Sistem Pembangunan Korpus Suara Spontan Bahasa Indonesia

A System for Building Spontaneous Speech Corpus for Bahasa Indonesia

Muhammad Maulana Ramadhan[1], Dr. Suyanto, S.T., M.Sc.[2]

[1,2]Prodi S1 Teknik Informatika, Fakultas Teknik, Universitas Telkom

[1]maulanaramadh@gmail.com, [2]suyanto2008@gmail.com

Abstrak

Pengenalan Ucapan Kontinu Kamus Besar atau PUKKB merupakan sistem pengenalan suara yang paling mutakhir. Sistem ini mampu mengenali berbagai macam suara dan kata yang diucapkan orang. Kemampuan pengenalan tersebut didapat dengan melatih sistem menggunakan korpus suara membaca dan korpus suara spontan.

Korpus suara merupakan elemen penting dalam melatih sistem tersebut, terutama korpus suara spontan. Korpus ini menjadi referensi cara pengucapan bagi sistem tersebut. Beberapa bahasa seperti bahasa Inggris, sistem seperti ini mudah dikembangkan karena terdapat banyak korpus suara yang beredar, tetapi untuk beberapa bahasa seperti bahasa Indonesia, korpus suara yang beredar masih sedikit.

Dengan memadukan desain aplikasi pengumpul suara seperti Eyra, Woefzela dan Data Hound penulis mengembangkan aplikasi serupa untuk diaplikasikan dalam pembangunan korpus suara bahasa Indonesia, terutama dalam pengumpulan data suara.

Kata kunci: *Under-resourced*, korpus suara spontan, cakupan *triphone*, pembangkitan pertanyaan, *balanced sentence set*.

Abstract

Large Vocabulary Continous Speech Recognition System or LVCSR is state of art of speech recognition system. This system is capable at recognizing various speech and words uttered by a person. The recognizing capability comes from training the system with reading speech corpus and spontaneous speech corpus.

Speech corpus is an integral element in order to train the system, especially spontaneous speech corpus. This corpus is a pronounciation reference for the system mentioned. For several languages such as English, building such recognition system is relatively easy as there are many speech corpora available. However, for several languages such as Bahasa Indonesia, the speech corpus is scarce in number.

Combining the design that exists in Eyra, Woefzela and Datahound, we develop a similar application for building speech corpus to ease the corpus development, especially in speech data acquisition.

Keywords: Under-resourced, spontaneous speech corpus, triphone coverage, question generation, balanced sentence set.

1. Pendahuluan

Korpus suara spontan memiliki peran yang krusial, yaitu sebagai sumber referensi cara pengucapan bagi PUKKB. Sama seperti pembuatan korpus suara lainnya, korpus suara spontan tidak mudah dibuat, ini dikarenakan banyaknya waktu, biaya dan sumber daya lain yang diperlukan untuk membuat suatu korpus suara spontan yang *robust* [3].Tantangan membuat korpus suara spontan tersebut menjadi lebih sulit bagi bahasa yang *under resourced* [3].

Kriteria bahasa yang *under resourced* adalah sedikitnya jumlah korpus suara yang beredar bebas [3]. Bahasa Indonesia merupakan bahasa yang termasuk dalam kategori under resourced [2].

Terdapat beberapa aplikasi yang telah dikembangkan untuk mengatasi permasalahan *under resourced* contohnya Eyra [3], Woefzela [4] dan Datahound [5]. Datahound merupakan aplikasi berbasis Android yang mengumpulkan data dan secara otomatis mengunggahnya ke basis data yang tertambat pada *Google App Engine*. Woefzela memanfaatkan pendekatan *offline* dibandingkan pada Datahound, sehingga dapat digunakan pada lingkungan dengan koneksi internet yang terbatas, yaitu kondisi dimana koneksi internet sulit ditemukan atau memiliki kecapatan yang rendah. Kelemahan pada pendekatan ini adalah penambahan beban kerja pengunggahan data suara yang telah dikumpulkan. Berdasarkan arsitektur yang terdapat pada Datahound dan Woefzela dikembangkan alat yang bernama Eyra. Eyra menggabungkan kehandalan Datahound dalam menangani *user* dan kehandalan Woefzela dalam mengumpulkan dan menyimpan data suara. Eyra, Datahound dan Woefzela merupakan aplikasi untuk mengumpulkan suara membaca bukan suara spontan (penjelasan lebih lanjut mengenai suara membaca dan suara spontan dijelaskan pada subbab Suara Spontan dan Suara Membaca), sehingga untuk mengumpulkan suara spontan diperlukan adaptasi lebih lanjut. Adaptasi ini merupakan tujuan dari tugas akhir yang dilakukan.

Adaptasi yang dilakukan berupa pembangkitan pertanyaan dari kalimat pada dokumen *balanced sentence set*. *Balance sentence set* merupakan dokumen yang berisi kumpulan *triphone* yang harus ada pada suatu korpus suara, sedangkan pembangkitan pertanyaan berguna untuk mendapatkan data suara spontan.

2. Kajian Pustaka

2.1. Suara Spontan dan Suara Membaca

Berdasarkan pendapat Warner pada [6] suara spontan adalah suara yang diucapkan oleh manusia secara" tidak hati-hati". Tidak hati-hati disini adalah, tanpa memperhatikan struktur kalimat yang baku. Kalimat baku yang dimaksud adalah tidak mengandung jeda dan deham yang tidak perlu serta memiliki pengucapan yang sesuai dengan standar pengucapan. Kemudian selain ketidak hati-hatian, suara spontan juga mengandung banyak kata asing yang tidak ada pada bahasa tersebut [2].

Suara membaca adalah suara yang diucapkan oleh seseorang ketika membaca suatu kalimat. Cara pengucapan kalimat ditentukan oleh tanda baca, sehingga ucapan yang dihasilkan menjadi baku dan teratur. Deham dan jeda yang dihasilkan juga akan ikut teratur karena ditentukan oleh tanda baca.

2.2. Pembangunan Korpus Suara

Sebelum membangun korpus suara, tim pengembang mengumpulkan daftar *triphone* yang harus ada pada korpus suara tersebut [7]. Pengumpulan ini biasanya dilakukan dengan cara *crawling* pada web untuk mendapatkan suatu dokumen teks yang berisi semua *triphone* yang ada yang disebut dengan *Motherset*. *Motherset* berukuran besar dan memiliki komposisi *triphone* yang tidak seimbang maka dokumen tersebut disederhanakan menjadi dokumen lain yang disebut dengan *balanced sentence* set. *Balanced sentence set* berukuran lebih kecil dari *Motherset* tetapi memiliki komposisi *triphone* yang lebih seimbang dan memiliki komposisi *triphone* yang sama dengan *Motherset.*

Balanced sentence set ini kemudian digunakan untuk mempercepat akusisi data audio seperti pada [7]. Data yang diakuisisi kemudian ditranskripsi, disegmenteasi dan dianotasi oleh tim pembangun korpus suara. Setelah mengalami ketiga proses tadi, data audio kemudian diberikan *metadata* berupa umur, tinggi, berat badan, dialek serta informasi lainnya. Tujuan pemberian *metadata* ini adalah membantu melatih model akustik.

2.3. Pembangkitan Pertanyaan

Pembangkitan pertanyaan adalah sebuah teknik komputasi untuk membuat daftar pertanyaan dari teks yang disediakan. Pertanyaan dibuat dengan cara meneliti pola hubungan antar kata yang ada pada suatu teks.

Salah satu algoritma pembangkit pertanyaan adalah algoritma pada [9] yang membuat pertanyaan dari paragraf. Langkah awal yang dilakukan adalah mencari kalimat penting diantara kalimat isi paragraf. Kalimat isi yang layak kemudian dijadikan kandidat untuk dijadikan pertanyaan. Kelayakan suatu kalimat dinilai berdasarkan

1. Jumlah kata benda, kata sifat dan kata kerja yang sama pada kalimat isi dan kalimat judul. Kata benda, kata sifat dan kata kerja dalam konteks pembangkitan pertanyaan selanjutnya disebut dengan kata penting.
2. Letak kalimat apakah berada di akhir atau awal paragraf.

3. Panjang kalimat, kalimat dengan jumlah kata lebih besar sama dengan 4 dianggap layak.
4. Ada tidaknya kata hubung kalimat.

Setelah pemilihan kandidat dilakukan, setiap kalimat yang terpilih akan dilihat bentuk hubungan antara subjek dan objek. Jenis pertanyaan yang dibangkitkan tergantung pada bentuk hubungan yang teramati.

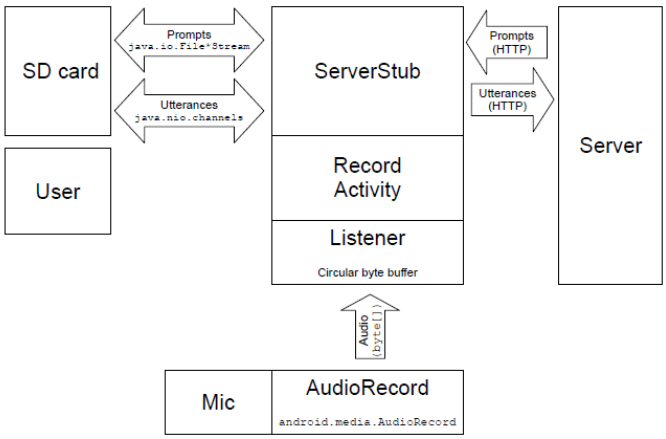
Tabel 2-5 Tipe pertanyaan yang dibangkitkan [9]

Subject	Object	Preposition	Question type
H	H		who,whom, what?
H	H	L	who, whom, what, where?
L	H		where, when?
C	C		How many?

Keterangan: H=Manusia, L=lokasi, C=*Count*

2.4. Eyra, Woefzela dan Datahound

Datahound merupakan aplikasi berbasis Android yang mengumpulkan rekaman suara dan secara otomatis mengunggahnya ke basis data yang tertambat pada Google App Engine. Arsitektur dari Datahound dapat dilihat pada gambar 2-2.

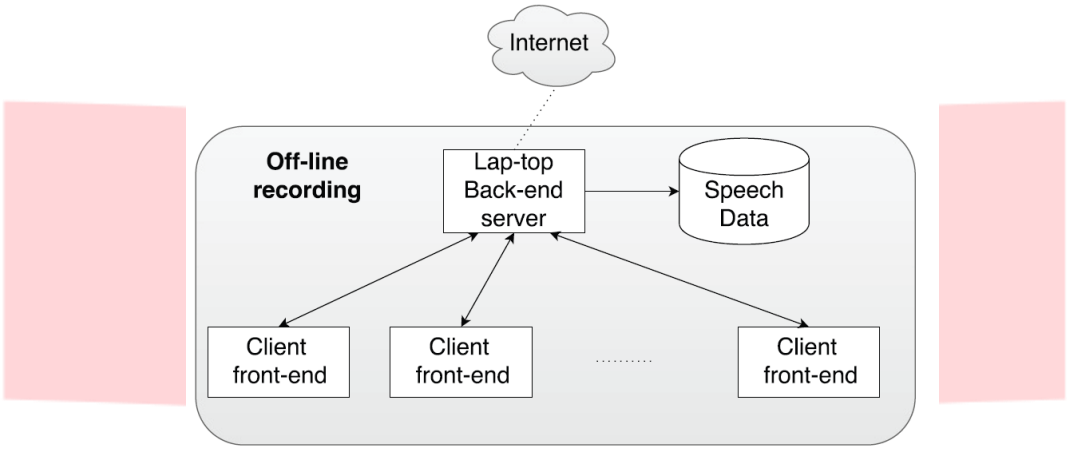


Gambar 0-1 Arsitektur Datahound

Datahound dapat mengumpulkan rekaman suara secara massal karena berbasis Android. Sejauh ini, Datahound telah digunakan untuk merekam korpus suara dengan total durasi 3000 jam dalam 17 bahasa [5], tetapi Datahound tidak dapat dipakai untuk mengumpulkan rekaman di lokasi yang minim koneksi internet [4] karena memiliki ketergantungan pada koneksi ke Google App Engine.

Dibandingkan dengan Datahound, Woefzela memanfaatkan pendekatan *offline* dengan menyimpan hasil rekaman pada kartu memori. Woefzela digunakan untuk mengumpulkan rekaman suara untuk bahasa yang ada di Afrika Selatan [4]. Kelemahan pada Woefzela adalah penambahan beban kerja pengunggahan data suara yang telah dikumpulkan [3].

Eyra dikembangkan berdasarkan kehandalan Datahound dalam menangani pengguna dan kehandalan Woefzela dalam mengumpulkan dan menyimpan data suara. Arsitektur Eyra dapat dilihat pada gambar 2-4.



Gambar 0-2 Arsitektur Eyra

Dari gambar diatas terlihat bahwa Eyra tergantung dengan jaringan internet untuk pengolahan data dan Eyra dapat menyimpan hasil rekaman sementara di dalam *back-end server*.

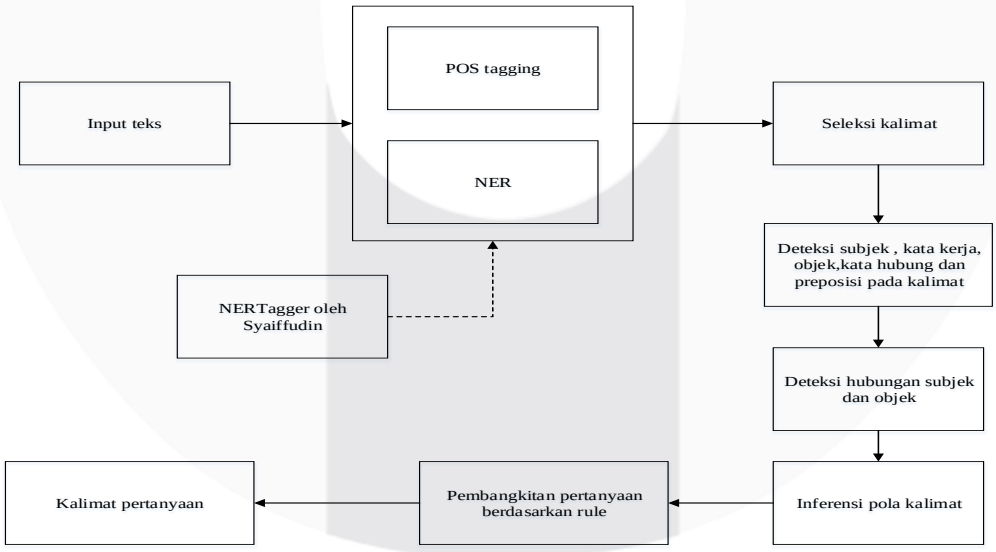
Ketiga aplikasi tersebut, Eyra, Woefzela dan Datahound mengumpulkan data suara dengan merekam suara orang yang membacakan suatu kalimat, sehingga dapat dikatakan bahwa ketiga aplikasi tersebut mengumpulkan suara membaca bukan suara spontan.

3. Desain Sistem

Input kalimat yang digunakan untuk membangkitkan pertanyaan adalah kalimat yang ada pada *balanced sentence set*. Alasan penggunaan *balanced sentence set* adalah *balanced sentence set* merupakan daftar *triphone* yang harus dikumpulkan untuk membangun suatu korpus suara. Algoritma pembangkitan pertanyaan yang digunakan adalah algoritma yang ada pada [9]. Algoritma ini membutuhkan input berupa paragraf, sedangkan kalimat pada *balanced sentence set* tidak membentuk suatu paragraf, sehingga untuk membentuk paragraf, kalimat pada *balanced sentence set* dikelompokkan menggunakan proses *clustering* menggunakan algoritma K-Means.

1.2. Desain Pembangkitan Pertanyaan

Secara umum proses pembangkitan pertanyaan dapat dilihat pada gambar 3-2.



Gambar 0-3 Cara kerja pembangkitan pertanyaan

Proses awal yang dilakukan adalah *POS tagging* dan *NER* dengan program dari [11] [12] untuk mengetahui peran tiap kata yang terdapat pada masukan. Masukan berupa kalimat *centroid* dan kalimat anggota. Ketika proses *POS tagging* dan NER selesai dilakukan, algoritma menyeleksi kalimat yang layak untuk dijadikan bahan pertanyaan. Kemudian algoritma mengidentifikasi kata kerja, subjek dan objek dari kalimat yang telah dipilih. Setelah subjek dan objek kalimat terdeteksi, algoritma mengkonversi entitas bernama subjek dan objek menjadi entitas bernama yang sesuai. Entitas sesuai yang dimaksud adalah manusia (dilambangkan dengan h) dan entitas selain manusia (dilambangkan dengan e).

Tujuan konversi ini adalah untuk mendeteksi hubungan antara subjek dan objek seperti pada algoritma pembangkitan pertanyaan pada [9]. Berdasarkan pola-pola yang sudah dideteksi, pertanyaan dibangkitkan berdasarkan aturan yang diadaptasi dari [9].

Algoritma pada [9] perlu diadaptasikan agar dapat berjalan dengan baik. Adaptasi yang dilakukan meliputi:

1. Aturan pembangkitan pertanyaan.
2. Penyederhanaan kalimat *balanced sentence set* dengan Deteksi Pola Kalimat secara Naif.
3. Pengubahan algoritma seleksi kalimat yang semula membandingkan irisan kata benda, kerja dan sifat pada kalimat isi dan kalimat judul menjadi irisan kata benda dan kata kerja pada *centroid*.
4. Perluasan aturan deteksi entitas kata benda berdasarkan *tag part of speech*.

Adaptasi pada poin 3 didasarkan pada ambiguitas kata benda yang dijelaskan oleh kata sifat pada *centroid*. Ambiguitas ini disebabkan karena *centroid* merupakan kalimat yang disusun dari *tf-idf* sehingga kata benda yang berada pada *centroid* tidak diletakkan untuk menjelaskan kata benda disekitarnya.

Detil perluasan aturan deteksi didasarkan pada POS *tag*, apakah POS *tag* kata benda mengarah kepada manusia seperti kata ganti orang atau tidak. Berdasarkan simbol pada alat deteksi entitas bernama yang digunakan, aturan deteksi yang digunakan dapat dilihat pada tabel 3-1.

Tabel 0-1 Aturan konversi POS *tag*

POS *Tag*	Entitas
NNP	H
NNPFC	H
PRP	H
NN	E
NNG	E
NNAC	E
NNFC	E

1.3. Adaptasi Algoritma Pembangkitan Pertanyaan

1.3.1. Deteksi Pola Kalimat secara Naif

Secara umum suatu rangkaian kata dikatakan sebagai kalimat jika mengandung subjek dan kata kerja. Berdasarkan prinsip diatas, algoritma Deteksi Pola Kalimat secara Naif akan mendeteksi kata kerja pertama yang ditemukan pada kalimat, kemudian bergerak ke awal kalimat mendeteksi subjek dan terakhir bergerak dari lokasi kata kerja ke akhir kalimat untuk mendeteksi objek. Algoritmanya secara lengkap adalah:

1. Deteksi kata kerja pertama yang ditemukan.
2. Dari posisi kata kerja, deteksi kata benda dari posisi sebelum kata kerja hingga awal kalimat. Kata benda pertama yang ditemukan menjadi subjek kalimat.
3. Dari posisi kata kerja, deteksi kata benda dari posisi sesudah kata kerja hingga akhir kalimat. Kata benda pertama yang ditemukan menjadi objek kalimat.

3.3.2. Pengembangan Aturan Pertanyaan

Mengembangkan aturan pada [9] aturan yang digunakan untuk kata tanya yang digunakan dan tipe pertanyaan adalah:

Tabel 3-0-2 Hubungan antara subjek dan objek serta kata tanya yang digunakan

Hubungan	Kata tanya yang dipakai		
	Target Subjek	Target Objek	Target Kata kerja
Manusia ke Manusia	Siapa	Siapa	Apa
Manusia ke Entitas	Siapa	Apa	Apa
Entitas ke Manusia	Apa	Siapa	Apa
Entitas ke Entitas	Apa	Apa	Apa

Tabel 3-0-3 Tipe pertanyaan untuk tiap pola kalimat yang dideteksi.

Pola kalimat	Target jawaban	Tipe Pertanyaan
S+V+O	Subjek, objek,kata kerja	Faktoid
S+V	Subjek	Faktoid
V+O	Objek	Faktoid
Kausalitas	Sebab	Faktoid,terbuka

Dengan S = Subjek, V=Kata kerja, O = Objek

Dari sumber [9] tipe pertanyaan faktoid adalah pertanyaan dengan kata tanya “apa” dan “siapa” yang langsung mengarah ke suatu pernyataan fakta. Contoh pada pertanyaan “Siapa yang membeli bayam?” yang dibuat dari kalimat “Ibu membeli bayam” akan langsung mengarah ke kata subjek “ibu”. Pertanyaan terbuka pada penelitian ini merupakan pertanyaan dengan kata tanya “mengapa” yang mengarah pada sebab suatu kejadian. Contoh pada pertanyaan “Mengapa Juli menangis?” yang dibuat dari kalimat “Juli menangis karena anjingnya meninggal” akan mengarah pada sebab Juli menangis yaitu anjingnya meninggal. Kausalitas merupakan tanda bahwa suatu kalimat dapat digunakan sebagai pertanyaan terbuka. Kalimat yang merupakan kausalitas ditandai dengan munculnya kata “karena”,”sebab” dan ”pasalnya”. Merujuk pada [9] kalimat yang digunakan untuk membangkitkan pertanyaan pada kalimat kausalitas adalah kalimat pertama sebelum kata “karena”,”sebab” dan “pasalnya”.

Pola S+V dan V+O dapat digunakan karena kalimat pola S+V dapat membangkitkan pertanyaan yang mengarah ke subjek dan kalimat V+O dapat digunakan untuk membangkitkan pertanyaan yang mengarah ke objek.

3.4. Seleksi Pertanyaan yang Dibangkitkan

Pertanyaan yang dibangkitkan dapat mengalami cacat contohnya

1. Apa yang dilakukan Lho kepada dong?
2. Apa yang dilakukan iya kepada helm?
3. Mengapa AirAsia ditermasuk?
4. Apa yang dimenjabat oleh ia?

Kesalahan pada pertanyaan 1 adalah kata “Lho” dan kata “dong” yang bukan merupakan kata subjek dan kata objek. Pada pertanyaan kedua, kata “iya” bukan merupakan subjek. Pada pertanyaan ketiga, terjadi kesalahan pada kata kerja “ditermasuk” yang seharusnya menjadi “termasuk”. Pada pertanyaan keempat, terjadi kesalahan pada kata “dimenjabat”. Kesalahan pada pertanyaan-pertanyaan diatas terjadi karena *error* pada proses POS *tagging*.

Solusi untuk menghindari cacat tersebut, kalimat pertanyaan yang dibangkitkan diseleksi menggunakanlah model klasifikasi kalimat pertanyaan. Model klasifikasi ini bertujuan untuk menseleksi kalimat pertanyaan yang tidak cacat dari kalimat pertanyaan yang dihasilkan. Model yang digunakan adalah model probabilistik Naïve Bayes. Alasan penggunaan model ini adalah peluang kata-kata yang muncul pada pertanyaan yang cacat saling independen satu sama lainnya. Model ini juga sering digunakan sebagai model untuk klasifikasi teks. Model Naïve Bayes pada penelitian ini selanjutnya disebut dengan model penyeleksi.

4. **Pengujian**

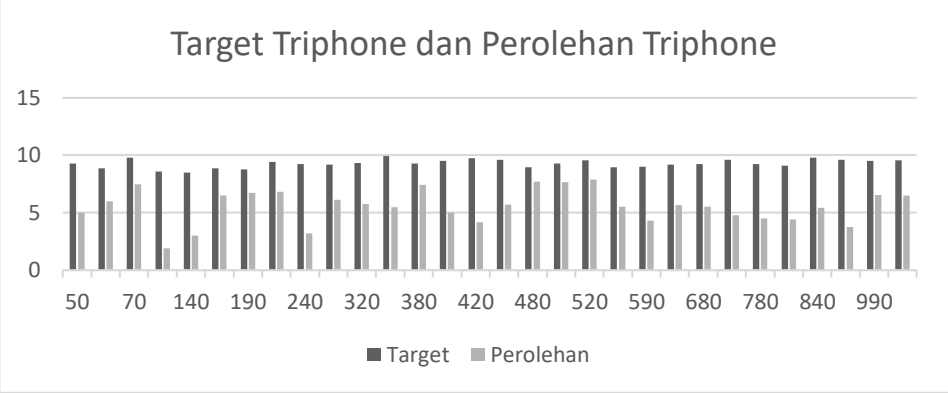
Untuk mengetahui performa sistem dilakukan pengujian berupa berapa banyak *triphone* yang diperoleh dari sampel dibandingkan dengan *triphone* yang ditargetkan. Skema pengujian yang dilakukan adalah:

1. Lakukan *clustering* dari K=x hingga K=y untuk mendapatkan kandidat sampel. Nilai x dan y dipilih masing-masing 50 dan 1000 untuk mempermudah pengambilan sampel.
2. Pilih secara acak hasil *clustering* yang telah dilakukan dan *cosine similiarity*-nya.
3. Ambil n kalimat secara acak dari hasil *clustering* tersebut untuk dijadikan masukan bagi pembangkitan pertanyaan. Untuk memudahkan pengambilan sampel, pilih n = 180.
4. Hitung jumlah triphone unik yang ada pada kalimat yang dipilih. Bandingkan dengan triphone unik yang ada pada *balance sentence set* dan catat dalam persentase. Persentase yang dihasilkan adalah target *triphone* yang ingin diperoleh.
5. Ambil data suara berdasarkan pertanyaan yang telah dibuat. Untuk memudahkan pengambilan data suara dan menghindari bias, total durasi data suara yang dikumpulkan berdurasi kurang lebih 10 menit.
6. Sampel yang telah didapat ditranskripsi menggunakan Google Speech API, lalu dihitung cakupan *triphone*-nya. Cakupan *triphone* dihitung dengan mencari irisian *triphone* unik pada transkripsi dan *triphone* unik pada *balanced sentence set*. Irisan yang didapat diubah ke dalam bentuk persentase. Persentase yang dihasilkan menjadi *triphone* yang diperoleh.

4.1. Analisa Pengujian

4.1.1. Perolehan Triphone dan Target Perolehan Triphone

Dari pengambilan data, didapat 30 sampel. Cakupan *triphone* serta target cakupan *triphone* pada tiap sampel disajikan pada gambar 4-1.



Gambar 0-4 Target Triphone dan Perolehan Triphone

Dari pengujian yang dilakukan terdapat ketimpangan antara cakupan *triphone* yang diinginkan dan yang diperoleh, yang ditunjukkan dengan variabel target *triphone* dan perolehan *triphone* pada gambar 4-1. Ketimpangan yang terjadi diduga karena ada kesalahan dalam proses *clustering*.

Berdasarkan dugaan bahwa ada kesalahan pada proses *clustering* dilakukan penyelidikan tentang korelasi *clustering* dengan cakupan *triphone* dilanjutkan dengan ketepatan *clustering* yang dilakukan. Dari penyelidikan tentang korelasi didapatkan bahwa *clustering* berkorelasi positif yang ditunjukkan dengan korelasi positif antara *cosine similiarity* dengan cakupan *triphone.*

Penyelidikan tentang ketepatan *clustering* dilakukan untuk mengetahui apakah *clustering* yang dilakukan dapat menghasilkan *cluster* dengan *cosine similiarity* yang besar. Dari penyelidikan ketepatan *clustering* diketahui bahwa proses *clustering* tidak bisa dioptimasi lagi, karena kalimat di dalam *balanced sentence set* memiliki keterkaitan yang kecil antara satu dengan kalimat lainnya. Keterkaitan tersebut dilihat dari nilai *singular* yang didapat pada LSA yang memiliki harga yang kecil.

5. Kesimpulan Dan Saran

Dari skema pengujian dan pembahasan yang telah dilakukan dapat diketahui bahwa:

1. *Cosine similiarity* berhubungan positif terhadap cakupan *triphone*.
2. Kalimat pada *balanced sentence set* memiliki keterkaitan yang kecil dengan kalimat lain pada dokumen *balance sentence set* tersebut.

Sedangkan dari pembahasan yang telah dilakukan dapat diambil saran untuk pengembangan lebih lanjut yaitu:

1. Dokumen *balanced sentence set* yang digunakan dapat diperbaiki lebih lanjut dengan menyertakan informasi tambahan mengenai kalimat-kalimat pada *balanced sentence set*. Informasi tambahan yang dimaksud adalah *metadata* berupa kata-kata kunci pada kalimat *balanced sentence set*. Sehingga akan memudahkan proses *clustering*.

DAFTAR PUSTAKA

[1] S. H. Suyanto, "Design of Indonesian LVCSR Using Combined Phoneme and Syllable Models," in *7th International Conference on Information & Communication Technology and Systems (ICTS)*, Bali, 2013.

[2] D. P. Lestari, "A Large Vocabulary Continuous Speech Recognition System," in *15th Indonesian Scientific Conference in Japan Proceedings*, Hiroshima, 2006.

[3] K. S. G. J. Petursson Matthias, "Eyra - Speech Data Acquisition System for Many Languages," in *5th Workshop on Spoken Language Technology for Under-resourced Languages*, Yogyakarta, 2016.

[4] B. J. H. D. M. ,. B. E. d. W. A. Nic J. de Vries, "Woefzela - An open-source platform for ASR data collection in the developing," in *Interspeech*, Florence, 2011.

[5] N. K. L. L. L. L. Hughes Thad, "Building transcribed speech corpora quickly and cheaply for many languages," in *INTERSPEECH 2010*, Makuhari, 2010.

[6] N. Warner, "Methods for Studying Spontaneous Speech," in *The Oxford Handbook of Laboratory Phonology*, Oxford, Oxford University Press, 2012, pp. 621-633..

[7] K. Samudravijaya, "Development of Multi-lingual Spoken Corpora of Indian Languages," in *International Symposium on Chinese Spoken Language Processing (ISCSLP 2006)*, Singapore, 2006.

[8] V. C. Sharada C. Sajjan, "Speech Recognition Using Monophone and Triphone Based Continuous Density Hidden Markov Models," in *2nd International Conference on Multidisciplinary Research & Practice*, Gujarat, 2015.

[9] I. J. I. Swali Dhaval, "Automatic Question Generation from Paragraph," in *International Journal of Advance Engineering and Research*, India, 2016.

[10] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53-65, 1987.

[11] Y. Syaifudin, *QUOTATIONS IDENTIFICATION FROM INDONESIAN ONLINE NEWS USING RULE-BASED METHOD*, Yogyakarta: UGM, 2016.

[12] M. Fachri, *NAMED ENTITY RECOGNITION FOR INDONESIAN TEXT USING HIDDEN MARKOV MODEL*, Yogyakarta: UGM, 2014.