

PEMROSESAN BAHASA ALAMI UNTUK PENERJEMAHAN KELUHAN PADA ANAMNESIS DALAM BENTUK TEKS ATAU SUARA

Pradipta Yuwono

Program Studi Teknik Informatika Universitas Islam Indonesia

Jalan Kaliurang KM. 14.5, Sleman Indonesia

Email: pradipta_yuwono@yahoo.co.id

ABSTRAK

Anamnesis adalah wawancara yang dilakukan oleh dokter terhadap pasien atau orang-orang terdekat pasien tersebut untuk mendapatkan data-data medis mengenai pasien tersebut, salah satu data yang dikumpulkan dari anamnesis adalah data keluhan pasien. Pada umumnya data-data keluhan pasien sulit diproses oleh komputer karena umumnya tercatat dalam bahasa sehari-hari yang kurang baku dan tidak terstruktur, padahal data keluhan pasien berisi informasi-informasi penting yang dapat digunakan sebagai masukan untuk pengolahan data pada Sistem Pendukung Keputusan Klinis. Penelitian ini bertujuan untuk mengembangkan sebuah aplikasi yang mampu menerjemahkan data keluhan pasien dalam bahasa yang kurang baku dan tidak terstruktur tersebut ke dalam bahasa medis yang lebih baku dan terstruktur menggunakan algoritma-algoritma Pemrosesan Bahasa Alami. Masukan keluhan pasien yang digunakan dalam penelitian ini dapat berupa teks maupun rekaman suara pasien yang nantinya akan diubah menjadi teks menggunakan fitur speech recognition yang terdapat pada aplikasi. User dapat menambahkan maupun memperbaiki data-data keluhan dan terjemahan medisnya pada aplikasi agar hasil penerjemahan keluhan pasien menjadi semakin akurat. User juga dapat menambahkan kata-kata baru untuk dapat dikenali oleh fitur speech recognition pada aplikasi. Hasil uji coba aplikasi yang sudah dikembangkan menunjukkan bahwa aplikasi sudah mampu menerjemahkan keluhan pasien dengan tingkat akurasi yang cukup memuaskan dan mampu mengenali masukan rekaman suara dengan cukup baik.

Kata Kunci: *pemrosesan bahasa alami, keluhan pasien, anamnesis, speech recognition*

1. PENDAHULUAN

Anamnesis adalah tahap awal dari rangkaian pemeriksaan pasien yang dilakukan oleh dokter untuk penegakan diagnostik. Tujuan anamnesis adalah untuk mendapatkan informasi menyeluruh dari pasien sehingga diharapkan dokter mampu menyimpulkan dugaan organ atau sistem organ yang terganggu termasuk rumusan masalah kliniknya, selain itu tujuan anamnesis lainnya adalah untuk membina hubungan baik antara dokter dengan pasiennya (Markum, 2007). Data anamnesis terdiri atas identitas pasien, riwayat penyakit pasien, keluhan utama pasien, riwayat penyakit keluarga pasien, serta informasi pasien lainnya yang diperlukan untuk penegakan diagnostik. Data anamnesis ini nantinya akan disimpan di dalam rekam medis (PERMENKES, 2008).

Saat ini Rekam medis berbasis elektronik mulai menggantikan rekam medis berbasis kertas karena rekam medis berbasis kertas memiliki beberapa kekurangan seperti rekam medis berbasis kertas dapat mudah rusak atau hilang, penyimpanan rekam medis berbasis kertas membutuhkan ruangan besar, dan tulisan pada rekam medis berbasis kertas sering sulit dibaca oleh sebagian orang (Ajami & Bagheri-Tadi, 2013). Data rekam medis berbasis elektronik juga lebih mudah untuk diolah oleh komputer karena data tersebut tersimpan dalam format teks *digital* yang dapat dibaca langsung oleh komputer. Meskipun begitu, data teks pada rekam medis elektronik sering ditulis menggunakan format penulisan bebas atau format penulisan tidak terstruktur yang mempersulit proses penggalan data teks pada rekam medis elektronik secara otomatis menggunakan komputer (Pereira, Rijo, Silva, & Martinho, 2015). Padahal, data teks tidak terstruktur pada rekam medis elektronik tersebut mengandung banyak

informasi penting mengenai pasien yang dapat digunakan sebagai masukan untuk pengolahan data pada Sistem Pendukung Keputusan Klinis. Jumlah data rekam medis elektronik yang sangat banyak juga menyebabkan sangat sulitnya melakukan penggalian teks pada data rekam medis elektronik secara manual oleh manusia.

Untuk mengatasi permasalahan pengolahan data keluhan pasien pada rekam medis elektronik yang ditulis menggunakan format penulisan yang kurang baku dan tidak terstruktur tersebut, penulis melakukan penelitian untuk mengembangkan sebuah aplikasi yang mampu menerjemahkan data keluhan pasien tersebut ke dalam format teks yang lebih baku dan terstruktur sehingga teks tersebut dapat lebih mudah dibaca dan diolah oleh komputer. Penulis menggunakan beberapa metode *Natural Language Processing* (NLP) untuk menerjemahkan data keluhan pasien tersebut. *Natural Language Processing* (NLP) adalah sebuah cabang dari Ilmu Komputer yang berhubungan dengan pemrosesan cerdas pada bahasa manusia (Ford, Carroll, Smith, Scott, & Cassell, 2016). Salah satu pendekatan yang digunakan dalam penerjemahan data keluhan pasien pada aplikasi ini adalah penyederhanaan dari penerjemahan dengan pendekatan statistik. Penerjemahan dengan pendekatan statistik dihasilkan dari basis model-model statistik yang parameter-parameternya diturunkan dari analisis teks *corpora bilingual* (Okpor, 2014). Selain dengan pendekatan statistik, penerjemahan juga dilakukan dengan pencocokan kata-kata atau frase-frase kunci pada data keluhan dan dengan contoh penerjemahan yang diberikan oleh *user*.

Masukan data keluhan pasien selain dalam bentuk teks juga dapat dalam bentuk suara pasien maupun dokter. Masukan suara ini dapat dimasukkan langsung ke dalam aplikasi menggunakan mikrofon, maupun dari mengimpor rekaman suara yang tersimpan di dalam media penyimpanan. Agar aplikasi mampu mengenali masukan suara dari pasien maupun dokter, aplikasi ini dilengkapi dengan fitur *Automatic Speech Recognition* yang dikembangkan menggunakan Google Cloud Speech API dan Sphinx4 API.

Automatic Speech Recognition adalah sistem yang mengkonversi suara dari sebuah rekaman sinyal audio menjadi teks. Salah satu metode *Automatic Speech Recognition* yang paling umum dan yang terbaik saat ini adalah menggunakan pendekatan probabilistik. Pada pendekatan ini, setiap *input* suara yang dimasukkan bersesuaian terhadap kata atau kumpulan kata tertentu pada kamus yang ada pada aplikasi dengan sebuah probabilitas tertentu. Kata atau kumpulan kata dengan probabilitas tertinggi akan dipilih sebagai hasil pengenalan dari suara tersebut (Gruhn, Minker, & Nakamura, 2011).

2. BAHAN DAN METODE PENELITIAN

Untuk mengembangkan aplikasi pada penelitian ini penulis membutuhkan beberapa data sebagai berikut:

1. Data keluhan pasien yang paling umum dan terjemahan bakunya
2. Data kata-kata Bahasa Indonesia yang paling umum
3. Data kamus fonetis, model bahasa, dan model akustik untuk pengenalan suara

Penulis mengumpulkan data-data keluhan pasien yang paling umum dari situs-situs konsultasi kedokteran *online* populer di Indonesia seperti klikdokter.com, alodokter.com, dokter.id, dan tanyadok.com. Dari hasil analisis keluhan-keluhan pasien pada situs-situs tersebut, penulis memperoleh daftar sekitar 600 keluhan pasien yang paling sering muncul pada situs-situs tersebut. Keluhan dengan makna yang sama namun ditulis secara berbeda, misal ditulis dengan sinonimnya, dianggap sebagai satu keluhan. Untuk istilah medis yang baku dari daftar keluhan pasien tersebut, penulis menanyakan langsung kepada beberapa dokter dan dari membaca-baca beberapa kamus istilah kedokteran. Selain keluhan pasien dengan istilah medisnya yang baku, penulis juga harus mencari tahu penjelasan lokasi dan sistem organ dari keluhan tersebut, data ini penulis dapatkan dengan membaca-baca beberapa literatur kedokteran dasar. Semua data ini digunakan untuk proses penerjemahan *default* pada aplikasi. Nantinya *user* dapat menambahkan data-data keluhan dan terjemahan bakunya yang baru atau mengedit data-data yang sudah ada agar proses penerjemahan keluhan pasien menjadi lebih akurat.

Data kata-kata Bahasa Indonesia yang paling umum akan digunakan untuk menganalisis data keluhan pasien dalam proses penerjemahan dan juga digunakan untuk pengoreksian otomatis

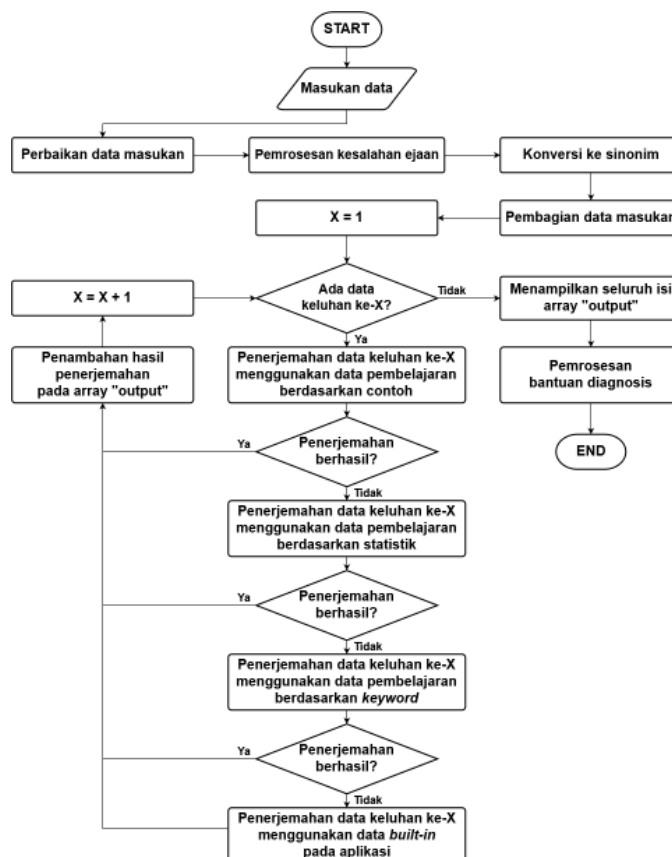
penulisan teks yang salah pada data masukan keluhan pasien. *User* sering salah mengetikkan teks masukan, misalnya mengetikkan "ksemutan" dimana seharusnya penulisannya yang tepat adalah "kesemutan". Dengan pengoreksian penulisan teks masukan secara otomatis, aplikasi akan memberikan peringatan kepada *user* akan adanya kesalahan ketik dan menyarankan kata lain yang memiliki probabilitas terbesar sebagai kata yang dimaksudkan oleh *user*. Data kata-kata yang paling umum dalam Bahasa Indonesia beserta probabilitas kemunculannya ini penulis dapatkan dengan penggalian teks yang ada pada situs Wikipedia Indonesia dan beberapa situs konsultasi kedokteran *online*. Untuk menganalisis data keluhan pasien dalam proses penerjemahan, aplikasi juga membutuhkan kelas kata dan daftar sinonim dari kata-kata tersebut. Untuk pencarian kelas kata pada kata-kata tersebut penulis menggunakan Kamus Besar Bahasa Indonesia (KBBI), dan untuk pencarian daftar sinonim dari kata-kata tersebut penulis mencarinya dari beberapa buku Tesaarus untuk Bahasa Indonesia. Nantinya *user* juga dapat menambahkan kata-kata baru untuk dikenali oleh aplikasi.

Data kamus fonetis, model bahasa, dan model akustik digunakan untuk fitur pengenalan suara menggunakan Sphinx4. Data kamus fonetis penulis ambil dari daftar kata-kata yang paling sering digunakan pada data keluhan pasien. Model bahasa penulis kembangkan dari kumpulan teks-teks pada situs-situs konsultasi kedokteran *online* dan Wikipedia Indonesia. Sedangkan untuk model akustik penulis dapatkan dari rekaman suara dari hasil pembacaan kata-kata pada kamus fonetis oleh manusia dan komputer. Nantinya *user* juga dapat menambahkan data kamus fonetis, model bahasa, dan model akustik agar proses pengenalan suara oleh aplikasi menggunakan Sphinx4 menjadi lebih akurat.

3. HASIL DAN PEMBAHASAN

3.1. GAMBARAN MODEL

Gambaran model proses penerjemahan data masukan keluhan pasien pada aplikasi ini dapat dilihat pada Gambar 1 di bawah ini:



Gambar 1. Model Penerjemahan Masukan Keluhan Pasien

Adapun penjelasan dari Gambar 1 di atas adalah sebagai berikut:

a. Masukan Data Keluhan Pasien

Masukan data keluhan pasien dapat berupa teks maupun suara. Masukan teks dapat diketikkan langsung ke dalam aplikasi maupun diimpor dari *file* berformat teks atau *excel* dengan format penulisan yang sudah ditentukan. Masukan suara dapat dimasukkan langsung ke dalam aplikasi menggunakan mikrofon maupun diimpor dari rekaman suara berformat WAV atau MP3 yang nantinya akan dikonversi ke format yang sesuai menggunakan aplikasi *open source* FFmpeg secara otomatis. Masukan suara tersebut nantinya diubah menjadi masukan teks menggunakan fitur pengenalan suara pada aplikasi. Aplikasi ini menggunakan Google Cloud Speech dan Sphinx4 yang *open source* untuk mengkonversi masukan suara menjadi teks. *User* dapat memilih apakah ingin menggunakan Google Cloud Speech atau Sphinx4 untuk mengkonversi masukan suara menjadi teks.

b. Perbaikan Data Masukan

Data teks masukan kemudian akan diperbaiki agar dapat diolah oleh komputer dengan tepat dan tidak terjadi ambiguitas. Perbaikan-perbaikan tersebut antara lain mengubah seluruh teks menjadi huruf kecil semua, menghapus kata-kata partikel yang tidak diperlukan dalam proses penerjemahan (misalnya: ah, dong, duh, loh, dll.), memperbaiki penulisan singkatan agar sesuai dengan tata bahasa baku Bahasa Indonesia, memperbaiki tanda baca agar sesuai dengan tata bahasa baku Bahasa Indonesia, menghapus baris kosong pada teks masukan, menghapus spasi-spasi di awal dan akhir kalimat (*trimming*), dan mengubah spasi ganda menjadi spasi tunggal. Data teks masukan yang sudah diperbaiki ini akan disimpan dalam sebuah variabel dan tidak dapat dilihat oleh *user*.

c. Pengoreksian Kesalahan Ejaan/Penulisan

Data teks masukan kemudian dicek seluruh kata-katanya apakah ada kesalahan penulisan. Kesalahan penulisan tersebut dapat berupa kurang satu huruf, misalnya kata "kesemutan" salah tulis menjadi "ksemutan"; kelebihan satu huruf, misalnya kata "keracunan" salah tulis menjadi "keracunaan"; salah tulis satu huruf, misalnya kata "diare" salah tulis menjadi "diari"; dan dua huruf yang berdekatan tertukar, misalnya kata "muntah" salah tulis menjadi "mutnah". Setiap kata pada teks masukan akan dicek apakah ada di dalam daftar kata-kata yang dikenali oleh aplikasi, jika tidak ada maka aplikasi akan mencari kata lain yang paling mirip dengan kata tersebut. Jika ada beberapa kata yang paling mirip, aplikasi akan memilih kata dengan probabilitas kemunculan pada teks Bahasa Indonesia yang paling tinggi. Setelah mendapatkan kata yang paling sesuai, aplikasi akan menampilkan peringatan kepada *user* bahwa kemungkinan ada kesalahan penulisan pada kata tertentu dan menampilkan kata rekomendasi yang kemungkinan besar adalah kata yang dimaksudkan oleh *user*. Jika *user* mengklik kata rekomendasi tersebut, maka aplikasi secara otomatis akan memperbaiki seluruh kata yang salah penulisannya pada data teks masukan menjadi kata rekomendasi tersebut.

d. Konversi ke Sinonim

Kata-kata pada data teks masukan kemudian diubah ke bentuk sinonimnya yang paling umum. Hal ini dilakukan agar dua kata yang memiliki makna sama tidak dianggap berbeda oleh komputer. Misalnya frase "nyeri telinga" dan "nyeri kuping" menunjukkan keluhan yang sama namun ditulis dengan kata yang berbeda, sehingga untuk memudahkan pemrosesan oleh komputer semua kata "kuping" pada teks masukan diubah menjadi kata "telinga". Proses konversi ke sinonim juga dilakukan untuk mengubah kata-kata dalam bentuk yang tidak baku ke bentuk yang baku, misalnya kata "gak" menjadi "tidak"; mengubah singkatan-singkatan menjadi kepanjangannya, misalnya kata "yg" menjadi "yang"; dan mengubah penulisan bilangan dalam bentuk kata menjadi bentuk angka, misalnya kata "seratus lima puluh" menjadi "150".

e. Pembagian Data Masukan

Data teks masukan yang sudah diperbaiki dan kata-katanya sudah dikonversi ke sinonimnya kemudian dibagi menjadi data masukan untuk tiap-tiap pasien, karena data teks masukan dapat berisi

daftar keluhan untuk lebih dari satu pasien yang dipisahkan oleh baris baru. Kemudian data masukan untuk tiap-tiap pasien tersebut dibagi lagi menjadi data teks keluhan-keluhan berbeda yang dipisahkan oleh tanda titik (.).

f. Penerjemahan Data Keluhan

Ada beberapa opsi penerjemahan data teks keluhan pada aplikasi. Data teks keluhan akan diterjemahkan menggunakan data dan algoritma *default* pada aplikasi jika *user* tidak mengajari aplikasi untuk penerjemahan data teks keluhan tertentu. Aplikasi akan mencari beberapa kata kunci atau frase kunci pada teks keluhan beserta lokasi kemunculannya pada teks keluhan untuk mencari tahu terjemahan teks keluhan yang paling tepat. Aplikasi juga mencari kata-kata kunci tertentu pada teks keluhan untuk mengetahui waktu kemunculan, frekuensi, dan sifat keluhan. Jika teks keluhan mengandung tanda koma (,) atau kata "dan", aplikasi akan mencari tahu apakah tanda koma dan kata "dan" tersebut memisahkan keluhan yang berbeda untuk nantinya diterjemahkan secara terpisah atau tidak.

User dapat mengajarkan aplikasi untuk menerjemahkan keluhan baru dengan tiga metode, menggunakan contoh, menggunakan kata-kata atau frase-frase kunci, dan menggunakan statistik. Pada pengajaran menggunakan contoh, *user* memberi contoh keluhan dan terjemahannya yang tepat, kemudian kata-kata pada contoh keluhan ini akan dikonversi menjadi sinonimnya yang lebih umum. Aplikasi hanya akan menerjemahkan teks keluhan yang 100% sama persis dengan contoh keluhan dari hasil pengajaran tersebut, misalnya contoh keluhan pada pengajaran adalah "sakit telinga" dengan terjemahannya adalah X, maka hanya teks masukan keluhan "sakit telinga" atau "sakit kuping" yang akan diterjemahkan menjadi X.

Pada pengajaran menggunakan kata-kata atau frase-frase kunci, *user* memasukkan dua atau lebih kata atau frase kunci yang harus ada pada sebuah teks keluhan agar menghasilkan terjemahan tertentu. Misalnya *user* mengajarkan aplikasi bahwa jika ditemukan kata kunci "muntah" dan "darah" maka akan menghasilkan terjemahan X, sehingga teks masukan keluhan "muntah mengeluarkan darah" dan "darah keluar saat muntah" akan menghasilkan terjemahan X yang sama. *User* juga dapat menambahkan aturan apakah urutan kemunculan kata-kata atau frase-frase kunci tersebut harus diperhatikan oleh aplikasi atau tidak, karena dua atau lebih kata atau frase yang muncul dengan urutan yang berbeda dapat memiliki arti yang berbeda, misalnya teks masukan "flu burung" memiliki arti berbeda dengan teks masukan "burung flu".

Pengajaran menggunakan statistik hampir sama dengan pengajaran menggunakan contoh, *user* memberikan contoh keluhan dan terjemahannya, kemudian contoh keluhan akan dikonversi ke sinonimnya yang lebih umum, tapi nantinya teks masukan keluhan tidak harus 100% sama dengan contoh keluhan dari pembelajaran. Teks masukan keluhan hanya cukup memiliki jumlah kata yang sama dengan contoh keluhan, memiliki seluruh kata yang ada pada contoh keluhan (dimana kata-kata pada teks masukan keluhan dan contoh keluhan sudah dikonversi ke sinonimnya), dan memiliki bigram berjumlah tertentu yang sama dengan bigram dengan probabilitas kemunculan tertinggi pada data contoh keluhan, untuk diterjemahkan menjadi terjemahan dari hasil pembelajaran. Jumlah bigram pada teks masukan keluhan yang harus sama dengan contoh keluhan berjumlah setengah dari jumlah kata pada contoh keluhan. Jumlah ini penulis pilih karena dari menganalisis ribuan keluhan pasien, probabilitasnya sangat kecil dua teks keluhan berbeda memiliki jumlah kata sama, kata-kata yang sama setelah dikonversi ke sinonimnya, dan bigram dengan jumlah tersebut yang sama. Pencarian jumlah kemiripan bigram yang terlalu sedikit akan menyebabkan tingkat kesalahan penerjemahan semakin tinggi, dan pencarian jumlah kemiripan bigram yang terlalu banyak menyebabkan jumlah data yang dibutuhkan untuk pengajaran semakin besar, sehingga penulis memilih pencarian jumlah kemiripan bigram yang tidak terlalu banyak dan tidak terlalu sedikit yaitu setengah dari jumlah seluruh kata pada contoh keluhan. Jika ada lebih dari satu alternatif penerjemahan maka dipilih terjemahan dari contoh keluhan dengan jumlah bigram yang sama dengan teks masukan keluhan yang paling banyak. Dengan menggunakan pengajaran metode ini, teks masukan "batuk berdarah setiap hari" memiliki terjemahan yang sama dengan teks masukan "setiap hari batuk berdarah".

Urutan penerjemahan teks masukan keluhan adalah sebagai berikut, dimulai dari nomor terkecil ke nomor terbesar, jika penerjemahan pada langkah dengan nomor tertentu gagal atau tidak ada maka dilanjutkan proses penerjemahan pada langkah dengan nomor yang lebih tinggi:

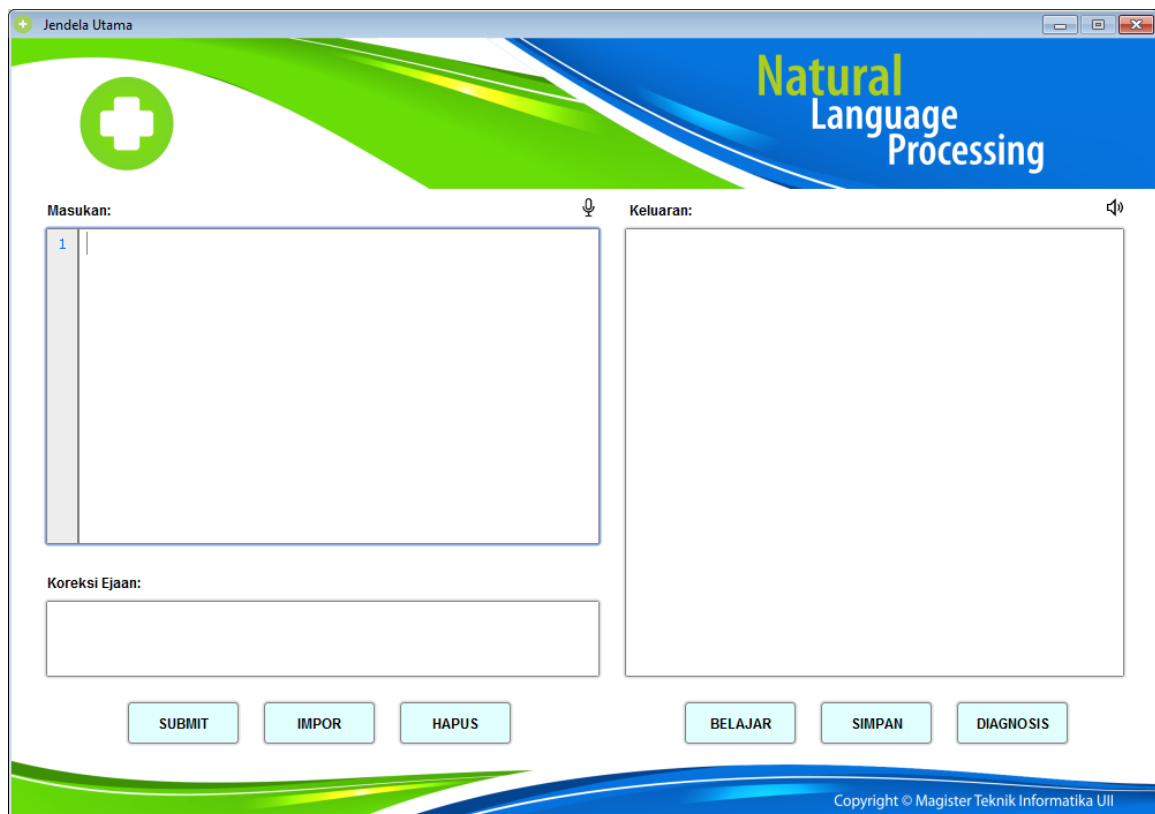
1. Penerjemahan menggunakan hasil pengajaran menggunakan contoh
2. Penerjemahan menggunakan hasil pengajaran menggunakan statistik
3. Penerjemahan menggunakan hasil pengajaran menggunakan kata atau frase kunci
4. Penerjemahkan menggunakan data dan algoritma *default* pada aplikasi.

g. Menampilkan Hasil Penerjemahan

Setelah didapatkan hasil penerjemahan, maka hasil penerjemahan akan ditampilkan kepada *user*. Hasil penerjemahan ini juga dapat disimpan ke dalam format teks, *excel*, atau diubah ke bentuk rekaman suara berformat WAV. Dari hasil penerjemahan ini aplikasi juga akan mencoba membantu memberikan diagnosis yang paling tepat menurut daftar keluhan pasien. Data diagnosis yang paling tepat berdasarkan daftar keluhan pasien ini dapat diisi sendiri oleh dokter. Karena fitur bantuan diagnosis ini hanya fitur tambahan dan bukan fitur utama dari aplikasi ini, maka tidak ada data *default* untuk fitur diagnosis ini.

3.2. IMPLEMENTASI DAN PENGUJIAN

Tampilan antarmuka aplikasi didesain sesederhana mungkin untuk memudahkan *user*. Gambar 2 berikut ini adalah tampilan antarmuka aplikasi yang sudah dikembangkan:

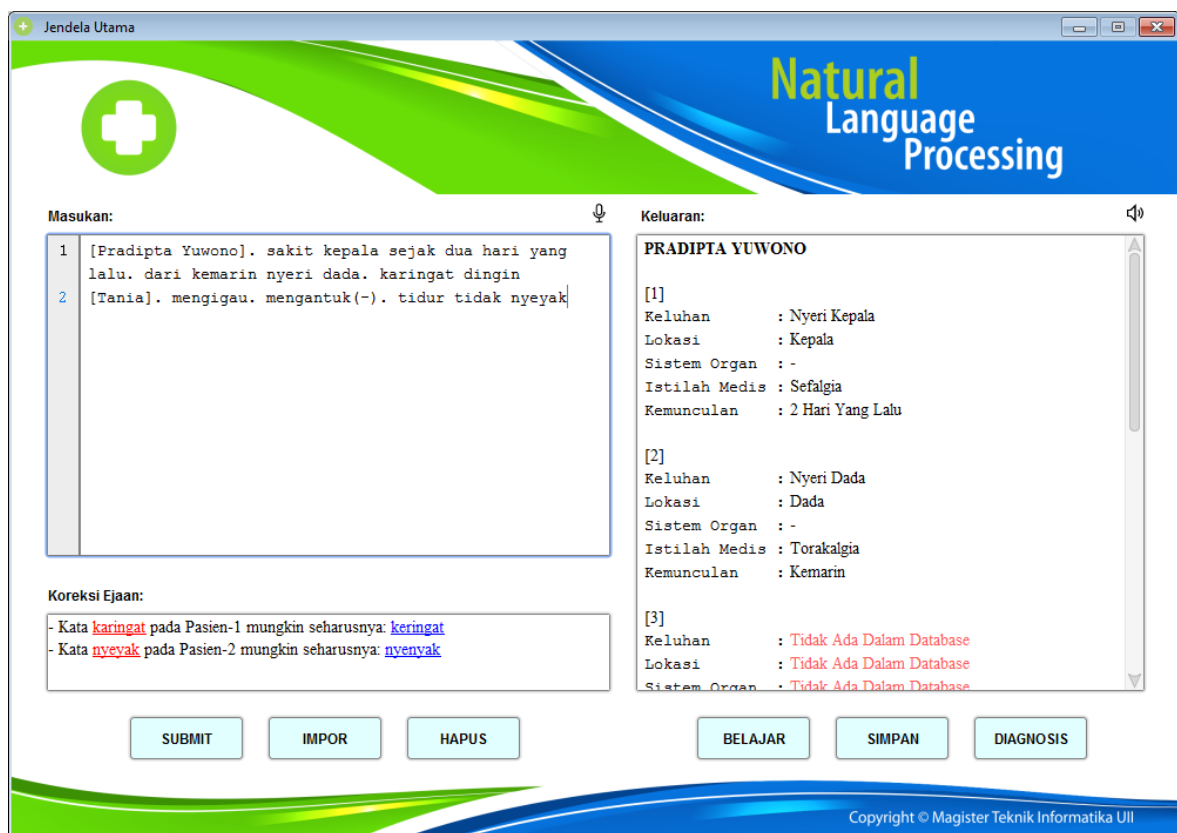


Gambar 2. Tampilan Antarmuka Aplikasi

Di sebelah kiri atas antarmuka aplikasi adalah *Text Area* Masukan tempat *user* memasukkan *input*, di sebelah kanan *Text Area* Masukan adalah *Text Area* Keluaran yang akan menampilkan hasil penerjemahan, dan di sebelah bawah *Text Area* Masukan adalah *Text Area* Koreksi Ejaan tempat aplikasi menampilkan peringatan kesalahan penulisan pada *Text Area* Masukan. Di sebelah kanan atas *Text Area* Masukan ada tombol bergambar mikrofon untuk memasukkan *input* berupa suara menggunakan mikrofon, di sebelah kanan atas *Text Area* Keluaran terdapat tombol bergambar *speaker* untuk mendengarkan pembacaan keluaran oleh komputer dalam bentuk suara, dua fitur ini bermanfaat bagi *user* yang memiliki disabilitas tertentu, misalnya memiliki penglihatan kurang bagus atau tidak mampu mengetik pada komputer. Di paling bawah antarmuka aplikasi ada enam tombol, Tombol

"Submit", "Impor", "Hapus", "Belajar", "Simpan", dan "Diagnosis". Tombol "Submit" digunakan untuk menerjemahkan teks masukan pada *Text Area* Masukan dan menampilkan hasil penerjemahannya pada *Text Area* Keluaran, serta menampilkan hasil koreksi ejaan pada *Text Area* Koreksi Ejaan jika ada kesalahan penulisan kata pada *Text Area* Masukan. Tombol "Impor" digunakan untuk mengimpor *file* teks, *file* audio WAV, atau *file* audio MP3 untuk dikonversi menjadi teks pada *Text Area* Masukan. Tombol "Hapus" digunakan untuk membersihkan *Text Area* Masukan, *Text Area* Keluaran, dan *Text Area* Koreksi Ejaan. Tombol "Belajar" digunakan untuk mengajarkan penerjemahan keluhan baru pada aplikasi. Tombol "Simpan" digunakan untuk menyimpan data keluaran dalam bentuk *file* teks, *excel*, atau rekaman audio berformat WAV. Tombol "Diagnosis" digunakan untuk menampilkan bantuan diagnosis oleh aplikasi berdasarkan daftar keluhan pasien, ini hanya fitur tambahan pada aplikasi dan data-data diagnosis berdasarkan daftar keluhan pasien dapat diisi sendiri oleh dokter jika dibutuhkan.

Gambar 3 berikut ini adalah contoh tampilan antarmuka ketika *user* memasukkan teks pada *Text Area* Masukan dan menekan tombol "Submit":



Gambar 3. Tampilan Antarmuka Aplikasi Setelah Men-Submit Teks Masukan

Data masukan untuk tiap pasien berbeda dipisahkan oleh baris baru, dan data masukan untuk tiap keluhan berbeda dipisahkan oleh tanda titik (.). *User* dapat memasukkan nama atau identitas unik untuk tiap pasien dengan format tertentu, yaitu harus ditulis di awal dan diapit oleh tanda kurung siku. *Text Area* Keluaran akan menampilkan terjemahan berupa informasi-informasi yang berhasil diperoleh dari teks masukan seperti nama keluhan yang dimaksudkan oleh pasien, lokasi keluhan, kemungkinan sistem organ dimana keluhan terjadi, istilah medis keluhan yang baku, waktu kemunculan keluhan, frekuensi keluhan, dan sifat keluhan (apakah ringan, parah, berbahaya, dll.). Jika sistem organ atau lokasi keluhan tidak jelas maka akan ditulis dengan tanda strip (-) dan jika teks masukan tidak memasukkan data waktu kemunculan keluhan, frekuensi keluhan, maupun sifat keluhan, maka tidak akan ditampilkan pada *Text Area* Keluaran. Pada contoh gambar di atas, tidak ada data frekuensi keluhan maupun sifat keluhan sehingga tidak ditampilkan pada *Text Area* Keluaran. Jika teks keluhan masukan tidak berhasil diterjemahkan ke bentuk teks keluaran yang terstruktur maka akan tertulis "Tidak Ada Dalam Database" pada *Text Area* Keluaran, pada Gambar 3 di atas terdapat kesalahan tulis

"karingat" dimana seharusnya ditulis "keringat" sehingga aplikasi gagal menampilkan hasil terjemahannya pada *Text Area* Keluaran.

Pada *Text Area* Koreksi Ejaan ditampilkan peringatan kata-kata yang menurut aplikasi kemungkinan salah penulisannya dan juga ditampilkan rekomendasi penulisan kata yang benar menurut aplikasi. Pada contoh Gambar 3 di atas, aplikasi menampilkan peringatan bahwa kata "karingat" dan "nyeyak" pada teks masukan kemungkinan seharusnya adalah "keringat" dan "nyenyak". Jika *user* mengklik kata-kata rekomendasi yang berwarna biru pada *Text Area* Koreksi Ejaan tersebut, maka kata-kata pada *Text Area* Masukan yang salah tulis tersebut secara otomatis akan diperbaiki.

User dapat memilih metode konversi masukan suara menjadi teks dengan menggunakan Google Cloud Speech atau dengan menggunakan Sphinx4. Penulis melakukan uji coba tingkat akurasi pengenalan suara menggunakan Google Cloud Speech pada dua jenis suara manusia berbeda sebanyak 100 kali. Dari hasil pengujian tersebut didapatkan bahwa tingkat akurasi pengenalan suara menggunakan Google Cloud Speech sebesar 92%. Penulis juga melakukan uji coba tingkat akurasi pengenalan suara menggunakan Sphinx pada dua jenis suara manusia berbeda sebanyak 100 kali. Dari hasil pengujian tersebut didapatkan bahwa tingkat akurasi pengenalan suara menggunakan Sphinx4 hanya sebesar 80% jika diuji menggunakan masukan dengan jumlah kata bebas, dan tingkat akurasinya meningkat menjadi 90% jika diuji menggunakan masukan dengan jumlah kata yang tidak lebih dari lima, sehingga metode pengenalan suara ini lebih cocok digunakan untuk pengenalan suara keluhan yang diucapkan secara singkat. Hal tersebut disebabkan karena fitur pengenalan suara menggunakan Sphinx4 hanya dilatih untuk mengenali kalimat dengan jumlah kata yang tidak lebih dari lima. Pengenalan suara dengan Sphinx4 juga hanya dapat mengenali 500 kata saja yang penulis pilih berdasarkan kata-kata yang paling umum digunakan pada keluhan pasien dan hanya dilatih serta diujicoba menggunakan dua macam suara manusia saja. *User* nantinya dapat memberikan pelatihan kalimat-kalimat, kata-kata, atau suara-suara baru pada Sphinx4 jika ingin meningkatkan akurasi pengenalan suara 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 CMUSphinx," n.d.). Karena keterbatasan waktu dan biaya, maka penulis hanya dapat melatih fitur pengenalan suara menggunakan Sphinx4 dengan data yang terbatas. Sejauh ini hasil pelatihan pengenalan suara menggunakan Sphinx4 sudah cukup baik menurut hasil uji coba penulis. Perbandingan pengenalan suara menggunakan Google Cloud Speech dan Sphinx4 dapat dilihat pada Tabel 1 di bawah ini.

Tabel 1. Perbandingan Pengenalan Suara Menggunakan Google Cloud Speech dan Sphinx4

Google Cloud Speech	Sphinx4
Harus terhubung secara <i>online</i> untuk menggunakannya	Dapat digunakan secara <i>offline</i>
Harus mendaftarkan akun Google untuk menggunakannya	Dapat digunakan tanpa perlu pendaftaran akun apapun
Kode program bersifat <i>closed source</i>	Kode program bersifat <i>open source</i>
Untuk penggunaan secara gratis, ada batasan durasi suara yang ditetapkan oleh Google	Gratis dan tidak ada batasan durasi suara dalam penggunaan
Mampu mengenali suara dengan jumlah kata sebanyak apapun dengan tingkat akurasi sangat tinggi	Hanya memiliki tingkat akurasi sangat tinggi untuk pengenalan suara dengan jumlah kata tidak lebih dari lima, namun dapat dilatih oleh <i>user</i> untuk dapat mengenali kalimat yang lebih panjang dengan tingkat akurasi yang tinggi
Mampu mengenali hampir semua kata dalam Bahasa Indonesia	Hanya mampu mengenali 500 kata yang paling umum digunakan untuk penulisan keluhan pasien, namun dapat dilatih oleh <i>user</i> untuk mengenali kata-kata baru.
Mampu mengenali berbagai macam jenis suara manusia.	Sejauh ini hanya dilatih dan diujicoba menggunakan dua jenis suara manusia saja, namun dapat dilatih oleh <i>user</i> untuk mengenali jenis suara manusia lainnya.

Hasil pengoreksian kesalahan penulisan pada teks masukan setelah diujicoba menggunakan 150 kata berbeda secara acak menunjukkan tingkat akurasi sebesar 80,67%. 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.

Hasil penerjemahan keluhan pasien menggunakan data *default* tanpa pembelajaran setelah diujicoba sebanyak 100 kali menggunakan teks masukan keluhan pasien secara acak menunjukkan tingkat akurasi sebesar 90%. Tingkat akurasi diharapkan dapat ditingkatkan oleh dokter dengan memberikan data-data pembelajaran baru untuk penerjemahan pada aplikasi. Pembelajaran penerjemahan oleh aplikasi berdasarkan uji coba oleh penulis menunjukkan hasil yang memuaskan. Kesalahan penerjemahan umumnya terjadi karena teks masukan tidak tertulis menggunakan kata maupun tata bahasa Bahasa Indonesia yang baku serta karena kurang lengkapnya keluhan pasien yang dapat dikenali oleh aplikasi.

4. KESIMPULAN

Dari hasil pengujian aplikasi, penulis menyimpulkan bahwa proses penerjemahan keluhan pasien pada aplikasi yang dikembangkan ini sudah memberikan hasil yang cukup baik. Kesulitan utama yang dihadapi dalam pengembangan sistem ini adalah banyaknya kemungkinan penulisan teks keluhan pasien termasuk penggunaan kata-kata maupun tata bahasa yang tidak baku yang ada pada teks masukan. Dengan fitur pembelajaran yang ada pada aplikasi ini, diharapkan semakin lama tingkat akurasi penerjemahan oleh aplikasi menjadi semakin tinggi dengan ditambahkannya data-data pembelajaran baru. Penulis juga akan mempelajari daftar kata-kata tidak baku dan pola tata bahasa tidak baku lainnya yang biasa digunakan dalam penulisan teks keluhan pasien untuk pengembangan aplikasi di penelitian yang akan datang.

Fitur pengenalan suara pada aplikasi sudah menunjukkan tingkat akurasi yang cukup baik menggunakan Google Cloud Speech. Untuk pengenalan suara menggunakan Sphinx4 terdapat beberapa kekurangan seperti kurang akuratnya pengenalan suara pada kalimat yang panjang, terbatasnya jumlah kata yang dikenali, dan sedikitnya jumlah suara yang digunakan untuk pelatihan pengenalan suara. Diharapkan pada penelitian yang akan datang pelatihan kalimat-kalimat, kata-kata, dan suara-suara baru dapat meningkatkan akurasi pengenalan suara pada Sphinx4. Karena *source code* Sphinx4 bersifat *open source*, kemungkinan *source code* pada Sphinx4 juga dapat sedikit dimodifikasi agar dapat mengenali suara dalam Bahasa Indonesia dengan lebih baik, karena Sphinx4 lebih fokus dikembangkan untuk pengenalan suara dalam Bahasa Inggris.

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. Hal ini dapat diperbaiki dengan pencarian kata rekomendasi selain berdasarkan probabilitas kemunculannya dalam teks Bahasa Indonesia juga dengan melihat probabilitas kemunculan kata sebelum maupun sesudahnya. Sehingga dengan perbaikan tersebut, teks "semakan nasi" akan dikoreksi menjadi "memakan nasi" bukan "semakin nasi", karena kata "nasi" memiliki probabilitas lebih tinggi ditulis setelah kata "memakan" ketimbang setelah kata "semakin".

5. DAFTAR PUSTAKA

- Ajami, S., & Bagheri-Tadi, T. (2013). *Barriers for Adopting Electronic Health Records (EHRs) by Physicians*. *Acta Informatica Medica*, 21(2), 129–134.
- Ford, E., Carroll, J. A., Smith, H. E., Scott, D. & 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*.
- Gruhn, R., Minker, W., & Nakamura, S. (2011). *Statistical pronunciation modeling for non-native speech processing*. New York - Heidelberg: Springer.

- Markum H.M.S. (2007). *Penuntun Anamnesis dan Pemeriksaan Fisis*. Jakarta: Pusat Penerbitan Departemen Ilmu Penyakit Dalam Fakultas Kedokteran Universitas Indonesia.
- 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.
- PERMENKES. (2008). *Peraturan Menteri Kesehatan Republik Indonesia Nomor 269/MENKES/PER/III/2008 Tentang Rekam Medis Pasal 3*. Indonesia.
- Training an acoustic model for CMUSphinx*. (n.d.) Retrieved September 30, 2017, from <https://cmusphinx.github.io/wiki/tutorialam/>