IMPLEMENTASI PENDIKTEAN BAHASA INDONESIA

Hari Bagus Firdaus dan Ayu Purwarianti

Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, Jalan Ganesha 10, Bandung, 40132, Indonesia

E-mail: hari.firdaus@gmail.com

Abstrak

Paper ini memaparkan hasil penelitian dalam membangun aplikasi pendiktean Bahasa Indonesia untuk waktu nyata. Dalam membangun sebuah aplikasi pendiktean, terdapat beberapa masalah seperti perintah suara (*voice command*), *Out Of Vocabulary* (OOV), *noise,* dan *filler*. Adapun yang menjadi fokus dalam penelitian ini adalah penanganan perintah suara dan OOV dari kata yang didiktekan. Pendiktean suara merupakan pengembangan lanjut dari pengenalan suara secara waktu nyata dengan tambahan metode untuk menangani hal-hal yang telah dinyatakan sebelumnya. Untuk menangani perintah suara, sebuah modul ditambahkan untuk mengecek hasil *decoding* dari sistem pengenalan suara. Adapun untuk menangani OOV, ditambahkan modul penanganan pengejaan setelah sebelumnya dinyatakan status ejaan. Model perintah suara dan model huruf ditambahkan ke dalam kamus dan digunakan sebagai pelatihan dari model bahasa n-gram. Dalam pengujian, dilakukan evaluasi terhadap sistem pengenalan suara, penanganan perintah suara, dan modul pengejaan sebagai strategi untuk menangani kata OOV. Untuk modul pengenalan suara, akurasi yang dicapai adalah 70%. Untuk modul penanganan perintah suara, pengujian menunjukkan bahwa perintah suara dapat ditangani dengan baik. Sedangkan untuk modul pengejaan, pengujian menunjukkan bahwa hanya 20 dari 26 huruf yang berhasil dikenali.

Kata Kunci: *aplikasi pendiktean, pengejaan kata, pengenalan suara bahasa indonesia, perintah suara*

Abstract

In this paper, we presented the results of research in building applications dictation of the Bahasa Indonesia for real-time. In developing a dictation application, there are some problems such as voice command, Out of Vocabulary (OOV), noise, and filler. As the focus in this research is the handling of voice command and OOV from dictated words. Voice dictation is a further development of real time voice recognition with an additional method to deal with things that have been stated before. To handle voice commands, a module is added to check the results of decoding of the voice recognition system. To handle OOV, spelling handling module is added after the previously stated spelling status. Voice command model and the model letter are added to the dictionary and used as the training of n-gram language model. In testing, we conducted an evaluation of speech recognition systems, voice commands and spelling handling module as a strategy to deal with OOV words. For the speech recognition module, the achieved accuracy is 70%. For voice commands handling module, the test showed that voice commands can be handled properly. As for the spelling module, testing showed that only 20 of the 26 letters that successfully recognized.

Keywords: *dictation application, indonesian speech recognition, spelling words, voice commands*

1. Pendahuluan

Aplikasi Pengenalan Suara Otomatis atau *Automatic Speech Recognition* (ASR) telah dikembangkan sejak tahun 1930 dari sejak mengenali suara yang sederhana hingga ucapan manusia yang kontinu. Aplikasi pengenalan suara otomatis ini selanjutnya digunakan dalam berbagai bidang kehidupan seperti pemasukan data, sistem dialog, perintah suara, dan seterusnya.

Salah satu aplikasi dari ASR adalah pendiktean. Sistem pendiktean konvensional dapat digambarkan sebagai sebuah proses di mana seseorang mendiktekan sesuatu yang kemudian direkam pada suatu bentuk rekaman, dan kemudian seseorang lain mencoba mengubahnya dalam bentuk teks dengan mendengarkan rekamannya tadi. Kedua proses dilakukan secara terpisah dengan alat bantu yang berbeda. Sistem seperti ini sering digunakan oleh pengacara, petugas medis, ataupun peneliti.

Adapun pendiktean digital dapat didefinisikan sebagai proses pendiktean yang lebih lanjut di mana suara diterima dalam bentuk format audio tertentu dan langsung diubah ke dalam bentuk teks. Ide utama dari sistem ini adalah untuk menggunakan suara sebagai *input* dari ketikan yang akan memudahkan pemasukan *input* serta dapat digunakan sebagai alat bantu bagi orang dengan keterbatasan organ tubuh [1].

Dalam pengetahuan penulis, pengembangan aplikasi ASR untuk Bahasa Indonesia termasuk pendiktean suara, berkembang tidak secepat bahasa lainnya di dunia. Beberapa penelitian untuk pengenalan suara Bahasa Indonesia telah mencapai hasil yang cukup baik seperti *Indonesian* LVCSR dan *Indonesian* ASR untuk mendeteksi OOV [2-4]. Meskipun menggunakan *Indonesian* ASR yang umum sebagai sebuah sistem pendiktean waktu nyata adalah hal yang mungkin tapi terdapat beberapa kelemahan seperti risiko mengenali kata sebagai perintah suara dan penanganan kata OOV. Berdasarkan hal ini, penelitian difokuskan pada penggunaan sistem pengenalan suara Bahasa Indonesia sebagai dasar dari pembangunan sistem pendiktean waktu nyata Bahasa Indonesia yang mampu menangani masalah perintah suara dan kata OOV. Strategi untuk setiap masalah di atas akan dipaparkan pada bagian-bagian pada penelitian ini.

2. Metodologi

Pengenalan suara otomatis merupakan sebuah sistem pengenalan suara otomatis dirancang untuk mengenali rangkaian suara dan menghasilkan respons berdasarkan hasil pengenalan yang diperoleh. Sistem pengenalan suara yang modern yang mampu menangani suara kontinu, terdiri atas antarmuka untuk mengekstrak fitur dari sinyal suara, model akustik, model bahasa, kamus, dan *decoder* [5].

Pertama, ucapan masukan diterima oleh modul antarmuka seperti terlihat pada gambar 1. Pada modul antarmuka ini, proses ekstraksi fitur akan menghasilkan sekumpulan vektor fitur yang mewakili sinyal ucapan masukan. Seperti digunakan oleh berbagai sistem pengenalan suara pada umumnya, dalam penelitian ini, digunakan teknik *Mel Frequency Cepstral Coefficients* (MFCC) sebagai metode ekstraksi.

Vektor fitur kemudian di-*decoding* untuk menghasilkan rangkaian kata yang paling mirip. Proses *decoding* ini menggunakan model akustik, model bahasa, dan kamus fonetik untuk menghasilkan graf pencarian dari rangkaian kata yang dikeluarkan.

Pemodelan akustik bertujuan untuk menghitung probabilitas dari vektor fitur untuk rangkaian fonem. Dalam penelitian ini, digunakan *Hidden Markov Model* (HMM) untuk merepresentasikan rangkaian fonem. Model kata yang didefinisikan dalam kamus, dapat dikomposisi dengan menyambungkan berbagai model fonem yang terkait.

HMM memiliki dua elemen penting yaitu *hidden states* dan *observable feature vector*. Setiap *hidden state* memiliki dua nilai probabilitas yaitu *transition probability* yang menyatakan nilai probabilitas suatu status berubah ke status lainnya serta *emission probability* yang menyatakan nilai probabilitas sebuah fonem direpresentasikan oleh segmen vektor fitur tertentu.

Model akustik dapat dilatih dalam bentuk *monophone*, yang merepresentasikan fonem tunggal, atau *tied-state N-phone*. *Tied-state N-phone* adalah *context-dependent model* yang menggunakan fonem sebelum atau sesudahnya seperti *2-phone* (*biphone*) atau *3-phone* (*triphone*). Perbedaan format fonem yang digunakan untuk model akustik dapat dilihat pada tabel I.

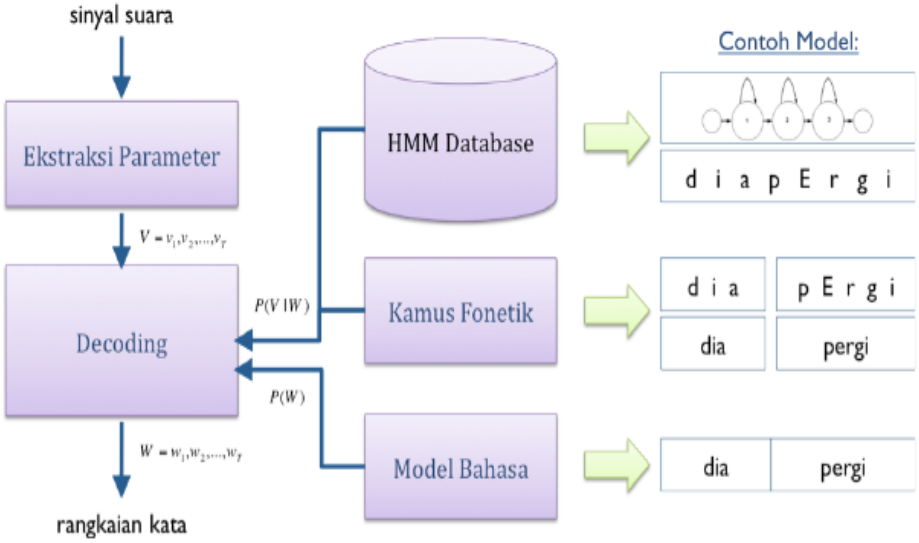
TABEL I
CONTOH TIPE PEMODELAN AKUSTIK

Monophone	s E n a ng
Left-biphone	s-E E-n n-a a-ng ng
Right-biphone	s s+E E+n n+a a+ng
Triphone	s-E s-E+n E-n+a n-a+ng a+ng

Fungsi model bahasa dalam pengenalan suara adalah untuk menghitung probabilitas hipotesis rangkaian kata. Kemudian nilai probabilitas ini bersamaan dengan nilai probabilitas dari model akustik akan membatasi ruang pencarian dari rangkaian kata dengan *maximum likelihood*.

Terdapat dua pendekatan utama untuk model bahasa yaitu berbasis aturan dengan menggunakan tata bahasa statik dan berbasis statistik yang menyediakan nilai probabilitas urutan kata. Model bahasa statistik yang umum digunakan adalah model n-gram. Pada model ini, kemunculan kata tergantung pada n-1 kata yang muncul sebelumnya (*priori likelihood*). Model bahasa n-gram memberikan nilai probabilitas, sebagai contoh, pasangan kata dalam *bigram* atau serangkaian tiga kata dalam *trigram*.

Aplikasi pendiktean merupakan aplikasi yang menggunakan kemampuan sistem pengenalan suara untuk menangani masalah-masalah pendiktean suara secara waktu nyata. Masalah yang dimaksud meliputi perintah suara, kata OOV, *filler*, dan *noise*. Perintah suara adalah bagaimana mengendalikan fungsi tertentu dari aplikasi pendiktean.



Gambar 1. Skema umum sistem pengenalan suara [5].

Kata OOV adalah kata-kata yang tidak terdapat pada kamus fonetik dan juga mungkin tidak terdapat pada korpus untuk model bahasa. *Filler* adalah suara yang berasal dari pengguna tapi bukan merupakan kata yang ingin dihasilkan, seperti batuk, hembusan nafas, dst. *Noise* adalah suara yang tidak berasal dari pengguna tapi masuk sebagai *input* bagi sistem pengenalan suara. Contoh *noise* adalah suara dari manusia lain yang berada pada ruangan yang sama, suara komputer, dst. Dalam penelitian ini, masalah yang diselesaikan adalah masalah perintah suara dan kata OOV di mana masalah *filler* dan *noise* membutuhkan teknik analisis model *filler* dan penghilangan *noise* yang lebih kompleks.

Dalam aplikasi pendiktean, perintah suara biasanya diperlukan untuk memperbaiki kata yang salah dikenali atau salah diucapkan oleh pengguna. Untuk menangani perintah suara, model perintah suara ditambahkan dalam kamus dan juga dalam model bahasa n-gram. Model perintah suara yang ditambahkan adalah semua kata dan semua frasa kata dari setiap perintah suara. Dengan seperti ini, sistem pengenalan suara akan mampu mengenali perintah suara baik dengan *interval short pause* ataupun tidak. Pengecekan hasil *decoding* dan aksi yang terkait dilakukan pada level aplikasi pendiktean.

Untuk menangani masalah kata OOV, digunakan strategi pengejaan untuk setiap kata yang dianggap tidak ada pada kamus. Dengan strategi ini, pengguna dapat memasukkan kata-kata baru meskipun dilakukan secara huruf per huruf. Model pengejaan yang ditambahkan adalah model untuk 26 huruf pada Bahasa Indonesia. Model ini ditambahkan pada kamus fonetik dan pada model bahasa n-gram.

3. Hasil dan Pembahasan

Untuk dapat mencapai tujuan pendiktean, sistem pengenalan suara harus mampu menangani ucapan yang kontinu, bersifat tidak tergantung pada pengguna (*speaker independent*) dan memiliki ukuran kamus yang cukup. Dalam mengimplementasikan sistem ini, korpus teks yang digunakan adalah korpus dari Kompas dan Tempo dengan format TREC. Korpus ini digunakan sebagai bahan untuk model bahasa dan kamus. Korpus teks diproses terlebih dahulu dengan *shell script* dalam Perl untuk menghilangkan *tag*, mengubah ke dalam bentuk *lower-case*, menghilangkan semua tanda baca, mengonversi angka/tanggal/waktu ke dalam bentuk pengucapannya (seperti “100” menjadi “seratus”), menghilangkan spasi dan tanda baris baru yang tidak diperlukan, menambah *tag* <sil> dan </sil> sebagai tanda awal dan akhir kalimat. Untuk memastikan kualitas perbaikan persiapan korpus, dilakukan pengujian manual untuk memperbaiki kata yang salah, menggabungkan kalimat pendek dan memecah kalimat panjang. Contoh dari korpus teks dapat dilihat pada gambar 2.

<sil> lima orang tewas dan sekitar seratus tiga puluh cedera serius </sil>
<sil> api bermula seetengah jam setelah awal milenium baru </sil>
<sil> … </sil>

Gambar 2. Contoh korpus teks.

Untuk membangun model bahasa dari korpus tersebut, digunakan CMU-Cambridge SLM *toolkit* untuk menghasilkan model bahasa 2-gram dan 3-gram. Untuk membangun kamusnya, Perl *shell script* dibuat untuk mengekstrak dan mengurutkan semua kata yang unik di mana transkripsi fonetiknya dilakukan secara manual.

Data untuk akustik model adalah data yang dibangun oleh Lestari, Dessi P., et al [2]. Dalam data ini, digunakan 300 kalimat dari 18 pembicara Bahasa Indonesia dalam lingkungan yang bersih, bebas dari *noise* maupun *filler*. Akustik model yang dibangun dengan menggunakan HTK 3.4.1 merupakan *triphone* berbasis fonem. Daftar semua fonem yang digunakan untuk model akustik dapat dilihat pada tabel II.

TABEL II
DAFTAR FONEM

Vowels	a, e, E, i, o, u
Diphtongs	ai, au, oi
Semi-vowels	w, y
Plosives	b, p, d, t, g, k, kh
Africatives	j, c
Fricatives	f, z, s, sy, h
Liquids	l, r
Nasals	m, n, ng, ny
Silence	sil, sp (*inter-word silence*)

Sebagai alat *decoding*, digunakan Julius versi 4.1.4 dengan konfigurasi *input* mikrofon waktu nyata [6]. Hasil detail dari implementasi sistem pengenalan suara dapat dilihat pada tabel III.

TABEL III
HASIL IMPLEMENTASI SISTEM PENGENALAN SUARA

Deskripsi	Ukuran
Korpus Teks	8,100 kalimat
Perplexity (LM)	134.93
Kamus	9,473 kata
OOV Rate (LM)	11.9%
HMM	*context-dependent triphone*
Decoder	Julius 4.1.4

Tujuan dari penelitian ini adalah untuk menambahkan komponen pada sistem pengenalan suara yang dapat menangani masalah pendiktean waktu nyata, dalam hal ini adalah perintah suara dan kata OOV, sehingga nilai akurasi dari sistem pengenalan suara tidak menjadi fokus penelitian. Perlu dicatat bahwa sumber data yang digunakan untuk penelitian ini berukuran lebih kecil daripada penelitian lainnya [2][4]. Menurut Lestari, Dessi P., et al. [2], ukuran korpus teks adalah 600,000 kalimat, dengan nilai *perplexity* adalah 87 dan ukuran kamusnya adalah 41,000 kata. Akurasi sistem pengenalan suara tertinggi yang diperoleh adalah 90%. Sedangkan menurut Sakti S., et al. [4], ukuran korpus teks adalah 160,000 kalimat, dengan nilai *perplexity* 67 dan ukuran kamus 40,000 kata. Akurasi dari sistem pengenalan suaranya mencapai 92%.

Dalam implementasi pendiktean suara, didefinisikan 18 perintah suara yang mencakup 32 kata dalam kamus fonetik. Seperti telah dinyatakan sebelumnya, perintah suara ini mencakup kata-kata yang menyusunnya serta frasa untuk setiap perintah suara tanpa ada *short pause* di antara kata. Selain itu, model perintah suara ini dilatih pada model bahasa 2-gram dan 3-gram. Adapun daftar perintah suara yang dibuat dapat dilihat pada tabel IV.

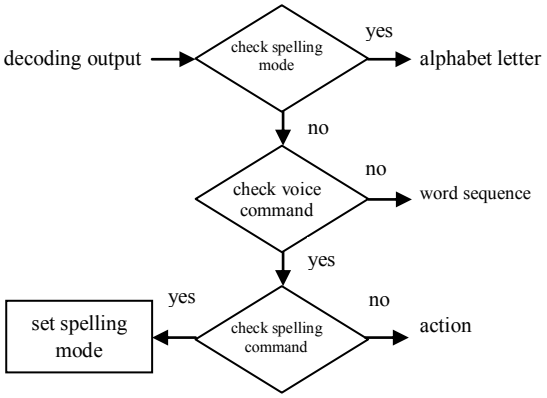
TABEL IV
DAFTAR PERINTAH SUARA

Perintah Suara	Deskripsi
Baris baru	Membuat baris baru (*new line*) pada editor
Koreksi	Menghapus kata terakhir
Koreksi semua	Menghapus semua teks yang ada di editor
Hapus	Menghapus kata terakhir
Hapus semua	Menghapus semua teks yang ada di editor
Awal baris	Memindahkan kursor ke awal baris
Akhir baris	Memindahkan kursor ke akhir baris
Seleksi	Menyeleksi kata terakhir
Seleksi semua	Menyeleksi semua teks yang ada di editor
Ubah kapital	Mengubah huruf pada kata terakhir atau kata yang diseleksi menjadi huruf kapital (*upper case*)
Mode eja	Masuk mode pengejaan (*spelling*)
Mode *spelling*	Masuk mode pengejaan (*spelling*)
Mode normal	Kembali ke mode normal untuk rekognisi kalimat biasa
Mode biasa	Kembali ke mode normal untuk rekognisi kalimat biasa
Tanda titik	Menambahkan tanda baca titik
Tanda koma	Menambahkan tanda baca koma
Tanda tanya	Menambahkan tanda baca tanda tanya
Tanda seru	Menambahkan tanda baca tanda seru

Untuk pengejaan, model huruf sebanyak 26 huruf dengan transkripsinya dimasukkan ke dalam kamus fonetik. Model huruf juga dilatih dalam kedua model bahasa seperti model perintah suara.

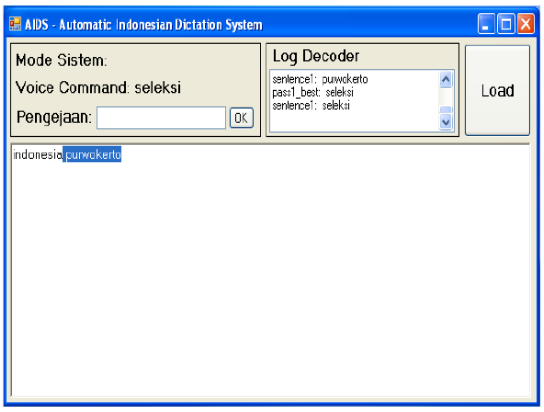
Alur proses dari aplikasi pendiktean dapat dilihat pada gambar 3. Hasil dari sistem pengenalan suara akan diterima dalam pengecekan modus pengejaan. Jika status sistem adalah modus pengejaan maka suara yang dikenali akan diperlakukan sebagai huruf. Jika tidak maka suara akan dicek lagi apakah merupakan perintah suara atau bukan. Jika bukan merupakan perintah suara, maka hasil dari pengenalan suara akan diterima sebagai kata yang didiktekan. Namun jika merupakan perintah suara, maka hasil dari pengenalan suara akan diterima sebagai perintah suara dan aplikasi akan melakukan aksi tertentu sesuai dengan perintah suara yang dimaksud. Dalam hal ini, jika perintah suara yang dimaksud adalah mengubah status

pengejaan maka aplikasi akan mengubah status pengejaan menjadi benar.



Gambar 3. Alur proses aplikasi pendiktean.

Adapun contoh antarmuka dari aplikasi pendiktean dapat dilihat pada gambar 4. Hasil dari sistem pengenalan suara terhadap masukan ucapan dari pengguna ditampilkan pada "*Log Decoder*" seperti terlihat pada gambar 4. Hasil ini kemudian diperiksa oleh aplikasi pendiktean untuk kemudian ditampilkan baik pada status "Mode Sistem","*Voice Command*", kotak teks pada "Pengejaan", atau kotak teks besar di bawah yang merupakan tempat dari kata yang didiktekan.



Gambar 4. Antarmuka aplikasi pendiktean.

Pengujian sistem pengenalan suara dilakukan dengan menggunakan ucapan baik dari *file* audio maupun dari *input* secara waktu nyata. Setiap pengujian terdiri atas 25 kalimat yang diucapkan oleh seorang laki-laki dan seorang perempuan dalam lingkungan dengan nilai *noise* yang kecil dan tanpa *filler*. Beberapa kalimat pengujian ada yang diambil dari transkripsi korpus ucapan yang digunakan pada pelatihan model akustik.

Terdapat 3 kondisi pengujian sistem pengenalan suara. Pertama, menggunakan *file* audio dari korpus ucapan. Data pengujian terdiri atas pembicara yang berbeda dengan yang digunakan pada pelatihan model akustik. Kedua, ucapan waktu nyata yang kontinu di mana kalimat yang diucapkan merupakan kalimat panjang yang utuh. Ketiga, ucapan waktu nyata yang semi-kontinu di mana kalimat yang diucapkan merupakan bagian dari sebuah kalimat panjang yang dibagi-bagi menjadi beberapa frase. Hasil pengujian sistem pengenalan suara yang lengkap dapat dilihat pada tabel V.

TABEL V
AKURASI PENGUJIAN PENGENALAN KATA

No. of Test	Laki-laki	Perempuan	Rata-rata
#1	58.02%	28.63%	43.32%
#2	73.66%	72.14%	72.9%
#3	79.39%	78.24%	78.81%

Pengujian perintah suara dilakukan dengan masukan ucapan waktu nyata dengan lingkungan bernilai *noise* minimum oleh dua pembicara, satu laki-laki dan satu perempuan. Parameter sukses yang digunakan adalah jika sistem melakukan aksi yang benar sesuai dengan masukan dari pembicara. Akurasi dari pengujian perintah suara ini menunjukkan nilai yang sempurna yaitu 100% untuk semua pembicara. Ini berarti bahwa sistem mampu mengenali semua perintah suara dan melakukan aksi yang benar untuk masukan *input* waktu nyata dengan lingkungan yang *noise*-nya hampir tidak ada.

Pengujian masalah pengejaan untuk menangani kata OOV dilakukan dengan lingkungan pengujian yang sama dengan pengujian perintah suara. Pengujian dilakukan untuk semua huruf (26 huruf). Hasil untuk pembicara laki-laki adalah terdapat 6 huruf yang tidak berhasil dikenali yaitu 'a', 'g', 'j', 'n', 's', dan 't'. Sedangkan untuk pembicara perempuan, terdapat 2 huruf yang tidak berhasil dikenali yaitu 's' dan 't'. Hasil pengenalan huruf selengkapnya dapat dilihat pada tabel VI.

Evaluasi dapat dibagi menjadi dua yaitu evaluasi hasil dari sistem pengenalan suara dan evaluasi dari sistem pendiktean. Berdasarkan hasil dari pengujian sistem pengenalan suara, terdapat 2 hal yang dapat disimpulkan. Pertama, pengujian masukan ucapan waktu nyata menghasilkan nilai akurasi yang lebih tinggi daripada pengujian dengan *file* audio. Hal ini mungkin disebabkan oleh kualitas suara yang lebih baik di mana pada ucapan waktu nyata, nilai volume mikrofon ditetapkan pada nilai maksimum dengan *noise* yang hampir tidak ada. Sedangkan pada *input* dari *file* audio, suara yang direkam memiliki nilai volume yang lebih kecil dan juga mengandung *noise* dari hembusan nafas pembicara. Kedua, masukan yang semi-kontinu menghasilkan nilai

akurasi yang lebih tinggi daripada yang kontinu. Hal ini menunjukkan bahwa masukan yang lebih panjang memiliki nilai probabilitas dari salah pengenalan yang lebih besar, ini dikarenakan lebih banyaknya segmentasi yang dilakukan dan penghitungan rangkaian segmen ucapan yang lebih besar. Sebagai tambahan, terdapat beberapa hal yang dapat menyebabkan salah pengenalan pada sistem pengenalan suara. Pertama, perbedaan kecepatan dan intonasi dari pembicara. Ucapan pada *file* audio memiliki kecepatan yang lebih rendah dan intonasi yang lebih beragam dibanding dengan ucapan waktu nyata. Kedua, pengucapan ejaan yang hampir mirip dari beberapa fonem seperti 'kh' dan 'h', 'au' dan 'aw', 'ai', dan 'ay'. Masalah ini dapat diperbaiki lebih lanjut dengan menambahkan data pada kamus fonetik. Ketiga, dialek regional dan aksen pembicara dapat menyebabkan beberapa pengucapan kata jadi salah dikenali. Keempat, kesalahan pada segmentasi kata, biasanya menyebabkan perbedaan peran dari sebuah kata atau suku kata seperti "dimulai" yang salah dikenali sebagai "di mulai".

TABEL VI
HASIL PENGENALAN HURUF

No.	*Input*	Hasil Pengenalan	
		Laki-laki	Perempuan
1	A	O	A
2	B	B	B
3	C	C	C
4	D	D	D
5	E	E	E
6	F	F	F
7	G	D	G
8	H	H	H
9	I	I	I
10	J	C	J
11	K	K	K
12	L	L	L
13	M	M	M
14	N	M	N
15	O	O	O
16	P	P	P
17	Q	Q	Q
18	R	R	R
19	S	F	F
20	T	P	N
21	U	U	U
22	V	V	V
23	W	W	W
24	X	X	X
25	Y	Y	Y
26	Z	Z	Z

Dalam pengujian ucapan waktu nyata, dilakukan juga pengukuran efisiensi dari sistem pengenalan suara. Hasil *decoding* dikeluarkan oleh sistem dalam rentang 1 detik, yaitu setelah kalimat selesai diucapkan oleh pembicara. Hal ini membuktikan bahwa *decoder* Julius mampu melakukan pengenalan waktu nyata dengan baik.

Untuk pengujian pendiktean, dapat dilihat bahwa penggunaan model perintah suara dalam kamus dan model bahasa cukup efektif dalam mengenali perintah suara. Sedangkan untuk pengejaan, hasil yang diperoleh masih belum baik, hal ini terkait dengan hasil pengenalan suara dengan permasalahan seperti dijelaskan pada bagian evaluasi hasil pengenalan suara. Selain itu, hasil ini dapat juga disebabkan oleh penggunaan model akustik *triphone* yang tidak menguntungkan bagi pengenalan sebuah huruf yang bersifat bebas konteks.

4. Kesimpulan

Paper ini berisi pemaparan hasil penelitian dalam membangun sebuah aplikasi pendiktean Bahasa Indonesia waktu nyata, khususnya dalam penanganan perintah suara dan kata OOV dengan strategi pengejaan. Dalam menangani perintah suara, dibangun model perintah suara dengan menambahkan
data pada kamus fonetik dan pada korpus untuk model bahasa. Untuk menangani kata OOV dengan strategi pengejaan, ditambahkan model huruf baik pada kamus maupun pada model bahasa. Hasil pengujian menunjukkan nilai akurasi pengenalan suara adalah lebih dari 70% untuk pengenalan kata yang didiktekan, 100% untuk pengenalan perintah suara, dan 85% untuk pengenalan huruf.

Untuk penelitian selanjutnya adalah dengan mengevaluasi kamus untuk mengantisipasi kata dengan lebih dari satu pengucapan, meningkatkan strategi pengejaan dan menangani permasalahan OOV dengan mengenali kata baru. Selanjutnya masalah *filler* dan *noise* juga perlu diperhatikan sebagai fokus penelitian.

Ucapan Terima Kasih

Dalam kesempatan ini, kami menyampaikan penghargaan dan rasa terima kasih pada Prof. Sadaoki Furui dan Dessi Puji Lestari dari Tokyo Insitute of Technology yang telah mengijinkan kami untuk menggunakan data suara Bahasa Indonesia.

Referensi

[1] A. Holmlid, A. Olsson, & J. Villing, Write with your voice: An Evaluation of a Dictation System, Departement of Computational Linguistics, Goteborg University, 2006, *http://www.ling.gu.se/~jessica/dictation.pdf*.

[2] D.P. Lestari, K. Iwano, & S. Furui, "A Large Vocabulary Continuous Speech Recognition

System for Indonesian Language" *In Proceeding 15th Indonesian Scientific Conference in Japan*, pp. 17-22, 2006.

[3] A. Juari & A. Purwarianti, "Implementation of Indonesian Automated Speech Recognition for OOV Detection" *In Proceedings ICACSIS 2009*, 2009.

[4] S. Sakti, E. Kelana, H. Riza, S. Sakai, K. Markov, & S. Nakamura, "Development of Indonesian Large Vocabulary Continuous Speech Recognition System within A-STAR Project" *In Proceedings Technology and Corpora for Asia Pacific Speech Translation*, pp. 19-25, 2006.

[5] R. Teruszkin & F.G.V. Resende Jr., "Implementation of a Large Vocabulary Continuous Speech Recognition System for Brazilian Portuguese," *Journal of Communication and Information System,* vol. 21, pp. 204-218, 2006.

[6] A. Lee, T. Kawahara, & K. Shikano, "Julius – An Open Source Real-Time Large Vocabulary Recognition Engine" *In Proceeding EUROSPEECH 2001*, pp. 3-6, 2001.