DETEKSI OOV MENGGUNAKAN HASIL PENGENALAN SUARA OTOMATIS UNTUK BAHASA INDONESIA

Aswin Juari dan Ayu Purwarianti

Teknik Informatika, Institut Teknologi Bandung, Bandung, Indonesia
aswin_tsy@yahoo.com, ayu@stei.itb.ac.id

Abstrak

Paper ini menjelaskan tentang implementasi pengenalan OOV (*Out of Vocabulary*) *words* pada Aplikasi Pengenal Suara Berbahasa Indonesia. Pengenalan OOV *words* penting karena masalah ini tidak dapat diselesaikan dengan menambah ukuran kamus. Untuk mengimplementasi pengenalan OOV *words*, dilakukan transduksi fonem ke kata. Klasifikasi kata-kata diberikan dengan melihat model bahasa dan probabilitas perubahan fonem untuk menentukan bagian yang termasuk OOV *words.* Pada *paper* ini juga dilakukan evaluasi terhadap beberapa jenis kamus yang digunakan pada sistem pengenal suara. Modifikasi pada kamus sistem pengenal bahasa Indonesia menghasilkan peningkatan sekitar 4% sedangkan hasil deteksi akurasi OOV sebesar sekitar 77%.

Kata kunci : *Sistem pengenalan suara otomatis, OOV, model bahasa, model akustik, kamus, dan deteksi OOV*

1. Pendahuluan

Pengenalan suara secara otomatis (*Automated Speech Recognition*), disingkat ASR, sudah dilakukan selama lima dekade [3]. Sistem ini harus dapat melakukan respon yang sesuai dari kata-kata yang diucapkan pengguna.

Salah satu masalah pada aplikasi pengenal suara otomatis adalah kemunculan OOV (*Out Of Vocabulary*) *words* [2]. OOV *words* merupakan masalah pada aplikasi pengenal suara otomatis karena kata-kata yang diucapkan oleh pembicara tidak ada dalam kamus. OOV tidak dapat diselesaikan dengan menambah ukuran basis data [6]. Hal ini karena *proper name* (contoh: nama tempat, nama sungai, dan nama orang) dan kata serapan tak mungkin didaftar semua, kata baru, dan kata yang berada di luar bahasa.

Kehadiran OOV *words* dalam suatu kalimat akan menyebabkan kesalahan pengenalan dalam suatu kalimat. Satu OOV *word* dapat menyebabkan kesalahan pengenalan beberapa kata dalam suatu kalimat. Contoh kasus, pada domain sistem cuaca JUPITER, nilai OOV *words* diperkirakan 2% dan lebih dari 13% ucapan mengandung OOV *words*. Ucapan yang mengandung OOV *words* memiliki WER (*Word Error Rate*) sekitar 51% sedangkan ucapan yang tidak mengandung OOV *words* memiliki nilai WER yang lebih rendah (10,4%) [1]. Jadi, kemampuan dalam mendeteksi lokasi OOV *words* akan meningkatkan kinerja pengenalan ucapan.

Pengenalan OOV *words* pada bahasa Indonesia belum pernah dibangun. Karena itu, *paper* ini bertujuan untuk melakukan eksperimen pengenalan OOV *words* pada bahasa Indonesia.

Beberapa metode telah digunakan untuk pengenalan OOV *words*, di antaranya model kata generik dan *word-level confidence*. *Generic word model* akan "menyerap" OOV *words* dengan menggunakan suatu struktur yang sudah dilatih dan didesain secara spesifik [1]. *Word-level confidence* merupakan pendeteksian OOV *words* berdasarkan tingkat kepercayaan suatu kata [8].

Ada juga metode transduksi fonem ke kata [7]. Teknik ini menjadi ide dasar dalam pengerjaan *paper* ini.

Untuk implementasi sistem pengenal suara berbahasa Indonesia telah dibangun [2], yaitu LVCSR (*Large Vocabulary Continuous Speech Recognition*). LVCSR merupakan sistem pengenal suara dengan data kamus yang besar.

Paper ini terorganisasi sebagai berikut. Bagian 1 menguraikan tentang pendahuluan yang berisi motivasi dan tujuan dan tujuan penelitian ini. Pada Bagian 2, dijelaskan tentang pengenalan suara otomatis. Bagian 3 berisi pembahasan tentang teknik pengenalan OOV. Bagian 4 berisi penjelasan tentang implementasi ASR untuk OOV. Hasil pengujian dijelaskan pada Bagian 5. Evaluasinya dipaparkan pada Bagian 6. Bagian 7 berisi tentang kesimpulan dan saran.

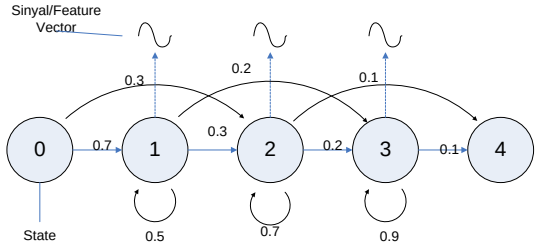
2. Sistem Pengenal Suara

Pengenalan suara otomatis merupakan teknik untuk mengenali masukan yang berupa suara untuk menghasilkan respon yang sesuai dengan masukan tersebut. Untuk membangun sistem pengenalan suara otomatis ini, dibutuhkan model akustik, model bahasa, dan kamus, yang dibahas berikut ini.

2.1. Model Akustik

Tahap pertama pemrosesan sinyal suara input adalah dengan melakukan *feature extraction* terhadap sinyal suara tersebut. Salah satu *feature extraction* yang paling banyak digunakan adalah MFCC (*Mel Frequency Cepstral Coefficients*).

Pemrosesan selanjutnya adalah pembangunan model HMM (*Hidden Markov Model*) yang terdiri atas *hidden state* (tidak dapat diamati/*hidden*) dan *feature vector* (dapat diamati/*observable*). Pembangunan model berarti pembangunan data probabilitas transisi antar-*hidden state* serta data probabilitas emisi (*emission*) yaitu pembangkitan *feature vector* oleh *hidden state*. Model HMM dapat dilihat pada Gambar 1.



Gambar 1. Model HMM

Model akustik dapat dinyatakan dalam bentuk *tied-state N-phone* atau *monophone*. Jika nilai N adalah dua, model tersebut berbentuk *tied-state biphone*. Perbedaan antara *N-phone* dan *monophone* dapat dilihat pada Gambar 2.

Monophone Sambung : s a m b u ng

Biphone Sambung : s s-a a-m m-b b-u u-ng

Gambar 2. *Monophone* dan *Biphone*

2.2. Model Bahasa

Model bahasa digunakan dalam *speech recognition* untuk membantu menentukan probabilitas dari urutan hipotesis kata. Selain itu, probabilitas model bahasa dan model akustik akan membuat sistem membatasi ruang pencarian selama pengenalan ke arah hanya urutan kata yang memiliki kemungkinan yang besar untuk benar. Jadi, hal ini akan mengurangi ruang pencarian kata sehingga proses pencarian lebih cepat dan tepat. Model bahasa dapat dibangun dengan dua pendekatan, yaitu model bahasa berbasiskan *rules* dan model bahasa *statistik*. Model bahasa berbasis *rules* artinya terdapat *rules* statis yang didefinisikan. Sedangkan, model bahasa statistik akan memberikan probabilitas dari suatu urutan kata.

2.2.1. Model berbasis *rules*

Grammar statis dari suatu bahasa ditulis. Dalam kasus ini, pengguna hanya boleh mengucapkan kata-kata yang secara eksplisit berada dalam *grammar*. Oleh karena itu, pendekatan ini hanya cocok untuk aplikasi ASR yang sederhana, misalnya pengenalan angka. Contoh model yang berbasis *rules* dapat dilihat pada Gambar 3.

```
$digit  = ONE | TWO | THREE | FOUR | FIVE |
      SIX | SEVEN | EIGHT | NINE;
$name   = [JOOP] JANSEN |
      [JULIAN] ODELL |
      [DAVE] OLLASON |
      [PHIL] WOODLAND |
      [STEVE] YOUNG;

 (SENT-START ( DIAL <$digit> | (PHONE|CALL)
```

Gambar 3. Model Berbasis *Rules*

2.2.2. Model berbasis statistik

Model bahasa berdasarkan statistik memberikan nilai probabilitas dari suatu urutan kata. Model N-gram adalah yang paling sering digunakan karena menghasilkan solusi yang lebih baik dan fleksibel. Model ini cocok untuk aplikasi ASR yang membutuhkan *vocabulary* yang besar [4].

2.2.2.1. N-Gram Model

Model bahasa N-Gram digunakan untuk menyediakan sistem pengenal dengan nilai probabilitas urutan kata tersebut muncul bersama-sama. Nilai ini diperoleh dari teks latih yang besar yang menggunakan bahasa yang sama. Jika jumlah urutan kata-kata yang dihitung probabilitas dilakukan per dua kata model bahasa tersebut adalah bigram. Jika jumlah urutan kata-kata yang dihitung probabilitas dilakukan per tiga kata, model bahasa tersebut adalah trigram.

Jika kita menganggap bahwa W adalah urutan kata, w merupakan kata-kata dalam W, dan q adalah jumlah kata, nilai P(W) dapat dilihat pada persamaan 1.

$$P(W) = P(w_1, w_2, \ldots, w_q) = \prod_{i=1}^{Q} P(w_i | w_{i-n+1}, \ldots, w_{i-1}) \quad (1)$$

Untuk memperoleh nilai probabilitas P (w_i | w_{i-2} w_{i-1}) dalam kasus trigram, dilakukan dengan hanya menghitung jumlah masing-masing kemunculan tiga kata secara berturut-turut dalam data latih. Jika N(a,b) menyatakan jumlah kemunculan a,b berturut-turut pada data latih, rumus matematisnya dapat dilihat pada persamaan 2.

$$P(w_i | w_{i-2}, w_{i-1}) = \frac{N(w_{i-2}, w_{i-1}, w_i)}{N(w_{i-2}, w_{i-1})} \quad (2)$$

2.2.2.2. N-Gram *Smoothing*

Model N-Gram merupakan model yang sederhana dan baik. Akan tetapi, masalah muncul ketika pasangan kata atau tiga buah kata ini tak muncul pada data latih. Pengubahan dari persamaan tersebut supaya tidak ada urutan kata W yang memiliki probabilitas nol disebut *smoothing*. Ide mendasar dari teknik *smoothing* adalah mengurangi sedikit nilai

probabilitas dari frekuensi relatif di persamaan tersebut (*discounting*) dan melakukan redistribusi terhadap nilai probabilitas yang terlalu kecil (*back-off*).

2.2.2.3. Penilaian Kualitas N-Gram

Jika model bahasa ada lebih dari satu, cara terbaik untuk memilih model bahasa yang digunakan adalah dengan menguji model tersebut satu per satu secara langsung dalam sistem. Namun, ada pendekatan lain yang dapat dilakukan untuk pengukuran suatu model bahasa, yaitu dengan menggunakan teori kuantitas informasi *entropy*. Jika kita mengganggap kata-kata yang diucapkan sebagai urutan dari w_1, w_2, w_3, …, w_q entropi dari sumber urutan kata dapat didefinisikan pada persamaan 3.

$$H_p = \frac{-1}{Q} \log \hat{P}(w_1, w_2, \ldots, w_q) \quad (3)$$

Di sini, $\hat{P}(w_1, w_2, w_3, \ldots, w_q)$ adalah probabilitas dari urutan kata yang diperoleh dari model bahasa dan H_p adalah nilai estimasi entropi yang menggambarkan akan seberapa mungkin model dapat memenuhi data uji. Nilai H_p yang lebih kecil melambangkan model yang lebih baik.

Ada metode lain yang berhubungan dengan *entropy* yang disebut *perplexity*. Semakin kecil nilai *perplexity*, semakin baik suatu model bahasa. Nilai *perplexity* didefinisikan pada persamaan 4.

$$P = 2^{H_p} \quad (4)$$

2.3. Kamus

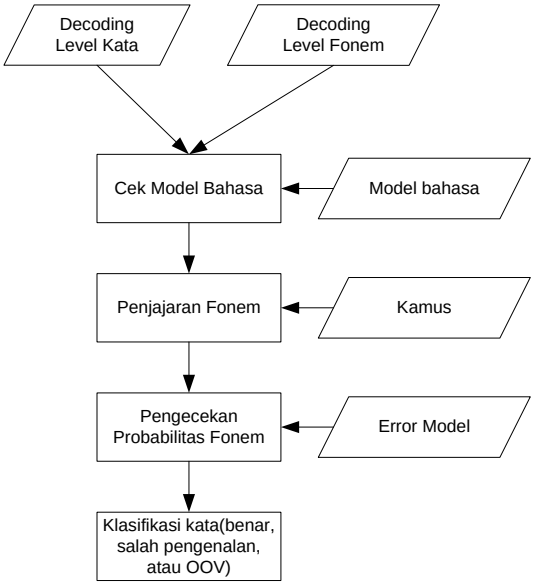
Kamus akan memberikan daftar kata yang dapat dikenali oleh sistem beserta cara pengucapannya. Kata-kata yang dikenali oleh sistem pengenal suara bergantung pada kamus.

3. Deteksi OOV *Words*

Teknik transduksi fonem ke kata dilakukan untuk melakukan deteksi OOV *words*. Teknik ini mencoba untuk melakukan perbandingan antara fonem dan kata yang dihasilkan dari proses pengenalan. Jika urutan fonem yang dihasilkan dari proses pengenalan membentuk kata yang sama dengan kata yang dihasilkan dari proses pengenalan, kemungkinan kata tersebut adalah benar semakin tinggi. Proses deteksi OOV dapat dilihat pada Gambar 4.

Secara detil, teknik ini dilakukan dengan cara sebagai berikut. Pertama, kita harus memiliki hasil *decode* level fonem dan kata dari ASR. Kemudian, penjajaran antara kata dan fonem dilakukan. Setelah itu, *error model* yang mengandung probabilitas perubahan fonem dibuat, contohnya probabilitas fonem ‘d’ yang dikenal sebagai ‘b’. Dengan cara ini, dapat diperoleh level kepercayaan dari suatu kata dari model akustiknya.

Untuk mendapatkan nilai kepercayaan dari suatu model bahasa, kita dapat mengecek probabilitas dari kehadiran pasangan kata (model bigram). Tingkat kepercayaan suatu kata akan lebih tinggi jika probabilitas N-gramnya bagus.



Gambar 4. Proses Deteksi OOV

4. Implementasi

4.1. Implementasi ASR

Text corpus digunakan untuk melatih model bahasa dan kamus yang digunakan dalam ASR ini. Koleksi dokumen yang digunakan berasal dari Koran Kompas dan Majalah Tempo. Format masing-masing artikel yang diperoleh mengikuti format TREC (*Text REtrieval Conference*). Contoh *text corpus* dapat dilihat pada Gambar 5.

```
<DOC>
<DOCID>TEMPO-020307-7687</DOCID>
<TITLE>Ekspor Bahan Baku Cat dari Sumba Timur Meningkat</TITLE>
<TEXT>
Kabupaten Sumba Timur, Nusa Tenggara Timur, akan mengembangkan kutulak yang menjadi
bahan baku cat guna mendongkrak pendapatan asli daerah karena ekspor komoditi ini terus
meningkat selama tiga tahun terakhir. Kutulak diharapkan menjadi sumber pendapatan baru
selain kayu cendana.
...
</TEXT>
</DOC>
```

Gambar 5. *Text Corpus*

Agar bisa digunakan untuk pelatihan model bahasa dan kamus, *text preprocessing* perlu dilakukan secara manual pada *corpus* ini karena format tersebut memiliki data yang tidak diperlukan dan perlu beberapa koreksi. *Preprocessing* ini meliputi mengubah semua huruf menjadi huruf kecil, menghilangkan semua tanda baca, mengubah bentuk numerik dan jam menjadi bentuk tulisannya (misal: “3000” menjadi “tiga ribu”), menggabungkan

beberapa kalimat pendek, memisahkan kalimat yang terlalu panjang, menghilangkan teks yang berada di antara ")" dan "(", mengganti tanda "-" pada kata ulang menjadi tanda " " (*whitespace*) , dan memperbaiki kata-kata yang salah penulisannya (seperti kata "utnuk" diperbaiki menjadi "untuk").

Selain itu, *text corpus* yang telah dilakukan proses *text preprocess*, dilakukan *parsing* untuk menghilangkan tag-tag, seperti judul dokumen dan id dokumen. Penambahan tag khusus "<sil>" dan "</sil>" untuk menandakan awal dan akhir kalimat pada dokumen. Format teks yang dihasilkan dapat dilihat pada Gambar 6.

<sil>kutulah diharapkan menjadi sumber pendapatan baru selain kayu cendana </sil>
<sil> .... </sil>
<sil>..... </sil>

Gambar 6. Format Teks Output

Dari teks format pada Gambar 3, dibangun model bahasa dan kamus. Model bahasa dibangun dengan menggunakan CMU *Cam Toolkit*. Sedangkan, kamus dibangun secara manual berdasarkan daftar kata.

Untuk membentuk kamus, diperlukan juga daftar fonem dalam bahasa Indonesia untuk membuat transkripsinya. Daftar fonem bahasa Indonesia dapat dilihat pada [2]. Beberapa fonem ditambahkan dan dikurangkan dari tabel tersebut

Beberapa fonem yang dihilangkan, antara lain, fonem /e/ pada kata enak dan /E/ pada kata pergi. Karena /e/ dan /E/ mirip, /e/ dan /E/ dilambangkan dengan /e/. Begitu pula dengan fonem /f/ pada kata fana dan /v/ pada kata televisi, sehingga /f/ dan /v/ dilambangkan dengan /f/ saja. Fonem /q/ ditambahkan karena dalam bahasa Indonesia banyak menggunakan serapan dari bahasa Arab. Untuk melambangkan huruf x yang digunakan pada nama dan kata-kata serapan digunakan fonem /k/ dan /s/.

Beberapa fonem lain ditambahkan untuk menandai awal kalimat, akhir sebuah kata, dan akhir kalimat. Awal kalimat dan akhir kalimat ditandai dengan sil, akhir sebuah kata dinyatakan dengan sp (*short pause*).

Agar terjadi proses pengenalan yang baik, kumpulan teks yang digunakan harus *bi-phonetically balanced* [2]. Teks dikatakan tidak *bi-phonetically balanced* jika ada kombinasi fonem yang muncul hanya satu atau dua kali saja.

Transkripsi yang dibuat pada kamus adalah sesuai dengan cara pengucapan. Jadi, kata yang merupakan singkatan, transkripsinya merupakan cara pengejaannya. Contoh kata "mpr" memiliki transkripsi "e m p e e r" bukan "m p r". Kamus yang dibuat adalah kamus monofon. Kamus dibuat secara manual. Kamus yang dihasilkan dapat dilihat pada Gambar 7.

```
</sil>  sil
<sil>   sil
a   a
ab  a b e
abadia  a b a d i a
abang   a b a ng
abdul   a b d u l
abdullah    a b d u l l a h
abdurrahman a b d u r r a h m a n
abg a b e g e
abidin  a b i d i n
aborsi  a b o r s i
abri    a b e r i
abu a b u
abubaseer   a b u b a s e e r
ac  a c e
academy a c a d e m y
acara   a c a r a
aceh    a c e h
achmad  a c h m a d
actions a c t i o n s
```

Gambar 7. Contoh Kamus

Untuk melatih model akustik, *paper* ini menggunakan 12 pembicara. Setiap pembicara mengucapkan 300 kata. Semua sinyal suara direkam dalam ruangan yang sepi dan diproses menggunakan HTK 3.4. untuk membuat model akustik. Proses pembuatan model akustik dapat dilihat pada [5].

Model akustik yang digunakan pada *paper* ini berbasis fonem. *Paper* ini membuat dua jenis dari model akustik, yaitu *monophone* dan *tied-state biphone*. Pelatihan model akustik ini akan menghasilkan model HMM. Konfigurasi implementasi ASR disajikan pada Tabel 1.

Tabel 1. Konfigurasi Implementasi ASR

	Total
Text Corpus	2000 kalimat
Perplexity	1069
Kamus	5500 kata
OOV Rate	17%

4.2. Implementasi Deteksi OOV

Paper ini melakukan prosedur pada Bagian 3 untuk melakukan implementasi deteksi OOV. Gambar 8 menggambarkan hasil dari penjajaran fonem antara hasil *decode* level fonem dan kata. Hasil *decode* level fonem memiliki konstrain lebih lemah daripada level kata. Jadi, urutan fonem dapat tidak sama dengan kata yang dihasilkan.

Karena itu, riset ini membutuhkan *error model* untuk memperbaiki hal tersebut. *Error model* memiliki probabilitas satu fonem dikenali sebagai fonem lain. Riset ini menggunakan model statistik dan pasangan fonem untuk mengimplementasikan *error model*.

```
Phone : s p a q p r e s e g a i k o s y a f o a d i m ny r p u t
c y n

Strong : staf presiden rusia vladimir putin

Alignment : s p a | q p r e s e g a i | k o s y a | f o a d i m
ny r | p u t c y n |
```

Gambar 8. Penjajaran fonem

5. Pengujian

5.1 . Pengujian ASR

Data suara yang dipakai pada saat pengujian berbeda dengan data suara yang dipakai pada saat pelatihan. Selain itu, data suara yang dipakai untuk pengujian adalah data suara yang bebas dari *noise*. Data suara yang dilatih sebanyak 300 kalimat. Masing-masing tiga ratus kalimat ini diucapkan oleh 12 orang yang berbeda. Untuk pengujian, digunakan 16 kalimat yang masing-masing diucapkan oleh dua orang berbeda. Pengujian dilakukan sebanyak tiga jenis.

1) Pengujian *baseline* dengan model akustik monofon. Pengujian ini menggunakan model bahasa 2000 kalimat dan 5500 kata. Hasil dapat dilihat pada Tabel 2.
2) Pengujian *baseline* dengan menggunakan *tied-state biphone*. Hasil dapat dilihat pada Tabel 3.
3) Pengujian *baseline* dengan menggunakan kamus yang dimodifikasi di mana setiap entri pada kamus dapat memiliki beberapa transkripsi, contohnya kata ‘kalau’ memiliki beberapa transkripsi, yaitu ‘k a l a w’, ‘k a l a o’, ‘k a l a u’, ‘k a l au’, dan ‘k a l o’. Hasil dapat dilihat pada Tabel 4.

Tabel 2. Hasil Pengujian ASR I

Data	Akurasi
Data ke-1	33.91%
Data ke-2	31.61%

Tabel 3. Hasil Pengujian ASR II

Data	Akurasi
Data ke-1	26.14%
Data ke-2	26.14%

Tabel 4. Hasil Pengujian ASR III

Data	Akurasi
Data ke-1	37.36%
Data ke-2	37.36%

5.2 Pengujian Deteksi OOV

Data yang dipakai untuk deteksi OOV merupakan data hasil pengujian ASR. Karena hasil pengujian ASR kurang baik, dilakukan pemilihan terhadap data hasil uji ASR.

```
Hasil : kepala polisi yugoslavia  <OOV> semula khawatir
Reference: kepala polisi yugoslavia  <OOV> semula khawatir

Hasil : majalah  basis berawal rapat yang  <OOV>
Reference: majalah  basis berawal rapat yang dihadiri oleh <OOV>

Hasil : <OOV> sebuah tendangan penalti
Reference: <OOV> menembakkan sebuah tendangan penalti
```

Gambar 9. Kasus OOV yang Terdeteksi Adalah Benar

Sebuah kata dikatakan salah dikenali jika kata tersebut tidak sama dengan kata yang ada di *reference*. Sebuah segmen dikatakan memiliki OOV jika segmen tersebut mengandung kata yang tidak ada pada *reference*. Sebuah segmen yang dikatakan OOV dapat terdiri dari beberapa kata yang berurutan. Segmen yang mengandung OOV bisa jadi memiliki kata yang salah dikenali karena hasil transkripsi yang salah. Namun, tidak semua kata yang transkripsinya salah adalah OOV. Gambar 9 menggambarkan beberapa contoh kasus OOV yang terdeteksi adalah benar.

Hasil pengujian direpresentasikan dalam empat buah nilai. Pertama, nilai akurasi dari pendeteksian kesalahan pengenalan kata yang berarti seberapa baik sistem dapat mendeteksi kesalahan pengenalan kata. Kedua, akurasi deteksi OOV *words* yang berarti seberapa baik sistem dapat memprediksi lokasi OOV *words*. Ketiga, *false alarm* dari deteksi kesalahan pengenalan kata yang berarti seberapa sering sistem salah dalam prediksi kesalahan pengenalan kata (sistem mengenali sebagai kesalahan pengenalan kata, padahal sebenarnya tidak demikian). Keempat, *false alarm* dalam deteksi OOV *words* yang berarti sistem mengenali sebagai lokasi OOV *words*, walaupun sebenarnya tidak. Hasil pengujian dapat dilihat pada Tabel 5.

Tabel 5. Hasil Pengujian Deteksi OOV

Jenis	Nilai (%)
Acc. Kesalahan Pengenalan Kata	87.39
Acc. OOV	76.92
False Alarm Kesalahan Pengenalan Kata	15.79
False Alarm OOV	37.5

6. Evaluasi

Terdapat dua macam evaluasi yaitu: (1) evaluasi terhadap hasil pengenalan suara otomatis; (2) evaluasi terhadap hasil deteksi OOV.

Dalam *paper* ini, pengujian ASR belum memberikan hasil yang baik. Pada [2], ASR menghasilkan *output* sekitar 80%. Ada beberapa hal yang menyebabkan hasil tersebut. Pertama, fonem bahasa Indonesia memiliki cara pengucapan yang sama. Contohnya, fonem ‘au; dan ‘o’, fonem ‘kh’ dan ‘h’, dan fonem ‘b’ dan ‘d’. Memperbaiki kamus dapat

meningkatkan hasil pengenalan. Kedua, bahasa Indonesia merupakan bahasa yang *agglutinative*. Contohnya, kata 'di' memiliki fungsi yang berbeda, yaitu sebagai kata depan dan awalan. Ketiga, jumlah OOV yang besar turut menurunkan akurasi dari pengenalan ucapan. Keempat, pengucapan kata yang terlalu cepat akan menurunkan kualitas ucapan karena menjadi tak dapat dikenali dengan baik oleh sistem.

Untuk deteksi OOV, nilai *false alarm* deteksi OOV lebih tinggi daripada false alarm deteksi kesalahan pengenalan. Beberapa hal menyebabkan hal tersebut. Pertama, banyak kata dari data input yang merupakan kesalahan pengenalan sehingga membingungkan sistem untuk mengklasifikasi kata tersebut apakah OOV atau bukan. Kedua, sistem sulit melakukan klasifikasi kata yang salah dikenali tersebut apakah berupa OOV atau kesalahan pengenalan karena pada OOV *words* juga terdapat kesalahan pengenalan.

7. Kesimpulan

Paper ini telah menjelaskan eksperimen untuk membangun ASR serta melakukan deteksi OOV *words*. Evaluasi dari eksperimen membangun ASR adalah beberapa fonem dalam bahasa Indonesia memliki pengucapan yang mirip. Hal ini menyebabkan akurasi dalam pengenalan suara menjadi lebih rendah. Evaluasi dari deteksi OOV adalah kata-kata yang salah pengenalan dapat diprediksi dengan mengecek model bahasa dan model akustiknya.

Ada beberapa perbaikan dari riset ini yang dapat dilakukan di masa depan. Pertama, perbaikan dapat dilakukan dengan cara menggabungkan deteksi OOV dengan sistem pengenalan suara. Kedua, perbaikan juga dapat dilakukan dengan melakukan eksperimen tentang bagaimana membuat koreksi jika kata-kata yang dikenali adalah OOV. Ketiga, perbaikan juga dapat dilakukan dengan menerapkan metode lain untuk mengecek kehadiran OOV *words*.

8. Ucapan Terima Kasih

Ucapan terima kasih kami berikan pada Sadaoki Furui yang telah memberikan data suara pada kami sehingga tak perlu dilakukan perekam ulang dan Dessi Puji Lestari yang telah menyempatkan waktunya untuk berdiskusi dengan kami tentang topik riset.

REFERENSI

[1] Issam, James R. Glass, "Modelling Out Of Vocabulary *Words* For Robust Speech Recognition", in *Proceedings of ICLSP*, 2000.

[2] Puji Lestari, Dessi, Koji Iwano, Sadaoki Furui, "A Large Vocabulary Continuous Speech Recognition System For Indonesian Language", in *15th Indonesian Scientific Conference in Japan Proceeding.* ISSN:1881-4034, 2006.

[3] Rabiner, Lawrence, Biing-Hwang Juang. *Fundamental of Speech Recognition*. Prentice Hall, 1993.

[4] Zhang, Le dan Steve Renals. "Statistical Language Model, 2000". http://homepages.inf.ed.ac.uk/s0450736/slm.html. Tanggal akses: 30 Maret 2009.

[5] Steve Young et al. "The HTK Book (for version 3.4)". Cambridge University Department, 2006.

[6] Singhal, Amit, John Choi, Donald Hindle, Fernando Pereira."SDR Track", in *Proceedings of the Eighth Text Retrieval Conference, NIST Special Publication* 500-246, 2000, pp. 317-330.

[7] Zweig, Geoffrey et al. "Empirical Properties of Multilingual Phone-to-Word Transduction", in *Proceedings of ICASSP*, 2008.

[8] C. Pao, P. Schmid, dan J. Glass, "Confidence Scoring for Speech Understanding", in *Proceedings of ICSI,* 1998.