

SISTEM KONVERSI TEXT TO SPEECH PADA DELPHI MENGGUNAKAN METODE FINITE STATE AUTOMATA

Rizal ⁽¹⁾, Annahl Muharram Tresnandhy ⁽²⁾

Program Studi Teknik Informatika

Universitas Malikussaleh

e-mail : rizal@unimal.ac.id⁽¹⁾

Abstrak

Pada era modern ini, kecanggihan komputer dalam pengolahan data sangat beragam diantaranya pengolahan suara. Pengolahan suara menggunakan berbagai macam metode. Salah satunya pengubah tulisan menjadi suara atau biasa dikenal dengan *Text To Speech*. *Text To Speech* merupakan alat pengubah inputan tulisan menjadi suara dengan mengubah *phonem* tulisan. *Text To Speech* terdiri dari dua sub sistem yakni konverter kata ke *phonem* dan konverter *phonem* ke ucapan. *Phonem* itu sendiri terdiri dari pecahan kata yang memiliki *speech* berbeda satu dengan yang lainnya. Untuk melakukan proses pemisahan *phonem* digunakan metode *Finite State Automata*. *Finite State Automata* dapat mengenali bahasa paling sederhana dan merupakan mesin abstrak yang memasukkan dan keluarannya berbentuk diskrit. Sistem *input* yang diproses oleh sistem hanya terdiri dari dua buah kata. Apabila lebih dari dua *input*-an kata maka program tidak dapat melakukan pemisahan kata dengan baik. Setelah pemisahan kata dilakukan, maka sistem melakukan konverter *phonem* ke ucapan. Ucapan yang dihasilkan diproses oleh database Indo_TTS yang dikembangkan dengan menggunakan metode prosodi *diphone concatenation*. Namun Indo_TTS memiliki kekurangan tidak bisa membedakan kata *homograf* yaitu penulisan sama tetapi bacaan dan artinya berbeda. Kata yang telah dipisah akan disimpan kedalam database MySQL sebagai hasil kata yang telah diuji oleh

sistem. Nantinya sistem ini akan diperuntukkan kepada masyarakat umum, pelajar maupun penderita tuna wicara.

Kata Kunci : *Text To Speech, Phonem, Finite State Automata, Indo_TTS, Homograf*

Abstract

In this modern era, the sophistication of computers in data processing is very diverse including voice processing. Sound processing uses a variety of methods. One of them is a voice converter into sound or commonly known as Text To Speech. Text To Speech is a tool convert written input into voice by changing phoneme writing. Text To Speech consists of two sub-systems ie word to phoneme converter and phoneme converter to speech. Phoneme itself consists of fractions of words that have different speech one with the other. To perform phoneme separation process used Finite State Automata method. Finite State Automata can recognize the simplest language and is an abstract engine that inserts and outputs are discrete. The input system processed by the system consists of only two words. If more than two inputs of a word then the program cannot do a good word separation. After the word separation is done, the system performs a phoneme converter to speech. The resulting speech is processed by an Indo_TTS database developed using the pros and cons method of diphone concatenation. But Indo_TTS has the disadvantage of not being able to distinguish the word homograf ie the same writing but the reading and the meaning is different. The separated word will be saved into the MySQL database as a result of a word that has been tested by the system. Later this system will be intended for the general public, students and people with speech impaired.

Keywords : *Text To Speech, Phoneme, Finite State Automata, Indo_TTS, Homograf*

1. PENDAHULUAN

Dewasa ini, komputer tidak terdengar asing lagi, melainkan sudah merupakan kebutuhan bagi beberapa orang. Komputer sendiri digunakan sebagai alat untuk mempermudah pekerjaan seseorang maupun beberapa perusahaan ternama, tidak hanya untuk mempermudah pekerjaan, melainkan dapat digunakan juga

sebagai alat pencipta aplikasi lainnya seperti halnya aplikasi *translator* bahasa, komputasi perhitungan dan alat deteksi suara (Lee, 1989). Banyak teknologi yang telah dikembangkan untuk membantu mereka berkomunikasi dengan orang lain maupun gadget, salah satunya adalah *Text-To-Speech* (TTS) Bahasa Indonesia. TTS adalah sistem yang dapat mengubah suatu teks menjadi ucapan secara otomatis dengan cara fonetisasi (penyusunan fonem-fonem untuk membentuk ucapan) (Wijaya, Susanto, & Salman, 2013). Dengan teknologi TTS, dimungkinkan sebuah komputer mampu berkomunikasi dan berinteraksi dengan manusia tidak hanya melalui tulisan, namun juga dalam bentuk lisan menggunakan bahasa yang digunakan sehari-hari (Adriati W., Tolle, & Setyawati, 2016).

TTS Bahasa Indonesia yang banyak dibuat dan dikembangkan saat ini berbasis komputer, yang memiliki beberapa kelemahan antara lain ukuran PC yang relatif besar dan berat, sehingga menjadikan kurang praktis untuk dibawa ke mana saja. Padahal dalam perkembangan teknologi, alat penunjang yang paling banyak digunakan untuk membantu menjalankan atau menyelesaikan pekerjaan manusia saat ini adalah *smartphone*. Salah satu platform pada *smartphone* yang paling banyak digunakan oleh masyarakat Indonesia adalah platform Android, dikarenakan fitur-fitur yang tersedia dalam *smartphone* ini dapat memuaskan kebutuhan para konsumen. Dengan kemampuan tersebut, beberapa pakar *profesional* terdorong untuk mengembangkan beberapa aplikasi untuk memudahkan komunikasi antar *user* dan komputer menggunakan suara. Namun tidak suara dari *user* agar dibaca komputer, melainkan Penulis ingin kalimat yang dituliskan *user* disuarakan oleh komputer.

Suara sendiri merupakan gelombang *longitudinal* yang terpantul oleh benda padat, cair, maupun gas dan menghasilkan bunyi, dengan kemampuan komputer yang dapat menghasilkan bunyi (Dutoit, 1997). Penulis berinisiatif untuk membuat sebuah sistem aplikasi konversi *texttospeech* (tulisan ke suara) dengan

menggunakan metode *Finite State Automata (FSA)* dikarenakan FSA sendiri sangat cocok dalam melakukan pemenggalan kata terhadap *program* yang akan penulis buat nantinya. FSA sendiri adalah mesin abstrak yang dapat mengenali pola bahasa secara sederhana dan dapat diimplementasikan secara nyata (Nugroho, 2013).

Penulis merasa pembangunan sistem konversi *texttospeech* ini sangat diperlukan, karena dalam aplikasi ini akan ada beberapa fungsi yang akan sangat memudahkan penggunaannya dalam melakukan pemenggalan suku kata seperti halnya yang dipelajari dalam berbahasa Indonesia. Terutama kepada anak-anak yang sedang mempelajari materi pemenggalan suku kata didalam mata pelajaran mereka. Tidak hanya itu aplikasi ini juga berfungsi sebagai alat pengubah yang ditujukan kepada pengguna tuna wicara (bisu) agar mereka mudah dalam mengutarakan apa yang ingin mereka sampaikan dan orang asing yang ingin mempelajari bahasa Indonesia bagaimana tata cara berucap kata yang benar pada kata bahasa Indonesia.

2. Metode Penelitian

Adapun dalam membangun sistem konversi *texttospeech* ini, penulis menggunakan beberapa komponen dan *database* tambahan dari pihak lain, diantaranya:

- a) *Database* Suara Indo_TTS (Arman, 2003). IndoTTS menggunakan teknik *Diphone Concatenation*, salah satu teknik terbaik yang ada untuk membuat TTS berkualitas tinggi. *Diphone database* yang digunakan oleh IndoTTS adalah "id1". IndoTTS dilengkapi dengan model intonasi bahasa Indonesia. Tidak mudah untuk mengembangkan model intonasi bahasa Indonesia karena *data* linguistik bahasa Indonesia masih terbatas.

- b) *Database MySQLXAMPP* versi 3.0.12
Dalam pembuatan tabel, Penulis menggunakan *database MySQL* oleh Xampp. *MySQL* adalah *database* yang multi fungsi, artinya *database* ini dapat digunakan oleh berbagai

macam bahasa pemrograman seperti *java*, *php*, *Visual basic* atau *delphi*. Karena kecanggihannya *MySQL* sering menjadi pilihan dalam membangun aplikasi yang menampung banyak *data*. kabarnya untuk *data* yang dapat ditampung oleh *database* ini hingga ratusan ribu *data*.

c) *Component AlphaSkin* versi 7

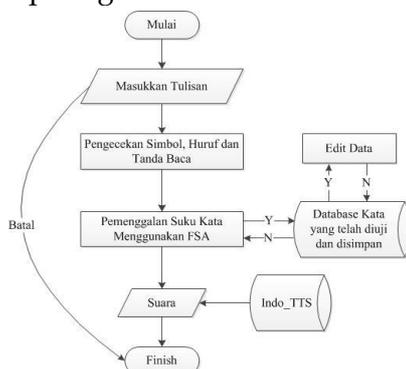
Komponen *AlphaSkin* merupakan komponen tambahan pihak ketiga pada *development tool Delphi* yang berfungsi sebagai alat mempercantik tampilan *User Interface*.

d) *IcoFX*

Aplikasi *IcoFX* penulis gunakan sebagai pembuat *icon* pada aplikasi sistem konversi *texttospeech* yang penulis bangun. Dari berbagai gambar yang penulis ambil dari bermacam sumber seperti *google image*, *yahoo* dan lain-lain. Kemudian gambar yang memiliki ekstensi **JPG*, **JPEG*, **GIF*, dan lain-lain penulis ubah menggunakan aplikasi *IcoFX* ini menjadi gambar yang berekstensi **ICO* agar dapat ditambahkan dalam aplikasi yang penulis bangun menggunakan *delphi*.

2.1. Skema sistem aplikasi keseluruhan

Sistem yang dibangun akan bekerja sebagaimana yang digambarkan seperti gambar berikut:



Gambar 1. Skema Sistem

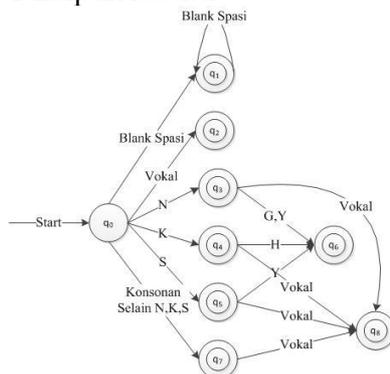
Pada skema sistem diatas, dapat kita lihat saat *program* sistem konversi mulai dijalankan, Pengguna mengetikkan beberapa kata yang akan dikonversikan menjadi suara. Setelah kata dimasukkan, lalu *program* melakukan pengecekan apakah kata yang telah diinputkan termasuk kedalam kriteria yang telah dibentuk dalam sistem aplikasi tersebut, jika iya, maka *program* akan melakukan pemenggalan suku kata. Jika tidak, *program* akan terhenti atau tidak berjalan.

Setelah melakukan pemenggalan suku kata, pengguna dapat menyimpan atau tidak kata yang telah diuji kedalam *database* sistem. Pengguna juga dapat mengubah *data* yang telah tersimpan didalam *database* kata. Setelah proses simpan menyimpan, Pengguna dapat melakukan Konversi kesuara dengan menekan tombol yang tersedia. Suara diambil dari *database* Indo_TTS yang telah tertanam didalam sistem dan sistem pun diakhiri.

2.2. Skema Finite State Automata

Dalam mengenali bahasa *regular* bahasa Indonesia, *Diagram* FSA memiliki tiga tahapan pengklasifikasian untuk membedakan jenis suku kata dalam bahasa Indonesia, yakni tahapan kesatu, tahapan kedua dan tahapan ketiga.

a) Tahapan kesatu



Gambar 2. Diagram FSA tahap kesatu (Isrokah & Yasin, 2013)

Keterangan:

q_0 = Status awal.

q_1 = Mengenali spasi.

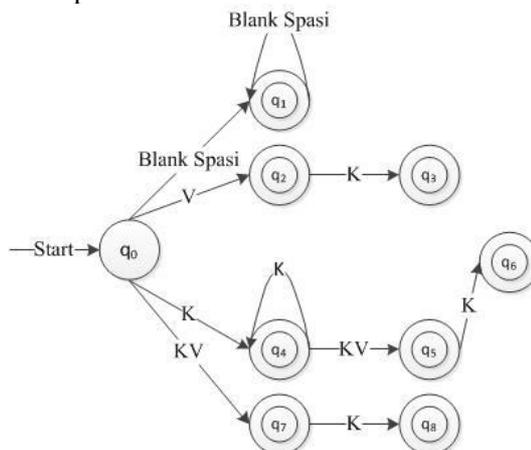
q_2 = Mengenali vokal.

q_3, q_4, q_5, q_7 = Mengenali konsonan.

q_6 = Mengenali konsonan yang terdiri dari dua huruf.

q_8 = Mengenali konsonan-vokal.

b) Tahapan kedua



Gambar 3. Diagram FSA tahap kedua (Isrokah & Yasin, 2013)

Keterangan:

q_1 = mengenali spasi

q_2 = mengenali V

q_3 = mengenali VK

q_4 = mengenali K

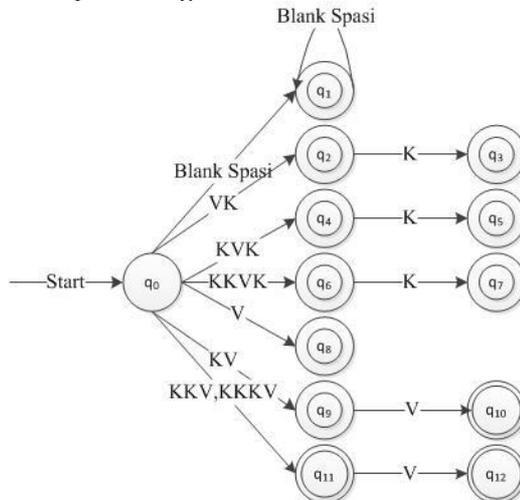
q_5 = mengenali KKV atau KKKV

q_6 = Mengenali KKVK atau KKKVK

q_7 = mengenali KV

q_8 = mengenali KVK

c) Tahapan ketiga



Gambar 4. Diagram FSA tahap keempat (Isrokah & Yasin, 2013)

Keterangan:

q1= mengenali spasi/blank

q2= mengenali VK

q3= mengenali VKK

q4= mengenali KVK

q5= mengenali KVKK

q6= Mengenali KKVK

q7= mengenali KKVKK

q8= mengenali V

q9= mengenali KV

q10= mengenali KVV

q11= mengenali KKV/KKKV

q12= mengenali KKVV/KKKVV

Hasil klasifikasi suku kata tersebut, dapat disimpulkan bahwa suku kata bahasa indonesia dapat dibagi dalam 12 kelas yaitu: spasi, VK, VKK, KVK, KVKK, KKVK, V, KV, KVV, KKV/KKKV dan KKVV/KKKVV (Isrokah & Yasin, 2013).

3. Hasil dan Pembahasan

Cluster yang telah dihasilkan oleh *algoritma Finite State Automata* dalam penelitian ini dibagi kedalam sebelas cluster sebagaimana urutan berikut:

- a) Cluster "V" disini diartikan bahwa sistem mengenali suku kata yang terdiri dari satu huruf vokal tunggal yang berdiri sendiri dan dapat membedakan arti.
Misalnya **a**-yam, **u**-dang, **i**-kan, **e**-mak, dan **o**-rang.
- b) Cluster "VK" disini diartikan bahwa sistem mengenali suku kata yang terdiri dari dua huruf, vokal dan konsonan yang saling berdampingan.
Misalnya **an**-da, **un**-ta, **as**-ma, **ir**-fan, **em**-bat, **ob**-li-ga-si.
- c) Cluster "KV" merupakan pengklasifikasian kebalikan dari cluster "VK".
Misalnya **ha**-ri, **li**-ma, **ba**-lon.
- d) Cluster "KVK" adalah cluster yang terdiri dari tiga huruf, vokal, konsonan dan vokal dalam satu kata.
Misalnya: ma-**kan**, **pin**-tu, o-**tak**.
- e) Cluster "KKV" dibaca oleh sistem ketika inputan yang diterima merupakan tiga buah huruf yang terdiri dari konsonan, konsonan, dan vokal didalam satu kata.
Misalnya: in-**fra**, **pra**-ja, **sas**-tra.
- f) Cluster "KKVK" dimana sistem membaca satu kata yang terdiri dari empat buah huruf konsonan, konsonan, vokal, dan konsonan dalam satu kata yang kemudian kata tersebut dijadikan dalam satu satuan suku kata.
Misalnya: **blok**, **prak**-tek, **trak**-tor.
- g) Cluster "VKK" merupakan suku kata yang terdiri dari tiga buah huruf yaitu vokal, konsonan, dan konsonan, biasanya suku kata ini terdapat pada kata-kata serapan dari bahasa asing.
Misalnya **eks**-po-nen, **ons**.
- h) Cluster "KVKK" adalah kelas suku kata yang terdiri dari konsonan, vokal, konsonan, konsonan.

Misalnya kam-**pung**, kon-**teks**, bung-**lon**.

- i) Cluster “KKVKK” dimana sistem membaca satu kata yang terdapat didalamnya konsonan, konsonan, vokal, konsonan, konsonan dan menjadikannya sebagai satu suku kata.

Misalnya kom-**pleks**.

- j) Cluster “KKKV” suku kata yang terdiri dari konsonan, konsonan, konsonan dan vokal pada akhiran suku katanya. Misalnya **stra**-te-gi, in-**stru**-men.

- k) Dan yang terkahir cluster “KKKVK” yang terdiri dari konsonan, konsonan, konsonan, vokal dan konsonan.

Misalnya in-**fra-struk**-tur, in-**struk**-si.

3.1. Implimentasi Sistem

Saat sistem dimulai, *user* memasukkan kalimat berupa *file* bertipe *.txt ataupun mengetiknya secara manual didalam *box memo* aplikasi yang telah disediakan, setelah itu sistem akan mengkoreksi apakah kalimat yang dimasukkan sesuai kondisi yang telah dibuat oleh sistem, jika iya sistem akan berjalan sesuai *algoritma*.



Gambar 5. Form inputan di sistem

Deteksi *Null* diatas merupakan variabel yang menandakan apakah didalam sistem aplikasi terdapat teks atau tidak, jika terdapat teks maka sistem akan menjalankan *algoritma* yang telah tersusun seperti diatas, dan jika tidak

maka sistem akan langsung mengakhirinya. Huruf **N**, **Y**, dan **G** dipisahkan dari suku konsonan dikarenakan agar memudahkan untuk pengkelasan suku kata seperti huruf diftong.

3.2. Pengujian system

Dari kesebelas kelaster yang telah diuji masing-masing satu kata untuk mewakili kata yang lain, dapat disimpulkan untuk pemisahan suku kata dan pembacaan kata yang telah dipisah masih cukup baik diproses oleh sistem yang telah dibangun oleh Penulis. Metode yang telah ditanam kedalam sistem sebagai dasar pemenggalan suku kata pada *program* tidak lari dari apa yang telah di rancang. Pemenggalan tetap menggunakan empat cara yang telah dibahas pada bab dua yakni:

- a. Jika di tengah kata ada vokal yang berurutan, pemenggalan itu dilakukan diantara kedua huruf vokal. Contoh: Ma-in, sa-at, bu-ah, dan lain-lain. Huruf diftong ai, au, dan oi tidak pernah diceraikan sehingga pemenggalan kata tidak dilakukan diantara kedua huruf itu. Contoh: Au-la bukan a-u-la, sau-da-ra bukan sa-u-da-ra, am-boi bukan am-bo-i.
- b. Jika di tengah kata ada huruf konsonan, termasuk gabungan huruf konsonan diantara dua buah huruf vokal, pemenggalan dilakukan sebelum huruf konsonan. Contoh: ba-pak, la-wan, mu-ta-khir, ba-rang, de-ngan, su-lit, ke-nyang.
- c. Jika di tengah kata ada dua huruf konsonan yang berurutan, pemenggalan dilakukan di antara kedua huruf konsonan itu. Gabungan huruf konsonan tidak pernah diceraikan. Contoh: man-di, cap-lok, makh-luk, som-bong, ap-ril, swas-ta, bang-sa.
- d. Jika di tengah kata ada tiga buah huruf konsonan atau lebih, pemenggalan dilakukan di antara huruf konsonan yang pertama dan huruf konsonan yang

kedua. Contoh: ins-tru-men, in-fra, ben-trok, ul-tra, bang-krut, ikh-las.

Dalam tabel berikut penulis membedakan hasil pemisahan kata, jenis huruf, kalimatnya, suara serta jumlah uji coba kalimat pada *program*. Setiap uji coba *program* pada kalimat dilakukan dua kali uji coba. Berikut tabel unjuk kerja pada aplikasi konversi *text to speech*.

Tabel 1. hasil uji kalimat

No	Kalimat	Huruf	Pisah	Suara	Jumlah Uji Coba	
					Pertama	Kedua
1	Ayam kalkun	v-kvk kvk-kvk	a-yam kal-kun	a-yam kal-kun	✓	✓
2	Baja keras	vk-kv kv-kvk	ba-ja ke-ras	ba-ja ke-ras	✓	✓
3	Ambal kotor	vk-kvk kv-kvk	am-bal ko-tor	am-bal ko-tor	✓	✓
4	Jambang panjang	kvk-kvkk Kvk-kvkk	jam-bang pan-jang	jam-bang pan-jang	✓	✓
5	Teks proklamasi	Kvkk kkvk-kv- kvkv	teks prok-la-ma- si	teks prok-la-ma- si	✓	✓
6	Teman muksal	kv-kvk kvk-kvk	te-man mu-ksal	te-man muk-sal	x	x
7	Mental siswa	kvk-kvk kvk-vk	men-tal sis-wa	men-tal sis-wa	✓	✓
8	Bola mēntal	kv-kv kvk-kvk	bo-la men-tal	bo-la men-tal	x	x
9	Ahli strategi	vk-vk kkkv-kv-kv	ah-li stra-te-gi	ah-li stra-te-gi	✓	✓
10	Kontraktor asing	Kvk-kkvk- kvk v-kvkk	kon-trak-tor a-sing	kon-trak-tor a-sing	✓	✓
11	Instruktur olahraga	Vkk-kkvk- kvk v-kvk-kv- kv	ins-truk- tur o-lah-ra- ga	ins-truk- tur o-lah-ra- ga	✓	✓

Keterangan:

✓ = OK

X = Kurang OK

Pada tabel 1 kita dapat melihat bahwa sistem masih memiliki kekurangan. Diantaranya sistem tidak dapat membedakan huruf berjenis homograf. Yakni tulisan sama ucapan dan artinya berbeda. Selain itu *program* juga tidak dapat memisahkan kata berhuruf konsonan ganda. Tidak semuanya dapat sempurna dipisahkan. Seperti halnya yang tertera dalam tabel berikut "muksal" dan "teks", pada kata tersebut pemenggalan pada huruf "ks" dilakukan pada huruf vokal yang pertama kali jumpa namun jika ada huruf vokal berikutnya maka huruf "ks" dipisahkan mengikuti vokal yang terakhir kali. Seharusnya kata "muksal" menjadi "muk-sal" bukannya "mu-ksal". Hal ini terjadi dikarenakan pada sistem, penulis telah mengelompokkan bahwa huruf "ks" adalah konsonan rangkap sehingga huruf "ks" akan dianggap satu huruf. Namun pembacaan huruf yang telah diketikkan masih terbaca dengan baik.

Berikut beberapa daftar kata homograf yang tidak dapat dibedakan bacaannya oleh sistem aplikasi TTS.

Tabel 2. Daftar kata homograf

No	Kata	Arti Kata
1	Apel	Buah Upacara
2	Bela	Menjaga/Memelihara Tuntut/Balas Balik
3	Mental	Jiwa Terpantul
4	Teras	Inti Depan rumah
5	Sedan	Jenis Mobil Sedu sedan
6	Sepak	Tendangan dengan kaki Tamparan dengan kaki
7	Serak	Suara yang parau Tidak tersusun
8	Semak	Kawasan belukar Memeriksa dengan teliti

9	Rendang	Nama masakan Rimbun
10	Gelang	Bahan pembuat arang kayu Alat perhiasan wanita
11	Tekad	Menghasilkan bentuk dengan menggunakan acuan Membuat keputusan
12	Selak	Kunci Menyingkap kain
13	Perang	Sengketa Warna rambut

4. Kesimpulan

Adapun kesimpulan dari penelitian ini adalah sebagai berikut:

1. Aplikasi ini masih memiliki kekurangan diantaranya tidak dapat membedakan huruf bertipe homograf dan konsonan ganda. Namun tidak semua konsonan ganda tidak dapat dipisahkan oleh aplikasi.
2. Banyaknya sampel yang diuji seperti kata homograf, tidak dapat mengubah perubahan hasil *output* suara dikarenakan kata homograf yang sangat terbatas disimpan kedalam *database* Indo_TTS.
3. Pemisahan kata tidak sempurna apabila melebihi batas kata (maksimal 2 kata) yang dimasukkan kedalam *program*. Jika kata yang diketikkan lebih dua kata dalam satu kalimat, maka *program* akan menampilkan tetap menjadi dua kata, namun kata pertama dan kedua berada pada baris yang sama sebagai kesatuan *string* yang sama, sedangkan kata ketiga berada pada baris kedua sebagai *string* berikutnya. Itu disebabkan *array* pada *program* hanya dapat menampung dua buah *string*.
4. Suara yang dihasilkan oleh sistem *database* Indo_TTS sudah sangat baik dikarenakan banyak kata yang tidak lari dari apa yang telah di ketikkan pada sistem.

5. Berdasarkan kinerja sistem, pemenggalan kata lebih *efisien* jika kata langsung di ketikkan pada *box memo* daripada input-an melalui *file* bertipe *.txt. Namun jika aplikasi di-*restart* terlebih dahulu, sistem tetap dapat membacanya dengan baik. Sistem ini masih butuh pengembangan selanjutnya.

DAFTAR PUSTAKA

- Adriati W., R., Tolle, H., & Setyawati, O. (2016). Pengembangan Aplikasi Text-to-Speech Bahasa Indonesia Menggunakan Metode Finite State Automata Berbasis Android. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, Vol 5 (No. 1).
- Arman, A. A. (2003). *Konversi dari teks ke ucapan*. Bandung.
- Dutoit, T. (1997). *An Introduction to Text-to-Speech Synthesis*. Springer Netherlands.
- Isrokah, & Yasin, M. (2013). APLIKASI KONVERSI TEKS MENJADI SUARA DENGAN MENGGUNAKAN METODE PENGGAL KATA FINITE STATE AUTOMATA (FSA). *CYBER-TECHN*, Vol 7 (No. 02).
- Lee, K.-F. (1989). *Automatic Speech Recognition*. New york: Springer US.
- Nugroho, A. S. (2013). *Teori Bahasa dan Automata*. Graha ilmu.
- Wijaya, T., Susanto, S., & Salman, A. G. (2013). *SPEECH RECOGNITION BAHASA INDONESIA UNTUK ANDROID*. Jakarta: Universitas Bina Nusantara.