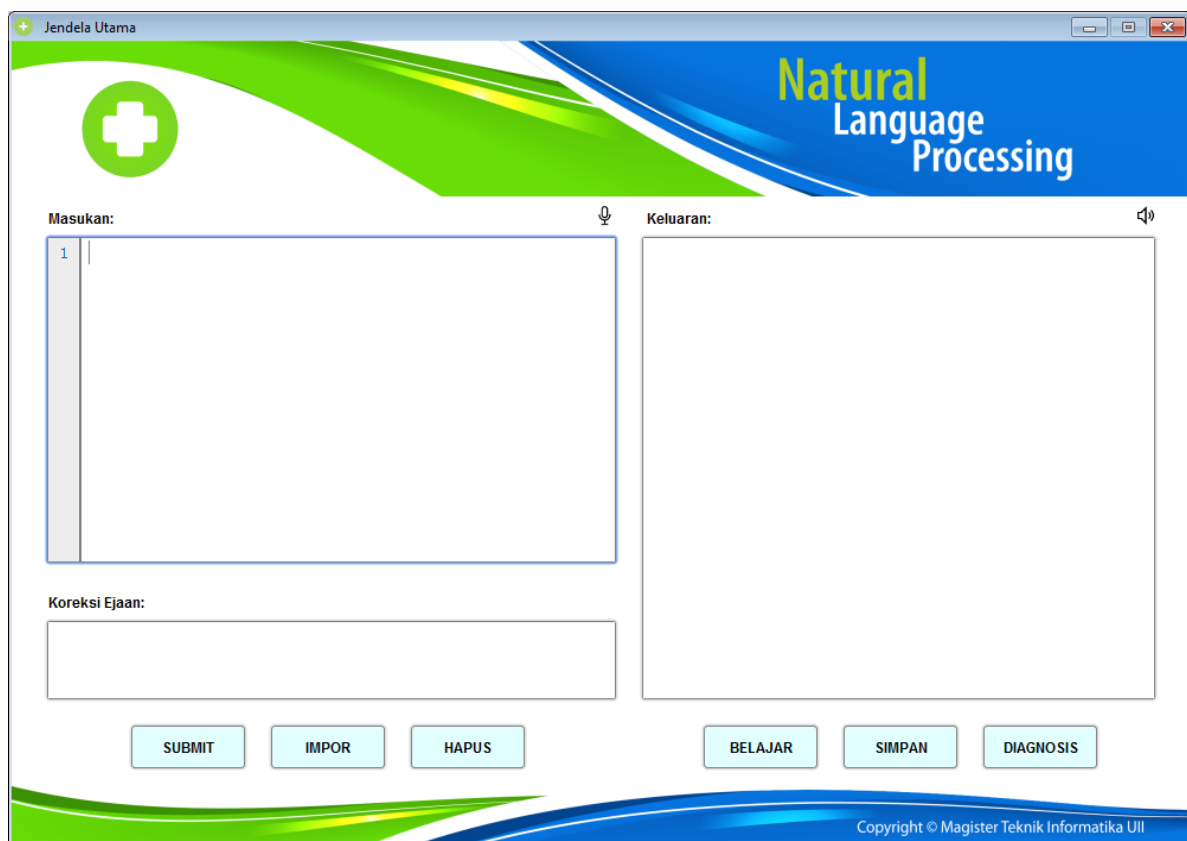


BAB V

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:

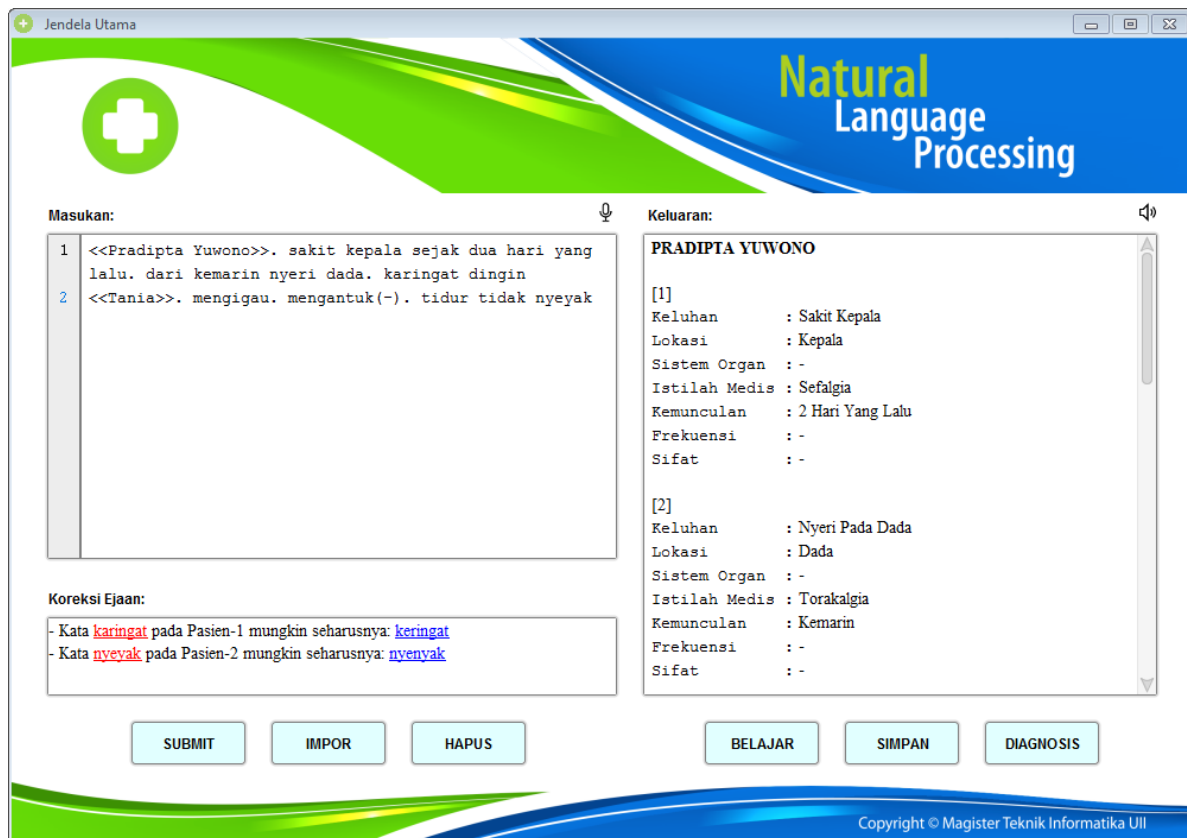


Gambar 5.1. Tampilan antarmuka jendela utama aplikasi.

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:

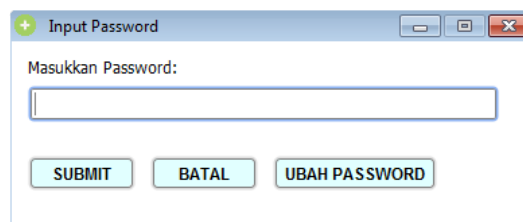


Gambar 5.2. Tampilan antarmuka jendela utama aplikasi setelah *user* men-submit teks masukan.

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 bawah nya 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

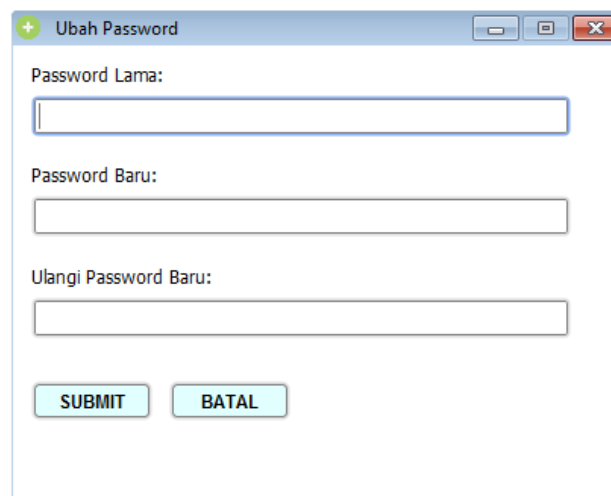
jika aplikasi menganggap *user* salah menuliskan kata tertentu pada *Text Area* Masukan, dimana jika rekomendasi penulisan kata tersebut diklik maka seluruh kata yang dianggap salah penulisannya pada *Text Area* Masukan akan secara otomatis diperbaiki menjadi sesuai dengan rekomendasi penulisan kata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



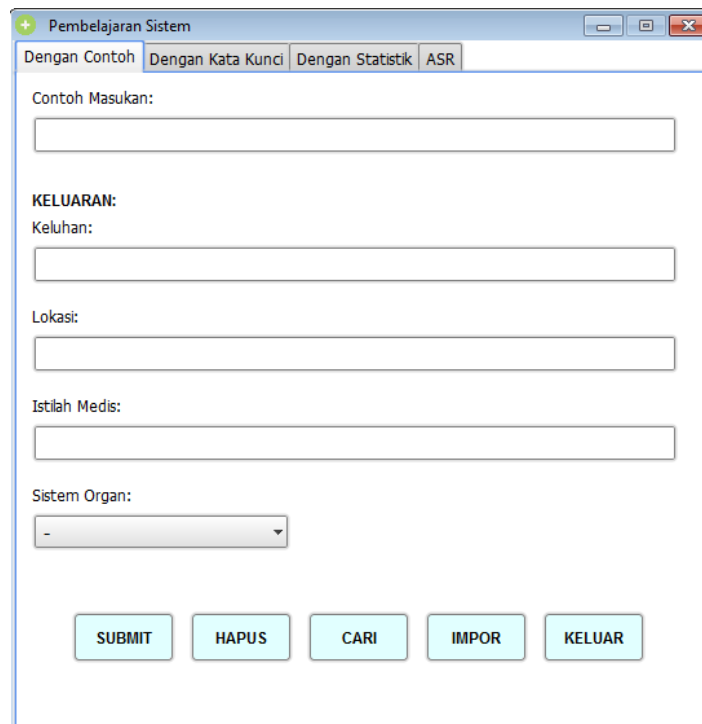
Gambar 5.3. Tampilan antarmuka untuk memasukkan *password*.

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.



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

- A text input field labeled "Contoh Masukan:".
- A section labeled "KELUARAN:" containing three more text input fields:
 - "Keluhan:"
 - "Lokasi:"
 - "Istilah Medis:"
- A dropdown menu labeled "Sistem Organ:" with a "-" symbol.
- Five buttons at the bottom: "SUBMIT", "HAPUS", "CARI", "IMPOR", and "KELUAR".

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.

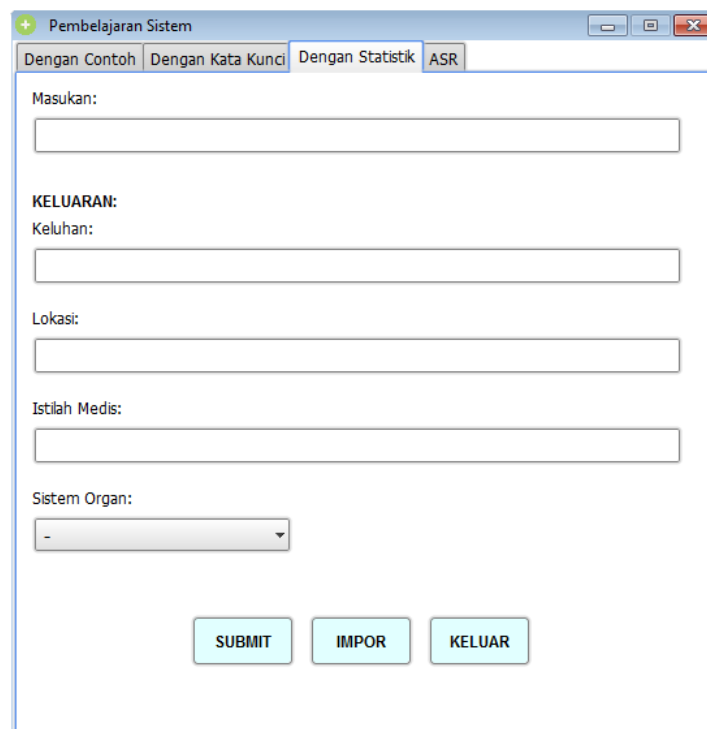
User dapat mengganti mode pembelajaran aplikasi dengan mengklik *tab* yang ada di atas. Gambar 5.7. di bawah ini menunjukkan jendela untuk mengajari sistem penerjemahan berdasarkan kata/frasa kunci. *User* tuliskan 2 atau 3 kata atau frasa kunci dimana kata atau frasa kunci ketiga bersifat opsional, 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.

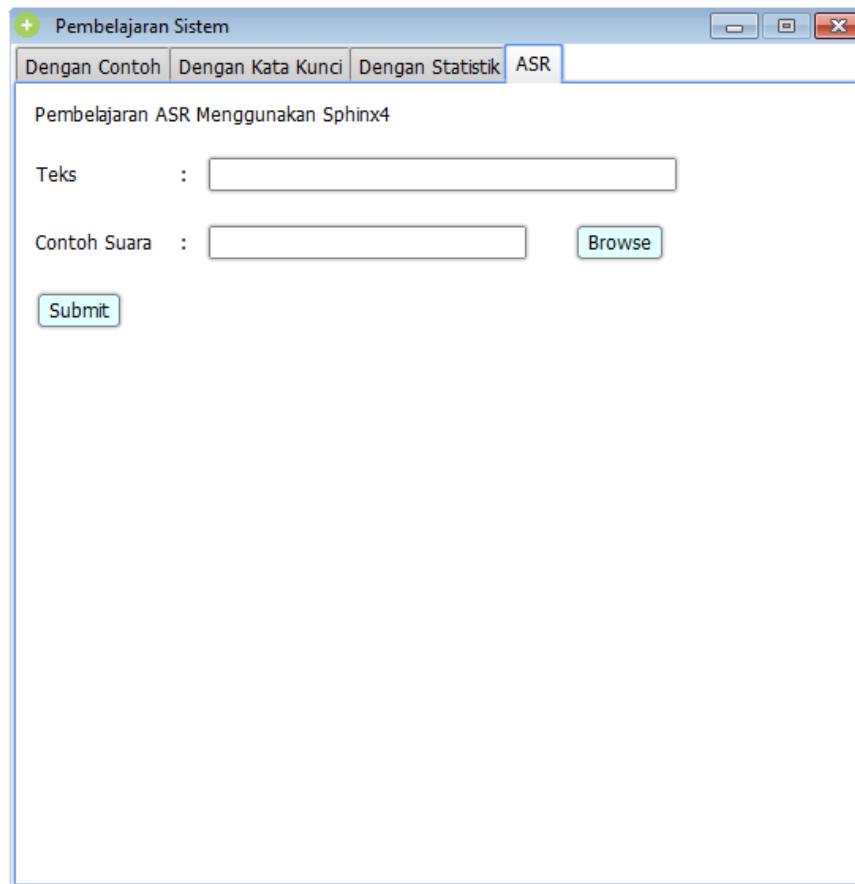
Gambar 5.9. di bawah ini menunjukkan jendela untuk mengajari sistem penerjemahan berdasarkan data statistik. Sama seperti pembelajaran berdasarkan contoh, *user* tuliskan contoh masukan keluhan, nama keluhan, lokasi keluhan, dan istilah medis keluhan pada *Text Area* yang sesuai, serta pilih sistem organ 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 software window titled "Pembelajaran Sistem". It has four tabs: "Dengan Contoh", "Dengan Kata Kunci", "Dengan Statistik" (which is selected), and "ASR". The main area contains several input fields: "Masukan:" with a text box, "KELUARAN: Keluhan:" with a text box, "Lokasi:" with a text box, and "Istilah Medis:" with a text box. Below these is a "Sistem Organ:" dropdown menu showing a minus sign. At the bottom of the window are three buttons: "SUBMIT", "IMPOR", and "KELUAR".

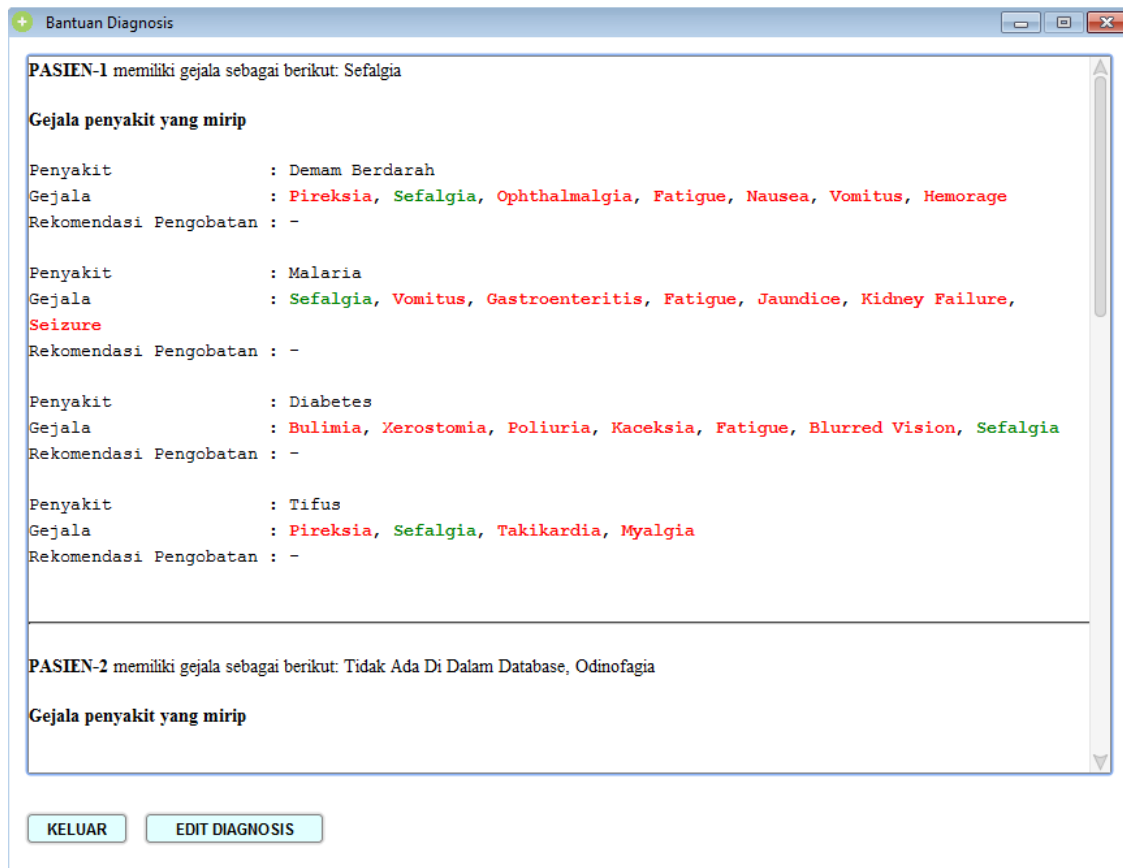
Gambar 5.9. Tampilan antarmuka untuk pembelajaran berdasarkan statistik.

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

1 Keluhan	
Daftar Keluhan	Waktu Pemrosesan (detik)
batuk berdahak	5,38
pusing tiap hari	5,37
sakit tenggorokan	5,4
kepala panas	5,45
tubuh pegal-pegal	5,47
Rata-Rata Waktu Pemrosesan 1 Keluhan: 5,41 detik	
2 Keluhan	
Daftar Keluhan	Waktu Pemrosesan (detik)
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
Rata-Rata Waktu Pemrosesan 2 Keluhan: 5,56 detik	
3 Keluhan	
Daftar Keluhan	Waktu Pemrosesan (detik)

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
Rata-Rata Waktu Pemrosesan 3 Keluhan: 5,58 detik	
5 Keluhan	
Daftar Keluhan	Waktu Pemrosesan (detik)
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
Rata-Rata Waktu Pemrosesan 5 Keluhan: 5,55 detik	
10 Keluhan	
Daftar Keluhan	Waktu Pemrosesan (detik)
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
Rata-Rata Waktu Pemrosesan 10 Keluhan: 5,83 detik	

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

Rata-Rata Waktu Pemrosesan 1 Kata: 11,17 detik	
2 Kata	
Daftar Kata	Waktu Pemrosesan (detik)
sakit kepala	10,76
gagal ginjal	15,25
kepala pusing	14,5
batuk batuk	15,34
hidung pilek	23,78
Rata-Rata Waktu Pemrosesan 2 Kata: 15,93 detik	
3 Kata	
Daftar Kata	Waktu Pemrosesan (detik)
pegal tadi pagi	37,91
otot sangat sakit	44,25
sakit ginjal parah	31,51
kaki tidak bertenaga	33,46
mata sulit melihat	46
Rata-Rata Waktu Pemrosesan 3 Kata: 38,63 detik	
4 Kata	
Daftar Kata	Waktu Pemrosesan (detik)
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
Rata-Rata Waktu Pemrosesan 4 Kata: 38,42 detik	
5 Kata	
Daftar Kata	Waktu Pemrosesan (detik)
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

	dr. Nadia	dr. Setya	dr. Andam	dr. Ari	dr. Novi
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

Nama Dokter	Saran-Saran
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.