Langkah Praktis Membangun Sistem Pengenalan Suara dengan HTK

Zulkarnaen Hatala
Jurusan Teknik Elektro Politeknik Negeri Ambon, Indonesia
dzulqarnaenhatala@gmail.com

***Abstract*—Dipaparkan prosedur untuk mengembangkan Sistem Pengenalan Suara otomatis, *Automatic Speech Recognition* System (ASR) untuk kasus *online recognition*. Prosedur ini secara cepat dan efisien membangun ASR menggunakan Hidden Markov Toolkit (HTK). Langkah-langkah praktis ini dipaparkan secara jelas untuk mengimplementasikan ASR dengan daftar kata sedikit (*Small Vocabulary*) dalam contoh kasus pengenalan digit Bahasa Indonesia. Dijelaskan beberapa teknik meningkatkan performansi seperti cara mengatasi noise, pengejaan ganda dan penerapan Principle Component Analysis. Hasil akhir berupa *Word Error Rate*.**

Keywords—automatic speech recognition, small vocabulary, noise robust recognizer

I. PENDAHULUAN

Sistem pengenalan suara otomatis, *automatic speech recognition system* (ASR) bertujuan untuk menerjemahkan sinyal digital suara manusia ke bentuk teks atau tulisan bermakna menurut tata bahasa tertentu. Sinyal suara manusia ditangkap oleh *microphone* dan diolah komputer untuk mendapatkan tulisan teks yang beraturan. Dengan demikian ASR bisa digunakan untuk tujuan lebih lanjut seperti memberi perintah kepada komputer untuk melakukan sekelompok tugas dengan hanya berbicara. Contoh aplikasi ASR misalnya seseorang di rumahnya menginstruksikan komputer untuk mematikan dan menghidupkan lampu kamar mandi, lampu taman tanpa harus menekan saklar listrik cukup dengan memberikan perintah suara. Kasus lain dari penerapan ASR adalah ketika sesorang sedang berkendara dan ingin menginstruksikan *smartphone*nya untuk memberitahukan posisi sekarang, atau informasi rute tujuan bahkan informasi tentang kondisi *on/off* semua saklar listrik dirumahnya.

Berdasarkan jumlah kata yang didukung maka ASR dibedakan menjadi *small size vocabulary ASR* (SVASR) dan *large vocabulary* ASR (LVASR). SVASR Cuma mendukung jumlah kata sangat sedikit dibanding LVASR. Contoh SVASR ketika diaplikasikan untuk mematikan saklar-saklar listrik di rumah atau aplikasi pengenalan digit atau pengenalan *phonebook contact* untuk men*dial* otomatis ditelpon. Sebaliknya LVASR digunakan sebagai pengganti *keyboard* untuk mengetik dengan cara berbicara secara alami *natural* di *search engine* untuk topik dan susunan kata yang tidak terbatas. Bahkan LVASR dapat digunakan sebagai mesin penerjemah otomatis antar dua bahasa yang berbeda seperti antara bahasa Inggris ke Bahasa Indonesia dan sebaliknya. Di sini pembicara dengan mesin (*smartphone*) LVASR tidak perlu memahami bahasa lawan bicaranya karena akan diterjemahkan oleh mesin tersebut [1].

Tetapi LVASR mensyaratkan *hardware* yang mumpuni dengan *processor* dan *memory* minimal yang tinggi. Hal ini membuat LVASR sulit diterapkan langsung pada lingkungan hardware terbatas seperti *embedded devices*, *smartphones* dan *home appliance peripherals*. Di sinilah SVASR berpeluang untuk dikembangkan. Dengan jumlah kata yang sedikit tentu tidak membutuhkan prasyarat *hardware* yang tinggi. SVASR juga bisa dikembangkan secara cepat dengan akurasi yang tinggi. Beberapa hal tersebut membuat SVASR akan mendapatkan tempat yang signifikan untuk menerjemahkan perintah-perintah suara sederhana yang berguna bagi keseharian manusia.

II. STATE OF THE ART

A. Pengujian ASR

Pengujian sistem ASR bisa dilakukan secara *offline* dan *online*. Pengujian *offline* dilakukan terhadap data yang sudah direkam dan tersimpan dalam suatu basis data *speech corpus database*. Sedangkan pengujian online adalah pengujian aplikasi yang dilakukan langsung secara nyata *real time* ditempat aplikasi diimplementasikan. Kecocokan antara data pelatihan dan data pengujian sangat menentukan keberhasilan implementasi ASR itu sendiri. Tetapi walaupun kondisi pengujian tidak sama dengan kondisi pelatihan ASR tetap bisa diimplementasikan dengan menggunakan teknik-teknik transformasi dan adaptasi tertentu.

B. Hidden Markov Toolkit

Hidden Markov Toolkit (HTK) [1] adalah sekelompok program dan pustaka yang digunakan untuk mengembangkan ASR dengan model Hidden Markov. Tetapi sebenarnya HTK sendiri sudah lengkap untuk membangun sebuah basis data suara (*speech corpus*), ekstraksi fitur, pembentukan dan pelatihan model akustik serta melakukan pengujian baik *offline* maupun *online real time* .

III. METODOLOGI

Di usulkan prosedur untuk membangun *small vocabulary automatic speech recognition system (SVASR).* Secara umum langkah-langkah tersebut adalah menentukan *grammar* atau tata bahasa yang akan didukung, membuat basis data pelatihan *training database*, ekstraksi ciri *feature extraction*, membuat dan melatih *subword acoustic model training*,

pengujian *testing*, perbaikan *refining* dan penanaman *deployment*.

A. Menentukan Grammar

Dalam merancang SVASR hal yang paling pertama adalah menentukan tata bahasa *grammar* yang diterima oleh ASR tersebut. *Grammar* untuk SVASR berbeda dengan *grammar* dari LVASR dalam hal jumlah kata untuk SVASR adalah lebih sedikit. *Grammar* SVASR lebih sederhana dengan *production rules* yang juga lebih sedikit. Setelah *grammar* selesai ditentukan maka menyusul hal-hal berikut yang dilakukan:

a) Menentukan Daftar Kata (word list): Berdasarkan grammar, maka semua daftar kata bisa ditemukan dan diurutkan.

b) Membuat kamus pengejaan (subword pronounciation dictionary): Setiap kata yang terdaftar pada *word list* dipetakan ke *subword*nya (*phoneme*). Untuk daftar *phoneme* yang digunakan mengikuti konvensi menurut kaidah *International Phonetic Association (IPA)* [2], *atau TI-MIT* [3][4].

B. Membuat basis data suara

Untuk membuat basis data suara maka prosedur yang dilakukan adalah:

a) Perekaman suara: Suara direkam menggunakan *microphone* yang sesuai dengan kondisi yang diinginkan dan cocok dengan kondisi pengujian. Dalam artian agar menghindari perbedaan terlalu besar antara kondisi pelatihan dan kondisi pengujian. Misalkan pada saat perekaman dalam kondisi laboratorium yang senyap tetapi pada saat pengujian dilakukan dalam kondisi gangguan tinggi, maka akurasi pengujian tentu akan berkurang. Suara yang direkam tentunya mengacu pada daftar kata (*word list*) di langkah A (*grammar*) sebelumnya. Dengan tidak merekam kata yang diluar *word list* tersebut maka proses perekaman tentu lebih cepat di banding jika merekam kata-kata tambahan. Frekuensi pensampelan yang digunakan adalah 16KHz [4], dan hasilnya disimpan sebagai file dengan format PCM 16 bit. Dalam perekaman juga diatur sensivitas *microphone* sehingga menghasilkan basis data suara yang baik. Apabila *microphone* di atur terlalu sensitif maka suara yang terjadi bisa mengandung *noise* yang terlalu kuat, di mana hal ini bisa mengurangi performansi ASR.

b) Pelabelan basis data suara : tiap file suara yang telah direkam kemudian ditandai berdasarkan *subword* atau *fonem*. Suara di*label* menggunakan *software* seperti *Speech Filing System*, SFS [5] dan Praat [6]. Dalam hal ini SFS memiliki kelebihan dalam hal kompatibilitas dengan HTK [1]. File suara ditampilkan dalam domain spektral kemudian ditandai setiap fonemnya secara berurutan berdasarkan waktu. Penamaan fonem menggunakan panduan *International Phonetic Association* (IPA) [2] atau ARPABET dan TIMITBET. Sedangkan contoh-contoh pelabelan tiap fonem tertentu terhadap spektralnya mengikuti contoh dari CMU Arctic Speech Databases [7] atau TIMIT Database [3]. Perekaman juga dilakukan terhadap *noise* selain terhadap kata-kata dalam *grammar*.

C. Extraksi fitur (Feature Extraction)

File-file suara yang telah direkam kemudian dikonversi ke format ciri atau format fitur. Misalnya saja yang digunakan adalah format *Mel Frequency Cepstral Coefficient* (MFCC) [8]. Setelah itu dari 39 koefisien itu difilter menggunakan metoda PCA [9]. Hal ini untuk meningkatkan level akurasi karena suara yang diinput telah terdistorsi oleh *noise* [10].

D. Pembuatan Model Akustik

Setelah diperoleh file-file fitur dalam format MFCC maka dikonstruksilah model akustik dalam format Hidden Markov Model dengan distribusi emisi berupa Gaussian Mixture Model (HMM-GMM) [11]. Fase ini sering juga disebut fase pelatihan (*training phase*). Model akustik yang diusulkan adalah yang berbasis *subword*. Yaitu satu model HMM untuk satu fonem (*subword)* tertentu, contohnya satu HMM untuk /**s**/, satu HMM untuk /**ah**/ dan seterusnya.

E. Pengujian

Setelah diperoleh model akustik maka saatnya melakukan pengujian terhadap model. Dalam fase ini diukur tingkat akurasi atau keberhasilan suatu SVASR.

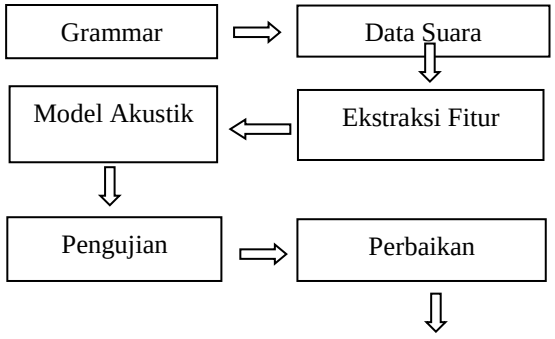
F. Perbaikan

Kadang setelah pengujian di lapangan secara *online* atau *real time* ternyata terjadi *error*. *Error* ini bisa diperbaiki jika data pengujian tersebut telah direkam. Data dari lapangan ini bisa dimanfaatkan untuk memperbaiki *error* tersebut. Dalam hal ini sistem kembali dilatih ulang dengan mengikutkan data *error* yang baru diperoleh tersebut.

G. Penanaman

Setelah dicapai hasil yang diinginkan maka ASR ditanam *deployed* di lingkungan yang diinginkan. Lingkungan penanaman bisa berbeda dan juga bisa sama dengan lingkungan pengujian *offline* dan *online* atau berbeda dari lingkungan perekaman basis data. Jika lingkungan penanaman berbeda maka harus dilakukan teknik-teknik adaptasi tertentu. Proses adaptasi tidak dibutuhkan jika lingkungan penanaman adalah sama persis dengan lingkungan pengujian dan sama juga dengan lingkungan perekaman basis data suara.

Langkah-langkah ini bisa dirangkum dalam Figure-1



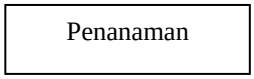


Fig. 1. *Metodologi pengembangan SVASR*

IV. ILUSTRASI DAN HASIL

Berikut diilustrasikan pengembangan SVASR dengan metoda yang diusulkan menggunakan *software* HTK [1]. Aplikasi yang dicontohkan adalah membangun sistem pengenalan angka *Indonesian digit recognizer (*IDRec*).*

A. Software pendukung

Untuk mengimplementasikan ASR selain menyiapkan *hardware* berupa komputer dan *bluetooth microphone*, harus juga diinstall *software* pendukung sebagaimana terlihat pada TABLE I.

TABLE I. DAFTAR SOFTWARES DAN LIBRARIES

No	Software, Libraries		
	Software	***Version***	***Lisensi***
1	Speech Filing System	Release 4.9, SFSWin 1.9	UCL
2	HTK	3.4.5	Cambridge University
3	Visual Studio	2008	Microsoft
4	Praat	6.0.43	Paul Boersma
5	Windows OS	Pro 10.0.1439	Microsoft
6	Cygwin-gcc	2.11.2(0.329/5/3) - i686	GPL

HTK diunduh kemudian dikompilasi menggunakan Cygwin-gcc atau Visual Studio.

B. Grammar

Grammar untuk IDRec adalah sebagaimana terlihat dalam Fig.1. Di sini 4 angka dalam bahasa Indonesia 1,2,3,4 melambangkan nomor switch elektrik pada instalasi sederhana di perumahan.

```
DIGIT = NUM_0 | NUM_1| NUM_2 | NUM_3 | NUM_4 |
        NOISE;
SILENCE = SIL ;
ON_OFF = ON| OFF|HIDUP|MATI;
CONT = <DIGIT>ON_OFF | SIL | NOISE ;
SENTENCE = { <$CONT SILSP> | <$CONT > SILSP  } ;
(SILENCE SENTENCE SILENCE)
```

Fig. 2. *Grammar* untuk IDRec

Pada Fig. ? tanda *bar* '|' berarti logika 'OR' atau 'ATAU'. Dan untuk kamus pengejaan *subword pronounciation dictionary* mengikuti daftar fonem pada [4] dikutip sebagian sebagaimana dalam Fig.2

```
NOISE    t sp
NOISE    d  sp
NUM_0    k oh s oh ng
NUM_1    s ah t uh
NUM_2    d uw  ah
NUM_3    t ih g ah
NUM_4    ah m p ah t
...
```

Fig. 3. *Pronounciation dictionary* untuk IDRec

Jadi pada kamus pengucapan di Fig.? di atas kata (*word)* diletakkan pada kolom pertama dan pengucapannnya diletakkan pada kolom-kolom berikutnya. *Subword* sp pada akhir beberapa entri di kamus di atas adalah singkatan dari *short* pause yaitu jika *silence* terjadi tetapi dalam waktu singkat. Kata NOISE merepresentasikan gangguan *noise* yang mungkin saja terjadi pada saat pengujian. Sebagai contoh di Fig.? *Noise* yang muncul terdeteksi sebagai pola urutan fonem tertentu misalnya urutan dua konsonan berturut tanpa vokal di antara keduanya.

Dalam bahasa Indonesia beberapa kata muncul berurutan sebagai *contraction* yaitu akhiran suata kata merupakan awalan kata berikutnya. Contohnya adalah jika kalimat NUM_3 NUM_4 diucapkan maka fonem /**ah**/ pada akhiran kata NUM_3 sekaligus merupakan awalan pada kata NUM_4. Dalam kasus ini Hvite sering Cuma mengenali satu kata saja yang berarti *word deletion error* terjadi. Solusi yang digunakan adalah menciptakan kata baru dalam kamus subkata. Contohnya untuk kasus NUM_3 NUM_4 solusinya adalah seperti pada Fig. 3

```
NUM_3_4   t ih g ah m p ah t
```

Fig. 4. *Pemodelan contraction dengan subword dictionary*

Pemodelan *contraction* dengan *dictionary* ini efektif sekali untuk SVASR, tetapi akan sangat bermasalah untuk LVASR.

C. Basis Data Suara

a) Perekaman suara: suara direkam menggunakan software HSLab dan HVite dengan perangkat input *bluetooth microphone*. Dengan sedikit modifikasi dan rekompilasi maka didapatkan replay buffer dari **HVite** yang merupakan data suara *raw* pada pengujian *online*. Bisa juga menggunakan *softwares* perekam yang lain seperti Audacity dan CoolEdit, tetapi *softwares* tersebut tidak digunakan, karena pada saat perekaman diinginkan basis data yang benar-benar sama dengan kondisi pengujian. Untuk *bluetooth microphone* yang digunakan adalah yang bermerek umum dan mudah dijumpai di pasaran. Perangkat ini bisa diperoleh dengan harga murah tetapi tanpa fungsi *noise cancellation*. Microphone ini juga cuma berisikan satu sensor tunggal sebagaimana menurut berbagai literatur adalah kurang baik dibanding yang berisikan *array of censor*. Contoh microphone yang digunakan bisa di lihat pada Fig. ? Keuntungan menggunakan wireless microphone adalah pembicara bebas bergerak dan tidak terikat ke smartphone atau komputer pemroses ASR.

Fig. 5. *Bluetooth Microphone*

Tempat perekaman adalah kondisi perumahan, ruangan kantor dan laboratorium tidak kedap suara dengan tingkat gangguan noise rendah tetapi tidak sepenuhnya bebas dari gangguan. Pengaturan sensitivitas *microphone* juga berpengaruh terhadap akurasi. Untuk *bluetooth microphone* maka volume suara diturunkan hingga *c* kali dari aslinya. Misalkan *x(t)* adalah sinyal audio yang dihasilkan oleh *microphone* maka *x^(t)* adalah sinyal yang diproses oleh HTK setelah diturunkan terlebih dahulu menurut persamaan di Fig. 4.

$$x\hat{}(t)=cx(t),\ 0<c<1$$

Fig. 6. *Pengaturan volume microphone*

Hal ini bertujuan untuk meredam bunyi *noise* seperti langkah manusia, kokok ayam bahkan suara mesin motor. Suara *noise* selain memicu *silence detector* dari HVite, sering juga *noise* ini dideteksi sebagai *word* dari percakapan yang tentunya mereduksi akurasi ASR. Pada akhirnya di kondisi perekaman diperoleh *power sample* rata-rata 21 dB untuk kondisi *silence* dan di atas 50 dB untuk suara. Sebagai ilustrasi kondisi ruangan dengan kipas berdiri (*standing fan*) berdaun plastik berjarak 30 cm dari *mic* akan menunjukkan indikator 45 dB. Kondisi pengukuran *power* rata-rata berbagai lingkungan perekaman ini di sajikan dalam tabel II.

TABLE II. PENGUKURAN POWER BERBAGAI KEADAAN PEREKAMAN

No	Power berbagai kondisi perekaman		
	Kondisi	***Pengukur***	***SNR V-S (dB)***
1	Ruangan kamar tanpa kipas angin	HVite	60-20
2	Ruangan kamar dengan suara kipas angin 30 cm dari *mic*	HVite	70-35
5	Ruangan kamar Tengah Malam Sunyi	Hvite	70-15

Sampling frequency yang digunakan adalah 16000 Hz atau periode antar sample sebesar 625 ns. File hasil rekaman disimpan dalam format HTK *raw* 16 bit *signed integer*.

b) Pelabelan (annotating): Suara dilabel menggunakan software Praat [6], setelah itu dikonversi ke format SFS [5] kemudian dikonversi lagi ke dalam format HTK. Hal ini karena Praat belum bisa mengkonversi langsung ke format HTK. Untuk melabel data pelatihan maka panduan subword (*phonem*) mengikuti *grammar* yang dihasilkan pada seksi B. Pada Figure 5 diperlihatkan contoh menandai *subword* dengan Praat untuk urutan kata NUM_2 NUM_1.

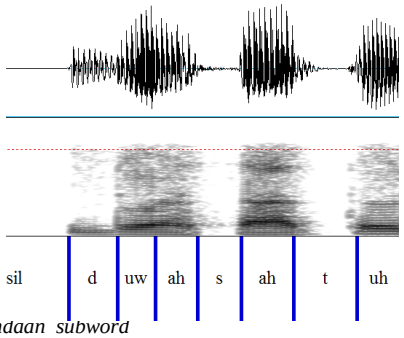


Fig. 7. *Penandaan subword*

c) Data gangguan (noise): pada tahap ini data *noise* juga diambil dan dilabel untuk dimasukkan ke basis data. Noise pasti muncul dalam lingkungan perekaman nyata di lapangan. Contoh *noise* pada tampilan SFS adalah seperti di Fig. 7?

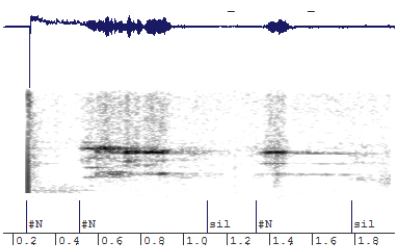


Fig. 8. *Menandai noise dalam basis data*

D. Ekstraksi fitur (Feature Extraction)

Ekstraksi fitur untuk merubah format suara ke format MFCC *(Mel Frequency Cepstral Coefficient)*. Dalam hal ini data yang diambil adalah format suara *raw* dari *ReplayBuffer* HVite atau dari hasil perekaman dengan program HSLab. Ekstraksi fitur dilakukan oleh program HTK bernama **HCopy**, yaitu untuk merubah *file-file* suara *raw* berekstensi **.ht**k menjadi *file-file* fitur berekstensi **.mfc**. Format yang digunakan untuk fitur adalah yang berkode HTK MFCC_0_D_A. Format ini adalah format MFCC dengan menambahkan koefisien spektral ke nol, (*zeroth coefficient)*, *delta coefficient* dan *acceleration coefficient* [8]. Sebelumnya dilakukan *preprocessing* yaitu *emphasizing* dengan koefisien 0.97 dengan *windowing* sebesar 250 ms (400 *samples*) yang terdiri dari *framing/striding* sebesar 100ms (160 *sample*) dan *overlapping* sebesar 150 ms (240 *samples*). Setelah diperoleh MFCC_0_D_A, kemudian dilanjutkan dengan *PCA*[9], di mana berdasarkan hasil dari PCA maka koefisien yang digunakan tersisa 15 koefisien saja yaitu 1, 26-39.

E. Pembuatan Model Akustik

Model akustik dibuat menggunakan program-program Hinit dan HERest. Hinit untuk melakukan inisialiasi parameter dengan algoritma Viterbi Extraction. Sedangkan HERest untuk mengestimasi parameter HMM-GMM dengan menggunakan algoritma Baum-Welch.

F. Pengujian

Pengujian dilakukan dengan program HTK Hvite. Dalam hal ini pengujian dilakukan secara offline dan online. Untuk

offline testing dilakukan terhadap basis data suara yang telah direkam. Sedangkan *online testing* dilakukan juga menggunakan perangkat komputer yang terpasang *bluetooth microphone* dengan pembicara langsung mengucapkan digit-digit secara acak. Software yang digunakan untuk online testing adalah Hvite dari HTK. Untuk *word error rate* diperoleh diperoleh bisa hingga di bawah 5% dengan artian tingkat pengenalan bisa mencapai 95%.

G. Perbaikan

Perbaikan dilakukan dengan melihat hasil dari online testing. Jika pada online testing terjadi kesalahan pada kalimat yang diucapkan maka kalimat tersebut diambil dan diberi *label* serta ditambahkan ke dalam basis data. Kemudian dilakukan pengujian ulang terhadap kalimat tersebut. Setelah itu diteliti apakah kalimat sebelumnya sudah bisa dikenali secara tepat sekarang. Hasil penelitian menunjukkan bahwa cara ini berhasil memperbaiki kesalahan-kesalahan pengujian secara *online*.

Kadang walaupun data *error* sudah dimasukkan dalam database tetapi tetapi tidak bisa menghilangkan kesalahan, maka perbaikan bisa dilakukan dengan melihat urutan fonem yang salah deteksi kemudian menambahkan entri tersebut ke *dictionary* sebagai *word* secara eksplisit. Pada Fig. 8? kita perlihatkan entri multi versi untuk pengejaan NUM_3, yang apabila ada entri yang tidak ditambahkan secara tegas maka akan menyebabkan *error*.

NUM_3	t+ih ih-g+ah g-ah+ah
NUM_3	ah-t+ih t-ih+g ih-g+ah uw-ah

Fig. 9. *Entri untuk diversitas pengejaan multi versi*

H. Penanaman

Penanaman dilakukan di komputer dengan sistem operasi Windows dengan lingkungan sama seperti tahap pengujian *online*. Software yang ditanam adalah **HVite** dari HTK. Pembicara akan mengucapkan digit-digit secara berulang-ulang dan acak melalui input *bluetooth microphone*. HVite akan merekam sekaligus mengenali ucapan tersebut berdasarkan model akustik HMM-GMM yang telah dilatih. Hasil dari penanaman bila dirasa perlu bisa dicatat untuk dilakukan perbaikan lagi berulang-ulang kali sehingga di peroleh target level akurasi tertentu.

V. KESIMPULAN

Small Vocabulary Automatic Speech Recognition (SVASR) bisa dikembangkan dengan cepat dan efisien menggunakan prosedur yang diusulkan. Keberhasilan implementasi SVASR sangat tergantung tingkat gangguan (*noise*). Faktor lainnya adalah kesamaan antara kondisi perekaman pelatihan dengan kondisi uji coba. Faktor-faktor ini termasuk di dalamnya adalah tingkat dan jenis gangguan sekitar, ketergantungan satu pembicara dan jenis input microphone yang digunakan. Namun peroleh *word error rate* di bawah 5% adalah tidak mustahil untuk *small size vocabulary*.

VI. DISKUSI

Prosedur memiliki ketergantungan dengan mensyaratkan kesamaan kondisi lingkungan pengujian dan pelatihan. Termasuk juga ketergantungan kesamaan pembicara tunggal dan perangkat *input* microphone. Pengembangan yang diinginkan adalah dukungan terhadap jumlah kata yang lebih banyak. Diharapkan juga untuk meminimalisir atau menghilangkan prosedur perekaman.

REFERENCES

[1] S. Young, E. Gunnar, G. Mark, T. Hain, and D. Kershaw, “The HTK Book version 3.5 alpha,” Cambridge University, 2015.

[2] C. D. Soderberg and K. S. Olson, “Illustration of the IPA: Indonesian,” *J. Int. Phon. Assoc.*, vol. 38, no. 2, pp. 209–213, 2008.

[3] C. Lopes and F. Perdigão, “Phone Recognition on the TIMIT Database,” 2009.

[4] K. Lee and H.-W. Hon, “Speaker-Independent Phone Recognition Using Hidden Markov Models,” *IEEE Trans. Acoust.*, vol. 37, no. 11, pp. 1641–1648, 1989.

[5] M. A. Huckvale, D. M. Brookes, L. T. Dworkin, M. E. Johnson, D. J. Pearce, and L. Whitaker, “The SPAR Speech Filing System,” *Eur. Conf. Speech Technol.*, pp. 305–308, 1987.

[6] P. Boersma and V. van Heuven, “Speak and unSpeak with Praat,” *Glot Int.*, vol. 5, no. 9–10, pp. 341–347, 2001.

[7] K. John and A. W. Black, “The CMU ARCTIC Speech Databases,” in *5th ICSA Speech Synthesis Workshop - Pittsburg*, 2004, pp. 223–224.

[8] S. B. Davis and P. Mermelstein, “Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences,” *IEEE Trans. Acoust.*, vol. 28, no. 4, pp. 357–366, 1980.

[9] L. I. Smith, “A tutorial on Principal Components Analysis,” 2002.

[10] T. Takiguchi and Y. Ariki, “PCA-Based Speech Enhancement for Distorted Speech Recognition,” *J. Multimed.*, vol. 2, no. 5, pp. 13–18, 2007.

[11] L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” *Proc. IEEE*, vol. 77, no. 2, 1989.