *Natural Language Process: New Orientations in English Language Practices*. Makalah dipresentasikan pada National Conference on English Language and Literature, Karnataka, 2012.
- Liddy, E. D. (2001). *Natural Language Processing*. New York: Encyclopedia of Library and Information Science, Marcel Decker Inc.
- Lyons, J. (1991). *Natural Language and Universal Grammar*. New York: Cambridge University Press.
- Madeleine, B. (1995). Models of Natural Language Understanding. *Proceedings of National Academy Science*, pp. 9977-9982.
- Malmasi, S., Sandor, N. L., Hosomura, N., Goldberg, M., Skentzos, S., Turchin, A. (2017). Canary: An NLP Platform for Clinicians and Researchers. *Appl Clin Inform* 2017; 8: 447-453.
- Manaf, N. A. (2009). *Sintaksis: Teori dan Terapannya dalam Bahasa Indonesia*. Padang: Sukabina Press.

- Markum H.M.S. (2007). *Penuntun Anamnesis dan Pemeriksaan Fisis*. Jakarta: Pusat Penerbitan Departemen Ilmu Penyakit Dalam Fakultas Kedokteran Universitas Indonesia.
- Martin, E. A. (2010). *Concise Medical Dictionary* (8th ed.). Oxford: Oxford University Press.
- McCowan, I., Moore, D., Dines, J., Gatica-Perez, D., Flynn, M., Wellner, P., dan Boulard, H. (2005). *On the Use of Information Retrieval Measures for Speech recognition Evaluation* (IDIAP Research Report 04-73). Diambil dari <http://www.idiap.ch/ftp/reports/2004/rr04-73.pdf>
- Mendonça, E., Haas, J., Shagina, L., Larson, E., Friedman, C. (2005). Extracting Information on Pneumonia in Infants Using Natural Language Processing of Radiology Reports. *Journal of Biomedical Informatics*, 38(4), 314–321.
- Merriam Webster Inc. (2016). *Merriam-Websters Medical Dictionary* (1st ed.). Springfield, MA: Merriam Webster, Incorporated.
- Okpor M. D. (2014). Machine Translation Approaches: Issues and Challenges. *International Journal of Computer Science Issues*, Vol. 11, Issue 5, No 2, September 2014.
- Pereira, P., Rijo, R., Silva, C., Martinho, R. (2015). Text Mining Applied to Electronic Medical Records: A Literature Review. *International Journal of E-Health and Medical Communications (IJEHMC)*, Volume 6, Issue 3.
- Pusat Bahasa Departemen Pendidikan Nasional. (2008). *Kamus Bahasa Indonesia*. Jakarta Pusat, Daerah Khusus Ibukota Jakarta: Pusat Bahasa Departemen Pendidikan Nasional.
- Pusat Bahasa Departemen Pendidikan Nasional. (2008). *Tesaurus Bahasa Indonesia Pusat Bahasa*. Jakarta Pusat, Daerah Khusus Ibukota Jakarta: Departemen Pendidikan Nasional.
- Rabiner, L. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286.
- Rabiner, L., dan Juang, B. H. (1986). An Introduction to Hidden Markov Models. *ASSP Magazine, IEEE*, vol. 3, no. 1, pp. 4–16.
- Radev, D., Hovy, K., dan McKeown, K. (2002). Introduction to The Special Issue on Summarization. *Computational Linguistics*, vol. 28, pp. 399-408.
- Ramlan, M. (1985). *Morfologi Suatu Tinjauan Deskriptif*. Yogyakarta: UP Karyono.
- Ramlan, M. (2001). *Ilmu Bahasa Indonesia Sintaksis*. Yogyakarta: C.V. Karyono.

- Ratnasari, C. I., Kusumadewi, S., Rosita, L. (2014). Model Pengolahan Bahasa Alami untuk Perumusan Keluhan Pasien. *Seminar Nasional Informatika Medis (SNIMed)* V.
- Regina, B., dan Michael, C. (2005). *Natural Language Processing: Background and Overview*. Cambridge: Barzilay and Collins Lecture Note Series.
- Reshamwala, A., Mishra, D., dan Pawar, P. (2013). Review on Natural Language Processing. *IRACST Engineering Science and Technology: An International Journal (ESTIJ)*, 3, 113-116.
- Saetre, R. (2004). GeneTUC: Automatic Information Extraction from Biomedical Texts. *Proceedings of Computer Science Graduate Students Conference*.
- Singh, B., Kapur, N., Kaur, P. (2012). Speech Recognition with Hidden Markov Model: A Review. *International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 3, Maret 2012*.
- Sneddon, J. N., Adelaar, K. A., dan Djenar, D. N. (2015). *Indonesian: A Comprehensive Grammar*. Abingdon: Routledge.
- Steele, D., *Google Talk Neural Networks for Voice Recognition*. Diakses 25 September 2017, dari <http://www.androidheadlines.com/2014/10/google-talk-neural-networks-voice-recognition.html>.
- Stéphane, M., dan Peter, H. (2005). Comparing Natural Language Processing Tools to Extract Medical Problems from Narrative Text. *Proceedings of AMIA Symposium*, pp. 525-529, 2005.
- Tebelskis, J. (1995). *Speech recognition Using Neural Networks*. Pittsburgh: School of Computer Science Carnegie Mellon University.
- Tim Pengembang Pedoman Bahasa Indonesia. (2016). *Pedoman Umum Ejaan Bahasa Indonesia* (4th ed.). Jakarta Pusat, Daerah Khusus Ibukota Jakarta: Badan Pengembangan dan Pembinaan Bahasa Kementerian Pendidikan dan Kebudayaan.
- Training an acoustic model for CMU Sphinx*. (n.d.). Diakses September 30, 2017, dari <https://cmusphinx.github.io/wiki/tutorialam>.
- Utami, E., dan Hartati, S. (2007). Pendekatan Metode Rule Based dalam Mengalihbahasakan Teks Bahasa Inggris ke Teks Bahasa Indonesia. *Jurnal Informatika, vol.8, no.1, 2007*, pp: 42 – 53.
- Vyas, M. (2013). A Gaussian Mixture Model Based Speech Recognition System Using MATLAB. *Signal & Image Processing: An International Journal (SIPIJ) Vol.4, No.4, 2013*.

- Wahid, Adripratomo. (2012). *Sistem Pencatatan Rekaman Medik Secara Digital*. Diakses 30 September 2017, dari <http://repository.unhas.ac.id/handle/123456789/837>.
- Wang, S. C. (2003). Artificial Neural Network. *Interdisciplinary Computing in Java Programming*, pp. 81–100.
- Widjono, H. S. (2007). *Bahasa Indonesia: Mata Kuliah Pengembangan Kepribadian di Perguruan Tinggi*. Jakarta: Grasindo.
- Wolniewicz, R. (2015). Computer-Assisted Coding and Natural Language Processing. *3M Health Information Systems*.
- Wu, Y. (2005). Gaussian Mixture Model. *Connexions*.

## LAMPIRAN

Hasil uji coba masukan keluhan yang diisikan oleh kelima dokter beserta terjemahannya.

Masukan Keluhan oleh Dokter	Terjemahan oleh Aplikasi
1. nyeri kepala sejak 2 bulan lalu memberat saat istirahat 2. nyeri tenggorokan. nyeri telan 3. diare parah. nyeri perut 4. demam. bintik merah di kulit 5. nyeri boyok 6. kram otot tiap hari 7. batuk 8. kesemutan 9. mimisan 10. luka di kaki	<<PASIEN-1>>  [1] Keluhan: sakit kepala Lokasi: kepala Sistem Organ: - Istilah Medis: sefalgia Kemunculan: 2 bulan yang lalu Frekuensi: - Sifat: -  <<PASIEN-2>>  [1] Keluhan: nyeri tenggorokan Lokasi: tenggorokan Sistem Organ: Sistem Pencernaan Istilah Medis: faringitis Kemunculan: - Frekuensi: - Sifat: -  [2] Keluhan: sakit saat menelan Lokasi: tenggorokan Sistem Organ: sistem pencernaan Istilah Medis: odinofagia Kemunculan: - Frekuensi: - Sifat: -  <<PASIEN-3>>  [1] Keluhan: diare Lokasi: sistem pencernaan Sistem Organ: sistem pencernaan Istilah Medis: gastroenteritis

	<p>Kemunculan: - Frekuensi: - Sifat: parah</p> <p>[2] Keluhan: nyeri pada perut Lokasi: perut Sistem Organ: - Istilah Medis: abdominalgia Kemunculan: - Frekuensi: - Sifat: -</p> <p>&lt;&lt;PASIEN-4&gt;&gt;</p> <p>[1] Keluhan: demam Lokasi: - Sistem Organ: - Istilah Medis: pireksia Kemunculan: - Frekuensi: - Sifat: -</p> <p>[2] Keluhan: kulit merah Lokasi: tubuh Sistem Organ: sistem integumen Istilah Medis: ruam Kemunculan: - Frekuensi: - Sifat: -</p> <p>&lt;&lt;PASIEN-5&gt;&gt;</p> <p>[1] Keluhan: nyeri pada boyok Lokasi: boyok Sistem Organ: - Istilah Medis: lumbalgia Kemunculan: - Frekuensi: - Sifat: -</p>
--	--

	<p>&lt;&lt;PASIEN-6&gt;&gt;</p> <p>[1] Keluhan: kram Lokasi: otot Sistem Organ: sistem muskuloskeletal Istilah Medis: kram Kemunculan: - Frekuensi: tiap hari Sifat: -</p> <p>&lt;&lt;PASIEN-7&gt;&gt;</p> <p>[1] Keluhan: batuk Lokasi: tenggorokan Sistem Organ: sistem pencernaan Istilah Medis: tuisis Kemunculan: - Frekuensi: - Sifat: -</p> <p>&lt;&lt;PASIEN-8&gt;&gt;</p> <p>[1] Keluhan: tubuh kesemutan Lokasi: tubuh Sistem Organ: - Istilah Medis: parestesia Kemunculan: - Frekuensi: - Sifat: -</p> <p>&lt;&lt;PASIEN-9&gt;&gt;</p> <p>[1] Keluhan: mimisan Lokasi: hidung Sistem Organ: sistem pernapasan Istilah Medis: epistaksis Kemunculan: - Frekuensi: - Sifat: -</p>
--	---



	<p>&lt;&lt;PASIEN-10&gt;&gt;</p> <p>[1]  Keluhan: luka  Lokasi: kaki  Sistem Organ: sistem integumen  Istilah Medis: vulnus  Kemunculan: -  Frekuensi: -  Sifat: -</p>
<ol style="list-style-type: none"> <li>1. nyeri tengkuk</li> <li>2. BAB berdarah</li> <li>3. Nyeri perut, mual, muntah</li> <li>4. Batuk berdahak</li> <li>5. Bersin-bersin setiap pagi, hidung tersumbat</li> <li>6. Pusing berputar</li> <li>7. Keluar cairan dari telinga</li> <li>8. Pegel-pegel di bagian boyok</li> <li>9. Kesemutan pada jari-jari</li> <li>10. Nyeri telan, suara serak</li> </ol>	<p>&lt;&lt;PASIEN-1&gt;&gt;</p> <p>[1]  Keluhan: nyeri leher  Lokasi: leher  Sistem Organ: -  Istilah Medis: servikalgia  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>&lt;&lt;PASIEN-2&gt;&gt;</p> <p>[1]  Keluhan: pendarahan  Lokasi: pembuluh darah  Sistem Organ: sistem peredaran darah dan getah  Istilah Medis: hemorage  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>&lt;&lt;PASIEN-3&gt;&gt;</p> <p>[1]  Keluhan: nyeri pada perut  Lokasi: perut  Sistem Organ: -  Istilah Medis: abdominalgia  Kemunculan: -  Frekuensi: -  Sifat: -</p>

	<p>[2]  Keluhan: mual  Lokasi: perut  Sistem Organ: sistem pencernaan  Istilah Medis: nausea  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>[3]  Keluhan: muntah  Lokasi: tenggorokan  Sistem Organ: sistem pencernaan  Istilah Medis: vomitus  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>&lt;&lt;PASIEN-4&gt;&gt;</p> <p>[1]  Keluhan: batuk  Lokasi: tenggorokan  Sistem Organ: sistem pencernaan  Istilah Medis: tuisis  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>&lt;&lt;PASIEN-5&gt;&gt;</p> <p>[1]  Keluhan: tidak ada di dalam <i>database</i>  Lokasi: tidak ada di dalam <i>database</i>  Sistem Organ: tidak ada di dalam <i>database</i>  Istilah Medis: tidak ada di dalam <i>database</i>  Kemunculan: -  Frekuensi: setiap pagi  Sifat: -</p> <p>[2]  Keluhan: tidak ada di dalam <i>database</i>  Lokasi: tidak ada di dalam <i>database</i>  Sistem Organ: tidak ada di dalam <i>database</i>  Istilah Medis: tidak ada di dalam <i>database</i>  Kemunculan: -</p>
--	---

	<p>Frekuensi: - Sifat: -</p> <p>&lt;&lt;PASIEN-6&gt;&gt;</p> <p>[1] Keluhan: pusing Lokasi: kepala Sistem Organ: - Istilah Medis: sefalgia Kemunculan: - Frekuensi: - Sifat: -</p> <p>&lt;&lt;PASIEN-7&gt;&gt;</p> <p>[1] Keluhan: tidak ada di dalam <i>database</i> Lokasi: tidak ada di dalam <i>database</i> Sistem Organ: tidak ada di dalam <i>database</i> Istilah Medis: tidak ada di dalam <i>database</i> Kemunculan: - Frekuensi: - Sifat: -</p> <p>&lt;&lt;PASIEN-8&gt;&gt;</p> <p>[1] Keluhan: tidak ada di dalam <i>database</i> Lokasi: tidak ada di dalam <i>database</i> Sistem Organ: tidak ada di dalam <i>database</i> Istilah Medis: tidak ada di dalam <i>database</i> Kemunculan: - Frekuensi: - Sifat: -</p> <p>&lt;&lt;PASIEN-9&gt;&gt;</p> <p>[1] Keluhan: jari kesemutan Lokasi: jari Sistem Organ: -</p>
--	--

	<p>Istilah Medis: parestesia  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>&lt;&lt;PASIEN-10&gt;&gt;</p> <p>[1]  Keluhan: sakit saat menelan  Lokasi: tenggorokan  Sistem Organ: sistem pencernaan  Istilah Medis: odinofagia  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>[2]  Keluhan: serak  Lokasi: tenggorokan  Sistem Organ: sistem pencernaan  Istilah Medis: disfonia  Kemunculan: -  Frekuensi: -  Sifat: -</p>
<ol style="list-style-type: none"> <li>1. sakit pinggang. sakit perut.</li> <li>2. sesak. sakit dada.</li> <li>3. nyeri kencing dan kencing berdarah</li> <li>4. batuk dan pilek</li> <li>5. sakit menelan</li> <li>6. pusing berputar</li> <li>7. mual dan kembung</li> <li>8. susah kencing.</li> <li>9. susah bab</li> <li>10. diare.</li> </ol>	<p>&lt;&lt;PASIEN-1&gt;&gt;</p> <p>[1]  Keluhan: nyeri pada pinggang  Lokasi: pinggang  Sistem Organ: -  Istilah Medis: koxalgia  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>[2]  Keluhan: nyeri pada perut  Lokasi: perut  Sistem Organ: -  Istilah Medis: abdominalgia  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>&lt;&lt;PASIEN-2&gt;&gt;</p>

	<p>[1]  Keluhan: tidak ada di dalam <i>database</i>  Lokasi: tidak ada di dalam <i>database</i>  Sistem Organ: tidak ada di dalam <i>database</i>  Istilah Medis: tidak ada di dalam <i>database</i>  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>[2]  Keluhan: nyeri pada dada  Lokasi: dada  Sistem Organ: -  Istilah Medis: torakalgia  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>&lt;&lt;PASIEN-3&gt;&gt;</p> <p>[1]  Keluhan: sakit saat kencing  Lokasi: sistem urinaria  Sistem Organ: sistem urinaria  Istilah Medis: disuria  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>[2]  Keluhan: hematuria  Lokasi: hematuria  Sistem Organ: sistem urinaria  Istilah Medis: kencing berdarah  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>&lt;&lt;PASIEN-4&gt;&gt;</p> <p>[1]  Keluhan: batuk  Lokasi: tenggorokan  Sistem Organ: sistem pencernaan  Istilah Medis: tuisis</p>
--	---

	<p>Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>[2]  Keluhan: pilek  Lokasi: hidung  Sistem Organ: sistem pernapasan  Istilah Medis: rinore  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>&lt;&lt;PASIEN-5&gt;&gt;</p> <p>[1]  Keluhan: sakit saat menelan  Lokasi: tenggorokan  Sistem Organ: sistem pencernaan  Istilah Medis: odinofagia  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>&lt;&lt;PASIEN-6&gt;&gt;</p> <p>[1]  Keluhan: pusing  Lokasi: kepala  Sistem Organ: -  Istilah Medis: sefalgia  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>&lt;&lt;PASIEN-7&gt;&gt;</p> <p>[1]  Keluhan: mual  Lokasi: perut  Sistem Organ: sistem pencernaan  Istilah Medis: nausea  Kemunculan: -  Frekuensi: -</p>
--	---

	<p>Sifat: -</p> <p>[2] Keluhan: perut kembung Lokasi: perut Sistem Organ: sistem pencernaan Istilah Medis: flatulensi Kemunculan: - Frekuensi: - Sifat: -</p> <p>&lt;&lt;PASIEN-8&gt;&gt;</p> <p>[1] Keluhan: kencing tertahan Lokasi: kandung kemih Sistem Organ: sistem urinaria Istilah Medis: retensi urin Kemunculan: - Frekuensi: - Sifat: -</p> <p>&lt;&lt;PASIEN-9&gt;&gt;</p> <p>[1] Keluhan: sembelit Lokasi: sistem pencernaan Sistem Organ: sistem pencernaan Istilah Medis: konstipasi Kemunculan: - Frekuensi: - Sifat: -</p> <p>&lt;&lt;PASIEN-10&gt;&gt;</p> <p>[1] Keluhan: diare Lokasi: sistem pencernaan Sistem Organ: sistem pencernaan Istilah Medis: gastroenteritis Kemunculan: - Frekuensi: - Sifat: -</p>
--	---

<p>1. nyeri uluhati. mual. muntah. dada panas. lidah pahit.</p> <p>2. demam sejak tiga hari. muncul ruam. demam turun hari ketiga.</p> <p>3. pusing berputar. telinga berdenging. mual</p> <p>4. diare cair sejak satu hari. mual (-). muntah (-).</p> <p>5. nyeri punggung belakang. riwayat trauma (-)</p> <p>6. gatal di tungkai bawah. nanah positif.</p> <p>7. demam, batuk, pilek</p> <p>8. gatal di perut bawah. ada bercak putih-putih kalau digaruk. berkeringat makin gatal.</p> <p>9. nyeri telinga kiri post berenang.</p> <p>10. nyeri tenggorokan. terasa bengkak di leher kanan kiri. demam. nyeri menelan.</p>	<p>&lt;&lt;PASIEN-1&gt;&gt;</p> <p>[1] Keluhan: nyeri pada ulu hati Lokasi: uluhati Sistem Organ: - Istilah Medis: pirosis Kemunculan: - Frekuensi: - Sifat: -</p> <p>[2] Keluhan: mual Lokasi: perut Sistem Organ: sistem pencernaan Istilah Medis: nausea Kemunculan: - Frekuensi: - Sifat: -</p> <p>[3] Keluhan: muntah Lokasi: tenggorokan Sistem Organ: sistem pencernaan Istilah Medis: vomitus Kemunculan: - Frekuensi: - Sifat: -</p> <p>[4] Keluhan: tidak ada di dalam <i>database</i> Lokasi: tidak ada di dalam <i>database</i> Sistem Organ: tidak ada di dalam <i>database</i> Istilah Medis: tidak ada di dalam <i>database</i> Kemunculan: - Frekuensi: - Sifat: -</p> <p>[5] Keluhan: mulut pahit Lokasi: mulut Sistem Organ: sistem pencernaan Istilah Medis: disgeusia Kemunculan: - Frekuensi: - Sifat: -</p> <p>&lt;&lt;PASIEN-2&gt;&gt;</p>
--	---



	<p>[1]  Keluhan: demam  Lokasi: -  Sistem Organ: -  Istilah Medis: pireksia  Kemunculan: 3 hari yang lalu  Frekuensi: -  Sifat: -</p> <p>[2]  Keluhan: kulit merah  Lokasi: tubuh  Sistem Organ: sistem integumen  Istilah Medis: ruam  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>[3]  Keluhan: demam  Lokasi: -  Sistem Organ: -  Istilah Medis: pireksia  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>&lt;&lt;PASIEN-3&gt;&gt;</p> <p>[1]  Keluhan: pusing  Lokasi: kepala  Sistem Organ: -  Istilah Medis: sefalgia  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>[2]  Keluhan: telinga berdenging  Lokasi: telinga  Sistem Organ: sistem indera  Istilah Medis: tinitus  Kemunculan: -  Frekuensi: -  Sifat: -</p>
--	--

	<p>[3]  Keluhan: mual  Lokasi: perut  Sistem Organ: sistem pencernaan  Istilah Medis: nausea  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>&lt;&lt;PASIEN-4&gt;&gt;</p> <p>[1]  Keluhan: diare  Lokasi: sistem pencernaan  Sistem Organ: sistem pencernaan  Istilah Medis: gastroenteritis  Kemunculan: 1 hari yang lalu  Frekuensi: -  Sifat: -</p> <p>[2]  Keluhan: mual(-)  Lokasi: perut  Sistem Organ: sistem pencernaan  Istilah Medis: nausea  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>[3]  Keluhan: muntah(-)  Lokasi: tenggorokan  Sistem Organ: sistem pencernaan  Istilah Medis: vomitus  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>&lt;&lt;PASIEN-5&gt;&gt;</p> <p>[1]  Keluhan: nyeri pada punggung  Lokasi: punggung  Sistem Organ: -  Istilah Medis: dorsalgia  Kemunculan: -</p>
--	--

	<p>Frekuensi: - Sifat: -</p> <p>[2] Keluhan: tubuh trauma(-) Lokasi: tubuh Sistem Organ: - Istilah Medis: trauma Kemunculan: - Frekuensi: - Sifat: -</p> <p>&lt;&lt;PASIEN-6&gt;&gt;</p> <p>[1] Keluhan: tubuh gatal Lokasi: tubuh Sistem Organ: - Istilah Medis: pruritus Kemunculan: - Frekuensi: - Sifat: -</p> <p>[2] Keluhan: tubuh bernanah Lokasi: tubuh Sistem Organ: - Istilah Medis: purulen Kemunculan: - Frekuensi: - Sifat: -</p> <p>&lt;&lt;PASIEN-7&gt;&gt;</p> <p>[1] Keluhan: demam Lokasi: - Sistem Organ: - Istilah Medis: pireksia Kemunculan: - Frekuensi: - Sifat: -</p> <p>[2] Keluhan: batuk Lokasi: tenggorokan</p>
--	--

	<p>Sistem Organ: sistem pencernaan  Istilah Medis: tuisis  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>[3]  Keluhan: pilek  Lokasi: hidung  Sistem Organ: sistem pernapasan  Istilah Medis: rinore  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>&lt;&lt;PASIEN-8&gt;&gt;</p> <p>[1]  Keluhan: perut gatal  Lokasi: perut  Sistem Organ: -  Istilah Medis: pruritus  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>[2]  Keluhan: flek  Lokasi: tubuh  Sistem Organ: sistem integumen  Istilah Medis: spot  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>[3]  Keluhan: tubuh gatal  Lokasi: tubuh  Sistem Organ: -  Istilah Medis: pruritus  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>&lt;&lt;PASIEN-9&gt;&gt;</p>
--	---

	<p>[1]  Keluhan: nyeri telinga  Lokasi: telinga  Sistem Organ: sistem indera  Istilah Medis: otalgia  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>&lt;&lt;PASIEN-10&gt;&gt;</p> <p>[1]  Keluhan: nyeri tenggorokan  Lokasi: tenggorokan  Sistem Organ: Sistem Pencernaan  Istilah Medis: faringitis  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>[2]  Keluhan: leher bengkak  Lokasi: leher  Sistem Organ: -  Istilah Medis: edema  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>[3]  Keluhan: demam  Lokasi: -  Sistem Organ: -  Istilah Medis: pireksia  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>[4]  Keluhan: sakit saat menelan  Lokasi: tenggorokan  Sistem Organ: sistem pencernaan  Istilah Medis: odinofagia  Kemunculan: -  Frekuensi: -  Sifat: -</p>
1. muntah. diare	<<PASIEN-1>>

<p>2. nyeri ulu hati  3. BAB berdarah  4. demam lama  5. batuk lama  6. nyeri sendi  7. nyeri boyok  8. lemas  9. mimisan  10. sakit gigi</p>	<p>[1]  Keluhan: muntah  Lokasi: tenggorokan  Sistem Organ: sistem pencernaan  Istilah Medis: vomitus  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>[2]  Keluhan: diare  Lokasi: sistem pencernaan  Sistem Organ: sistem pencernaan  Istilah Medis: gastroenteritis  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>&lt;&lt;PASIEN-2&gt;&gt;</p> <p>[1]  Keluhan: nyeri pada ulu hati  Lokasi: uluhati  Sistem Organ: -  Istilah Medis: pirosis  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>&lt;&lt;PASIEN-3&gt;&gt;</p> <p>[1]  Keluhan: pendarahan  Lokasi: pembuluh darah  Sistem Organ: sistem peredaran darah dan getah  Istilah Medis: hemorage  Kemunculan: -  Frekuensi: -  Sifat: -</p> <p>&lt;&lt;PASIEN-4&gt;&gt;</p>
---	--

	<p>[1] Keluhan: demam Lokasi: - Sistem Organ: - Istilah Medis: pireksia Kemunculan: - Frekuensi: - Sifat: -</p> <p>&lt;&lt;PASIEN-5&gt;&gt;</p> <p>[1] Keluhan: batuk Lokasi: tenggorokan Sistem Organ: sistem pencernaan Istilah Medis: tuisis Kemunculan: - Frekuensi: - Sifat: -</p> <p>&lt;&lt;PASIEN-6&gt;&gt;</p> <p>[1] Keluhan: nyeri sendi Lokasi: sendi Sistem Organ: sistem muskuloskeletal Istilah Medis: artralgia Kemunculan: - Frekuensi: - Sifat: -</p> <p>&lt;&lt;PASIEN-7&gt;&gt;</p> <p>[1] Keluhan: nyeri pada boyok Lokasi: boyok Sistem Organ: - Istilah Medis: lumbalgia Kemunculan: - Frekuensi: - Sifat: -</p>
--	---

	<p>&lt;&lt;PASIEN-8&gt;&gt;</p> <p>[1] Keluhan: tubuh lemah Lokasi: tubuh Sistem Organ: - Istilah Medis: weakness Kemunculan: - Frekuensi: - Sifat: -</p> <p>&lt;&lt;PASIEN-9&gt;&gt;</p> <p>[1] Keluhan: mimisan Lokasi: hidung Sistem Organ: sistem pernapasan Istilah Medis: epistaksis Kemunculan: - Frekuensi: - Sifat: -</p> <p>&lt;&lt;PASIEN-10&gt;&gt;</p> <p>[1] Keluhan: tidak ada di dalam <i>database</i> Lokasi: tidak ada di dalam <i>database</i> Sistem Organ: tidak ada di dalam <i>database</i> Istilah Medis: tidak ada di dalam <i>database</i> Kemunculan: - Frekuensi: - Sifat: -</p>
--	--