Jurnal Linguistik Komputasional
Indonesian Association for Computational Linguistics
Vol 1, No 2
September
2018
e-ISSN :
2621-9336

Uji Coba Korpus Data Wicara BPPT sebagai Data Latih Sistem Pengenalan Wicara Bahasa Indonesia

Made Gunawan[#1], Elvira Nurfadhilah[#2], Lyla Ruslana Aini[#3], M. Teduh Uliniansyah[#4], Gunarso[#5], Agung Santosa [#5], Juliati Junde[#6]

[#]*Pusat Teknologi Informasi dan Komunikasi, Badan Pengkajian dan Penerapan Teknologi Jakarta, Indonesia*

[1] made.gunawan@bppt.go.id
[2]elvira.nurfadhilah@bppt.go.id
[3] lyla.ruslana@bppt.go.id
[4]teduh.uliniansyah@bppt.go.id
[5]gunarso@bppt.go.id
[6]agung.santosa@bppt.go.id
[7]juliati.junde@bppt.go.id

***Abstrak*— Kami menyajikan hasil uji coba pengenalan wicara menggunakan Korpus Data Wicara BPPT yang dikembangkan tahun 2013 (KDW-BPPT-2013) dengan menggunakan anggaran DIPA tahun 2013. Korpus ini digunakan sebagai data latih dan data uji. Korpus ini berisi ujaran dari 200 pembicara yang terdiri dari 50 laki-laki dewasa, 50 laki-laki remaja, 50 perempuan dewasa, dan 50 perempuan remaja dengan masing-masing mengucapkan 250 kalimat. Total lama ujaran data wicara ini sekitar 92 jam. Uji coba dilakukan dengan menggunakan Kaldi dan menghasilkan *Word Error Rate* (WER) GMM 2,52 % dan DNN 1,64%.**

***Kata kunci*— pengenalan wicara, korpus data wicara, GMM, DNN, WER.**

I. PENDAHULUAN

Sejak tahun 2008 BPPT telah mulai melakukan penelitian pengenalan suara terutama untuk bahasa Indonesia. Pada tahap awal penelitian telah berhasil dikembangkan aplikasi ILVC (*Indonesian Linux Voice Command*) yang merupakan prototipe sistem pengenal wicara bahasa Indonesia kata per kata yang digunakan untuk mengoperasikan komputer berbasis sistem operasi Linux. Sistem ini secara khusus dikembangkan untuk mengenali beberapa perintah Linux dalam bahasa Indonesia yang kemudian diubah ke dalam perintah Linux dan menjalankannya. Aplikasi ini selanjutnya dikembangkan untuk mengenali ujaran kontinyu (*continuous speech*) bahasa Indonesia sehingga bisa digunakan untuk menulis dokumen. Aplikasi ini disebut LiSan (Linux dengan liSan). Pengembangan lebih lanjut lagi menghasilkan Perisalah, suatu sistem pengenalan suara yang dapat membuat rangkuman rapat secara semi automatis.

Salah satu sumber daya yang diperlukan dalam mengembangkan sistem pengenalan wicara adalah ketersediaan data suara yang kualitas baik yaitu memiliki tingkat derau yang rendah dan memuat semua fonem yang ada dalam suatu bahasa. Di samping itu perlu diperhatikan juga komposisi pembicara yang direkam seperti jenis kelamin, dialek, dan usia. Untuk dapat membuat sumber daya ini diperlukan dana yang banyak karena perekaman harus dilakukan di studio dengan prasyarat tertentu dan data yang dihasilkan masih perlu dilakukan pasca pemrosesan untuk memastikan kualitas dan pemadanan pemotongan kalimat yang diujarkan dengan skrip yang diucapkan. Kebutuhan dana yang besar ini merupakan salah satu kendala dari berkembangnya riset pengenalan wicara bahasa Indonesia. Oleh karena itu BPPT berinisiatif untuk mengembangkan korpus wicara bahasa Indonesia yang nantinya dapat digunakan secara gratis untuk tujuan penelitian dan pengembangan pengenalan wicara bahasa Indonesia non komersial di Indonesia.

Untuk memastikan bahwa korpus yang dihasilkan dapat digunakan untuk penelitian dan pengembangan pengenalan wicara bahasa Indonesia, korpus ini telah diujicobakan untuk menghasilkan model pengenal wicara bahasa Indonesia menggunakan *framework* Kaldi [1]. Kaldi mendukung model konvensional seperti GMM dan model dengan teknik DNN yang merupakan topik terkini

Submitted 24-07-2018 ; Revised 17-09-2018; Accepted 24-09-2018

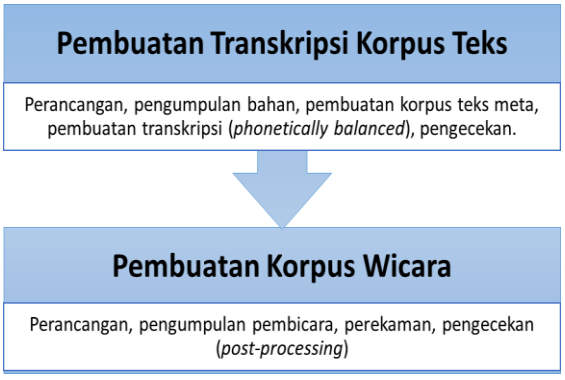
dalam *speech recognition*. Dalam sistem berbasis GMM-HMM, *states* HMM direpresentasikan sebagai GMM, dan tiap Gaussian direpresentasikan oleh *mean vector* dan sebuah diagonal *covariance matrix* [2]. Pada model dengan teknik DNN yang menggabungkan DNN dan HMM, distribusi probabilitas dalam HMM yang diwakili oleh GMM digantikan oleh DNN [3]. Dari uji coba pengenalan wicara korpus KDW-BPPT-2013 dengan menggunakan Kaldi diperoleh nilai Word Error Rate (WER) GMM 2,52 % dan DNN 1,64%.

II. PEMBUATAN KORPUS WICARA

Untuk selain bahasa Indonesia, khususnya bahasa-bahasa yang memiliki jumlah penutur banyak atau yang sudah banyak dilakukan penelitian/pengembangan mengenai sistem pengenalan wicara seperti bahasa Inggris, Jepang, Arab, dll., di Internet sudah banyak tersedia sumber-sumber yang bisa diunduh, sehingga peneliti/pengguna dapat langsung memanfaatkannya tanpa perlu melakukan tahapan-tahapan pembuatan korpus wicara yang memerlukan sangat banyak sumber daya waktu, manusia, dan dana. Berikut ini adalah daftar situs-situs yang menyediakan korpus wicara, antara lain untuk bahasa Inggris yaitu CSTR VCTK [4], LibriSpeech [5], TED-LIUM [6], Santa Barbara Corpus of Spoken American English[7], dan VoxForge [8], untuk bahasa Arab yaitu Arabic Speech Corpus [9], untuk bahasa Jepang yaitu UT-ML [10] dan PASL-DSR [11].

Secara garis besar, pada umumnya tahapan pembuatan korpus wicara dapat dibagi menjadi dua tahapan, yaitu:

- Pembuatan transkripsi korpus teks yang akan dibaca oleh pembicara sewaktu melakukan perekaman
- Pembuatan data perekaman suara.



Gambar 1. Proses Pembuatan Korpus Wicara

Tahapan pembuatan transkripsi korpus teks meliputi perancangan, pengumpulan bahan, pembuatan korpus teks meta, pembuatan transkripsi (*phonetically balanced*) dan pengecekan [12]. Sedangkan tahapan pembuatan data perekaman suara meliputi perancangan, pengumpulan pembicara, perekaman, dan pengecekan.

Tahapan perancangan pembuatan korpus teks meliputi dan tidak terbatas pada: penentuan jenis korpus, identifikasi fonem, dan pengumpulan/pembuatan program bantu untuk konversi dari format file sumber menjadi teks. Penentuan jenis korpus apakah berupa kumpulan kalimat berita, percakapan, atau hanya kumpulan kata-kata akan mempengaruhi proses pembuatan korpus wicara [13]. Proses pengumpulan bahan dilakukan secara online atau pun dengan melakukan pemindaian bahan-bahan *non-softcopy* yang kemudian dilanjutkan dengan mengubahnya menjadi teks dengan menggunakan perangkat lunak OCR (*optical character recognition*). Proses pembuatan korpus teks adalah pengumpulan bahan-bahan yang sudah ada menjadi satu korpus teks meta. Di sini, file-file berupa *softcopy* yang tidak dalam format teks diubah menjadi teks dengan menggunakan beberapa *tools* seperti pdf2text, antiword, dll.

Tahapan selanjutnya adalah normalisasi (mengubah simbol, angka, singkatan, dll. menjadi string teks), menghapus kalimat-kalimat yang memiliki kata-kata yang jarang diucapkan, dan pembuatan korpus teks yang *phonetically balanced*, sehingga hasil akhir adalah berupa korpus teks yang jumlahnya tidak begitu banyak, tetapi dapat merepresentasikan statistik kemunculan fonem-fonem dalam korpus teks meta.

Tahapan perancangan korpus wicara meliputi dan tidak terbatas pada pembuatan standar data perekaman, tatacara perekaman, penentuan jumlah pembicara yang disesuaikan dengan data demografi penutur, dan penentuan studio perekaman. Setelah data rekaman selesai dibuat, tahapan berikutnya adalah melakukan pengecekan kesesuaian antara data rekaman suara dan transkripsi teks. Di sini, kemungkinan perlu dilakukan proses segmentasi dan/atau penggabungan ujaran agar konten data suara sesuai dengan pasangan teksnya. Disarankan agar standar data perekaman suara adalah sama atau lebih daripada kualitas CD. Ada beberapa karya ilmiah seperti [14], [15], [16], dan [17] yang melaporkan isu-isu yang ditemui sewaktu mengembangkan korpus wicara untuk bahasa-bahasa yang tidak begitu banyak memiliki sumber daya kebahasaan (*under-resourced languages*) CD.

A. Pembuatan transkripsi korpus teks

Korpus teks *phonetically balanced* yang diucapkan pembicara pada saat perekaman dibuat dari sebuah korpus meta dengan menggunakan algoritma "*modified least-to-most greedy*"[18]. Korpus meta adalah berupa kumpulan kalimat sejumlah sekitar 9 juta kalimat unik yang dikumpulkan dari berbagai situs berita berbahasa Indonesia di internet. Korpus teks *phonetically balanced* berisi 5.000 kalimat (3.813 kalimat unik) dan 38.450 kata (10.930 kata unik), termasuk kata-kata berupa angka, simbol, dan singkatan yang dirubah dalam bentuk huruf.

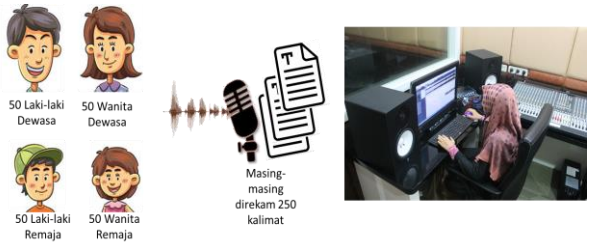
Dengan demikian, setiap kalimat dalam korpus teks *phonetically balanced* memiliki rata-rata 7~8 kata.

Jumlah pembicara yang direkam suaranya adalah 200 orang yang terdiri dari 100 orang dewasa (50 pria dan 50 wanita) dan 100 remaja (50 pria dan 50 wanita). Sebanyak 100 orang pembicara sengaja dipilih dari kalangan remaja karena semakin meningkatnya kecenderungan jumlah remaja yang menggunakan *gadget* dan komputer personal, sedangkan pembicara dewasa dipilih dari karyawan BPPT. Diharapkan agar karakteristik suara remaja dapat diperoleh melalui perekaman yang dilakukan. Pemilihan remaja yang direkam suaranya dilakukan melalui kerjasama dengan salah satu sekolah menengah pertama swasta di Jakarta.

B. Perekaman

Proses perekaman dilakukan di suatu studio yang sudah memiliki hubungan kerja dengan sekolah menengah swasta tersebut dengan beberapa pertimbangan termasuk faktor dana, waktu, ketersediaan sumber daya manusia, dan kedekatan hubungan antara studio rekaman dan sekolah menengah swasta tersebut.

Perekaman dilakukan secara bergantian karena terbatasnya ketersediaan ruangan perekaman. Sebelum melakukan sesi perekaman, masing-masing pembicara diminta untuk terlebih dahulu mempelajari teks yang akan diucapkan. Jadwal perekaman dilakukan melalui koordinasi dengan sekolah menengah pertama swasta, khususnya mengenai pembagian jadwal untuk siswa remaja.



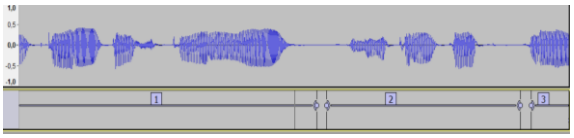
Gambar 2. Proses perekaman korpus wicara

Tatacara perekaman:

1. Perekaman dilakukan dalam ruang kedap suara.
2. Pintu masuk/keluar ruang perekaman harus dalam keadaan tertutup rapat.
3. Perekaman dilakukan secara bergantian sesuai keluangan waktu pembicara.
4. Pembicara diminta untuk diam beberapa saat sekitar 3 detik sebelum mengucapkan teks kalimat berikutnya.
5. Apabila ada kesalahan ucap, maka operator akan meminta pembicara untuk mengulang dari awal kalimat.
6. Pembicara diharuskan berada dalam keadaan sehat, setidaknya dalam kondisi yang tidak mengganggu kondisi suaranya.
7. Sebelum melakukan perekaman, jarak antara mikrofon dan pembicara serta posisi duduk pembicara diatur sedemikian rupa sehingga suara yang terekam tidak terlalu kecil/besar, tidak terekamnya suara hembusan nafas, dan pembicara merasa nyaman
8. Jarak antara mikrofon dan pembicara diusahakan agar tetap untuk memastikan kestabilan tingkat kekerasan (*loudness*)
9. Operator memberikan tanda baik secara lisan maupun gerakan kepada pembicara untuk memulai atau menghentikan perekaman
10. Apabila lelah, pembicara atau operator dapat meminta istirahat untuk beberapa menit.

C. Pasca-Pengolahan

Setelah proses perekaman data audio tersimpan dalam beberapa file data audio. Masing-masing berisi gabungan dari beberapa ujaran yang dipisahkan oleh segmen senyap. Untuk mendapatkan setiap ujaran sebagai file data audio tunggal, hasil proses pengeditan disegmentasi menggunakan Julius adintool [19]. Adintool menyegmentasikan file audio berdasarkan dua parameter: amplitudo ambang batas dan nilai *zero-cross*. Nilai *zero-cross* ditentukan dari berapa kali sinyal audio melewati nilai nol dengan mendeteksi perubahan dari nilai positif ke negatif atau sebaliknya pada data audio. Sinyal yang digunakan dalam proses pendeteksian hanya sinyal dengan amplitudo yang lebih besar dari nilai ambang batas yang telah ditentukan. Jika nilai *zero*-cross dalam periode waktu tertentu lebih besar dari nilai yang ditentukan, sinyal audio akan dianggap sebagai sinyal ujaran. Ini akan memicu tanda awal, jika tidak, sinyal akan dianggap sebagai sinyal senyap dan akan memicu penanda akhir dari ujaran. Sinyal data audio antara tanda awal dan penanda akhir akan disimpan sebagai satu file data audio yang mewakili satu ujaran.



Gambar 3. Proses segmentasi data rekaman menjadi data suara per kalimat (Julius Adintool) dengan parameter amplitudo dan nilai *zero cross*

File audio tersegmentasi ini kemudian diperiksa apakah sesuai dengan teks kalimat yang dibaca. Metode pemeriksaan menggunakan Perisalah (sistem ASR Indonesia) [20] untuk menghasilkan transkripsi pada setiap segmen audio, untuk kemudian dihitung jarak setiap hasil transkripsi terhadap kalimat transkrip yang seharusnya dipadankan. Jarak yang diukur adalah jarak sunting sederhana (*simple edit distance*), di mana metrik yang digunakan adalah jumlah sisipan, penghapusan, dan substitusi. Ambang nilai jarak digunakan untuk

menentukan apakah segmen file audio berpadanan dengan transkrip yang sesuai atau tidak. Dari proses pengecekan, juga dihasilkan informasi tentang rekomendasi penggabungan dua atau beberapa file data audio. Dua atau beberapa data audio direkomendasikan untuk digabungkan jika jarak sunting (*edit distance*) gabungannya lebih kecil daripada jarak sunting yang tidak digabungkan. Berdasarkan informasi yang dihasilkan, file yang tersegmentasi ini kemudian diolah untuk menghasilkan file yang siap digunakan sebagai bahan pembuatan model.

III. PENGUJIAN

A. Tentang Kaldi

Kaldi merupakan toolkit pengenal wicara yang bersifat otomatis dan sengaja dikembangkan untuk memenuhi kebutuhan akan *framework* yang berbasis FST (*finite-state transducer*), yang mendukung operasi aljabar linier dan yang dapat digunakan secara bebas tanpa dibatasi oleh masalah lisensi karena kode yang digunakan pada Kaldi di bawah lisensi Apache v2.0.

Kaldi menyediakan beberapa *recipe* [21] pengolahan data suara untuk menjadi model yang dapat digunakan untuk sistem pengenal wicara. Adapun variasi *Feature* yang digunakan: *Mel-Frequency Cepstral Coefficient* (MFCC), MFCC +*Delta*, *Preprosessing Feature: Linear Discriminant Analysis* (LDA) + *Maximum Likelihood Linear Transformations* (MMLT), dan *Objective Function*: *Maximum Mutual Information* (MMI) + MMI *Boosting*.

Fitur utama Kaldi:

1. Terintegrasi dengan FST
   Kaldi menggunakan OpenFST sebagai salah satu *library*. FST dan WFST memiliki berbagai fungsi untuk melakukan proses *parsing*. *Parsing* diperlukan untuk menguraikan teks menjadi komponen teks yang terkecil yang memudahkan dalam proses penyusunan kata/kalimat untuk pengenalan wicara.
2. Dukungan terhadap perhitungan numerik aljabar linier karena adanya *library* matriks yang berisi prosedur (*routine*) standar dari BLAS (*Basic Linear Algebra Subroutines*) dan LAPACK (*Linear Algebra PACKage*).
3. Rancangan yang dapat dikembangkan.
   Algoritma yang digunakan pada KALDI dibuat dalam bentuk yang generik. Sebagai contoh, dekoder dapat digunakan dengan antar muka yang memberi skor untuk *frame* dan simbol input FST tertentu. Dengan demikian, dekodernya dapat digunakan dari setiap sumber skor yang sesuai.
4. *Open Source*.
   Kode program yang digunakan berada di bawah lisensi Apache v2.0 yang merupakan salah satu aplikasi yang paling minimal keterbatasan lisensinya.
5. Kaldi memiliki *recipe* yang lengkap. Kaldi menyediakan *recipe* yang lengkap untuk membangun sistem pengenal wicara yang dapat berfungsi from *database* yang tersedia luas seperti yang disediakan oleh LDC (*Language Data Consortium*)
6. Kaldi memungkinkan pengujian bagi hampir semua kode program dengan menyediakan *routine* terkait sehingga proses pengujian dapat dilakukan secara rinci.
7. Kegunaan utama Kaldi adalah dalam penelitian model akustik dari pengenal wicara.

B. Pengujian Pengenalan Wicara

Pengujian pemanfaatan korpus KDW-BPPT-2013 sebagai data latih untuk pembuatan model bagi sistem pengenal wicara menggunakan Kaldi sebagai *framework*-nya. Kaldi telah memiliki banyak skrip yang dapat digunakan untuk mengolah berbagai jenis korpus wicara yang tersedia saat ini. Untuk mempercepat proses pengujian digunakan skrip pengolah korpus dari Voxforge sebagai basis untuk membentuk skrip yang dapat mengolah korpus KDW-BPPT-2013.

Langkah pertama yang dilakukan adalah membuat format file dan struktur direktori yang menyerupai korpus Voxforge dengan menggunakan KDW-BPPT-2013 sebagai datanya. Struktur direktori dari Voxforge menempatkan tiap-tiap pembicara dalam satu direktori tersendiri, dan dalam tiap direktori tersebut terdapat dua direktori. Direktori pertama adalah ‘etc’ yang didalamnya terdapat file-file berisi informasi dan teks dari file wicara, dan direktori kedua ‘wav’ yang berisi seluruh file wicara dari satu pembicara. Langkah berikutnya adalah melakukan modifikasi pada skrip pengolahan Voxforge agar dapat digunakan untuk mengolah KDW-BPPT-2013. Hal utama yang dilakukan adalah menyesuaikan nama direktori, nama bahasa, dan proses pembentukan leksikon (kamus ucapan). Pada KDW-BPPT-2013 leksikon dianggap sudah tersedia secara lengkap, OOV (*out-of-vocabulary*) tidak ditangani seperti halnya dalam pengolahan data Voxforge yang menggunakan *grapheme-to-phoneme* untuk menangani OOV. Hal lain yang dilakukan adalah mengadopsi skrip yang memungkinkan pengujian menggunakan DNN (*Deep Neural Network*) dalam hal ini yang digunakan adalah nnet2 dan nnet3.

Setelah langkah persiapan dilakukan maka pengujian kemudian dilakukan menggunakan skrip yang sudah dimodifikasi. Dalam proses pengujian terdeteksi beberapa file wicara yang mengalami kegagalan saat dilakukan proses *alignment* akibat dari kesalahan pada teks yang dipasangkan dengan file wicara tersebut, dalam pengujian ini yang dilakukan adalah mencatat file wicara yang bermasalah tersebut dan mengeluarkannya dari data latih. Dalam ujicoba ini terdapat 250 file wicara dari beberapa pembicara yang tidak digunakan. Khusus untuk percobaan

menggunakan nnet3 seluruh data wicara digunakan tanpa kecuali.

C. Evaluasi Hasil

Korpus KDW-BPPT-2013 ini telah berhasil digunakan untuk membuat model pengenal wicara bahasa Indonesia menggunakan Kaldi. Korpus dievaluasi dengan cara menggunakannya sebagai data latih dalam pembuatan model ASR berbasis Kaldi.

Kaldi memiliki beberapa *recipe* yang merupakan kombinasi dari *feature*, *pre-processing feature*, *objective function* dan atau HMM, GMM (atau DNN). Kombinasi ini menghasilkan beberapa *recipe* yang digunakan di Kaldi seperti:

1. Tri-phone *only* (tri1)[1] hanya menggunakan fitur MFCC.
2. Tri-phone + Delta (tri2a) [1] menggunakan kombinasi fitur MFCC dan Delta.
3. Tri-Phone+Delta+LDA +MLLT (tri2b) [1] menggunakan kombinasi fitur (MFCC dan Delta) dan fitur *pre-processing* (LDA dan MLLT).
4. Tri-Phone+Delta+LDA +MLLT+MMI (tri2b_mmi) [1] menggunakan kombinasi fitur (MFCC dan Delta), fitur *pre-processing* (LDA dan MLLT) serta *objective function* (MMI).
5. Tri-Phone+Delta+LDA +MLLT+MMI dengan *Boosting* (tri2b_mmi_b0.05) [1] menggunakan kombinasi fitur (MFCC dan Delta), fitur *pre-processing* (LDA dan MLLT) serta *objective function* (MMI dengan *boosting*).
6. Tri-Phone+Delta+LDA+MLLT+MPE (tri2b_mpe)[1] menggunakan kombinasi fitur (MFCC dan Delta), fitur *pre-processing* (LDA dan MLLT) serta menerapkan MPE (*Minimum Phone Error*).
7. Tri-Phone+Delta+LDA+MLLT+SAT (tri3b)[1] menggunakan kombinasi fitur (MFCC dan Delta), fitur *pre-processing* (LDA dan MLLT) serta SAT (speaker adaptive training).
8. Tri-Phone+Delta+LDA+MLLT+SAT+MMI (tri3b_mmi) [1] merupakan kombinasi fitur (MFCC dan Delta), fitur *pre-processing* (LDA dan MLLT) serta SAT dan MMI.
9. Graph-tri3b + iVector + DNN (nnet2)[2] merupakan kombinasi dari *graph* yang dihasilkan dari *recipe* tri3b dengan iVector dan DNN.
10. Graph-tri3b + TDNN (nnet3/tdnn_1)[2] merupakan kombinasi dari *graph* yang dihasilkan dari *recipe* tri3b dengan TDNN.

[1] Menggunakan GMM+HMM sebagai akustik modelnya
[2] Menggunakan DNN+HMM sebagai akustik modelnya

Tabel I menunjukan hasil uji coba korpus KDW- BPPT-2013 menggunakan *Kaldi recipe*.

TABEL I
HASIL DARI UJI COBA KALDI RECIPE

	Kaldi Recipe	**WER / SER (%)**
1	Tri-phone only (tri1)[3]	3.44 / 18.41
2	Tri-phone + Delta (tri2a) [1]	3.40 / 18.53
3	Tri-Phone+Delta+LDA +MLLT (tri2b) [1]	3.89 / 19.87
4	Tri-Phone+Delta+LDA +MLLT+MMI (tri2b_mmi) [1]	2.45 / 14.07
5	Tri-Phone+Delta+LDA +MLLT+MMI with Boosting (tri2b_mmi_b0.05) [1]	2.53 / 14.44
6	Tri-Phone+Delta+LDA +MLLT+MPE (tri2b_mpe) [1]	3.19 / 16.59
7	Tri-Phone+Delta+LDA +MLLT+SAT (tri3b) [1]	4.28 / 20.84
8	Tri-Phone+Delta+LDA +MLLT+SAT+MMI (tri3b_mmi) [1]	2.52 / 13.87
9	Graph-tri3b + iVector + DNN (nnet2)[4]	2.18 / 12.98
10	Graph-tri3b + TDNN (nnet3/tdnn_1)[2]	1.64 / 10.16

Sebagai informasi, dalam Kaldi, korpus wicara yang digunakan dalam proses pelatihan secara automatis dibagi menjadi dua yaitu korpus uji dan korpus latih. Pada uji coba ini ujaran 10 pembicara dari 200 pembicara yang ada di dalam korpus dipilih secara automatis menjadi korpus uji.

Hasil yang diperoleh sudah cukup memadai dengan nilai WER minimum 2,52 % menggunakan GMM dan 1,64 % menggunakan DNN. Hasil yang menggunakan DNN lebih baik dibanding yang menggunakan GMM, sejalan dengan hasil penelitian yang sudah dilakukan oleh grup peneliti Universitas Toronto, Microsoft Research (MSR), Google, dan IBM Research [3]. Dapat disimpulkan bahwa korpus KDW-BPPT-2013 sudah dapat digunakan sebagai bahan latih untuk pengembangan pengenal wicara bahasa Indonesia.

[1]Menggunakan GMM-HMM sebagai akustik modelnya
[2]Menggunakan DNN-HMM sebagai akustik modelnya

IV. KESIMPULAN

KDW-BPPT-2013 telah telah berhasil diujicobakan untuk pengembangan pengenal wicara bahasa Indonesia. Walaupun masih terdapat beberapa kesalahan dalam hal pemadanan teks dan ujarannya, secara umum korpus KDW-BPPT-2013 sudah dapat dimanfaatkan sesuai dengan target yang diharapkan. Selanjutnya korpus ini direncanakan dapat digunakan secara gratis untuk tujuan penelitian dan pengembangan pengenalan wicara bahasa Indonesia non komersial di Indonesia. BPPT mengharapkan pemanfaatan korpus ini dapat meningkatkan penelitian dan pengembangan di bidang NLP pada umumnya dan penelitian pengenalan wicara bahasa Indonesia pada khususnya.

REFERENSI

[1] D. Povey, et al. "The Kaldi speech recognition toolkit." IEEE 2011 workshop on automatic speech recognition and understanding. No. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[2] B. Popović, et al. "Deep neural network based continuous speech recognition for Serbian using the Kaldi toolkit." International Conference on Speech and Computer. Springer, Cham, 2015.

[3] H. Geoffrey, et al. "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups." IEEE Signal processing magazine 29.6 (2012): 82-97.

[4] J. Yamagishi (2010) "CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit". [Online]. Available: http://homepages.inf.ed.ac.uk/jyamagis/

[5] D. Povey (2015). "LibriSpeech ASR corpus". [Online]. Available: www.openslr.org/12/

[6] F. Fernandez, et al (2018). "Corpus: TED-LIUM Release 3". [Online]. Available: https://lium.univ-lemans.fr/en/ted-lium3

[7] UCSB Linguistics (2010). "Resources". [Online]. Available: http://www.linguistics.ucsb.edu/resources

[8] VoxForge (2009). "VoxForge Download" [Online]. Available: http://www.voxforge.org

[9] N Halabi (2018) "Arabic Speech Corpus" online available http://en.arabicspeechcorpus.com

[10] Speech Resources Consortium (2001), "University of Tsukuba Multilingual Speech Corpus (UT-ML)", online available http://research.nii.ac.jp/src/en/UT-ML.html

[11] Speech Resources Consortium (2015) "Spoken Language" and the DSR Projects Speech Corpus (PASL-DSR), online available: "http://research.nii.ac.jp/src/en/PASL-DSR.html

[12] Stănescu, Miruna, et al. "ASR for low-resourced languages: building a phonetically balanced Romanian speech corpus." Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European. IEEE, 2012.

[13] The Production of Speech Corpora, Florian Schiel, Christoph Draxler, Version 2.5: June 1, 2004 https://www.bas.uni-muenchen.de/forschung/BITS/TP1/Cookbook/Tp1.html

[14] M. Habib, F. Alam, R. Sultana, S.A. Chowdhury, M. Khan, "Phonetically balanced Bangla speech corpus," HLTD 2011, Alexandria, Egypt, 2011.

[15] S. Mandal, B. Das, P. Mitra, A. Bas, "Developing Bengali Speech Corpus for Phone Recognizer Using Optimum Text Selection Technique," IALP 2011, Penang, Malaysia, 2011.

[16] V. Pylypenko, V. Robeiko, M. Sazhok, N.Vasylieva, O. Radoutsky, "Ukrainian Broadcast Speech Corpus Development," SPECOM 2011, Kazan, Russia, 2011.

[17] A.A. Raza, S. Hussain, H. Sarfraz, I. Ullah, Z. Sarfraz, "Design and development of phonetically rich Urdu speech corpus," Oriental COCOSDA 2009, Urumqi, China, 2009.

[18] Suyanto, "Modified Least-to-Most Greedy Algorithm to Search a Minimum Sentence Set" in proc. TENCON, Hong Kong, 2006

[19] Lee, Akinobu, and Tatsuya Kawahara. "Recent development of open-source speech recognition engine julius." Proceedings: APSIPA ASC 2009: Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference. Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference, International Organizing Committee, 2009.

[20] Peluncuran "Perisalah". 2010. Retrieved July 10, 2016 from http://www.bumn.go.id/inti/berita/37/Peluncuran.

[21] Kaldi (2018). "kaldi-asr/kaldi". [Online]. Available: https://github.com/kaldi-asr/kaldi/tree/master/egs/wsj/s5/local